UNIVERSITY OF OSLO
Department of Informatics

OSS for health care in developing countries:

Comparative case studies of DHIS2 and patient based systems in Ethiopia and Vietnam

Master Thesis

Thanh Ngoc Nguyen

May 2007

Thanh Ngoc Nguyen

OSS For Health Care in Developing Countries:

Comparative Case Studies of DHIS2 and Patient Based Systems In Ethiopia And Vietnam

Master Thesis

Department of Informatics

May 2007

UNIVERSITETET I OSLO

OSS FOR HEALTH CARE IN DEVELOPING COUNTRIES: COMPARATIVE CASE STUDIES OF DHIS2 AND PATIENT BASED SYSTEMS IN ETHIOPIA AND VIETNAM

Thesis submitted as partial fulfillment of the Requirements of the degree "master of science in Information systems" at the Department of Informatics, University of Oslo,, Norway

Thanh Ngoc Nguyen

Supervisors

Sundeep Sahay

Margunn Aanestad

ACKNOWLEDGEMENTS

First and foremost I would like to express a huge debt of gratitude to Professor Sundeep Sahay, without whom this thesis would have never been completed. I am also grateful for the many great opportunities to do co-located work with him. I thank him for being my "travel agent", coordinating my trips to India, Ethiopia, South Africa and Brazil. My time in Ethiopia was so wonderful. I also thank Margunn Aanestad for her guidance and support, from the initial stages of sketching out the thesis proposal right up to its completion.

A very big thanks to Jorn Braa, who brought the HISP project to Vietnam. It gave me a chance to join an interesting project. My time with HISP was a lot of fun. I thank him for always trusting and defending me; and for giving me the responsibility to take care of HISP Vietnam.

I would like to also thank Jens Kassboll for coordinating the Master program and taking care of us during two years of the program. I also, thank Line from IFI and Michele from Student Advisor who always supported me with the administrative formalities.

I also thank the HISP team, Knut, Ola and other members in the core team. Kristian, Lars, Togeir, Hans, Oyvind, Anders who gave me support during my stay in Oslo.

I also send my thanks to the Ethiopian team. Seid, Yeshambel, Selam, Girma and especially Mulugeta and Hailu who made the learning by doing process achieve fruitful results. Time in Ethiopia was unforgettable.

I would also like to thank my classmates, and other friends Edwin, Lisa, Hans who helped me to do the proof reading.

I thank my family: Dad, Mom, brothers and sisters who always supported me during my study.

Last but not the least, I would like to thank my honey who is always beside me. Encouraging me and sharing some great times, even when I was far away for my studies. 7 times you have seen me off at the airport. I wish the 8th time you will be coming along with me. Thanks for the patience and love you have given me.

Oslo. May 2007
Thanh Ngoc Nguyen

ABSTRACT

This thesis investigates issues related to the development and implementation of OSS (Open Source Software) for health care in developing countries. The aim is to understand why the same OSS technology for health care leads to varying outcomes in different developing countries, and how to find effective ways to organize it. A theoretical perspective inspired by the Social Construction of Technology (SCOT) theory was employed to set a base for the analysis process. A comparative case analysis was carried out over the development and use processes for two OSS applications (DHIS 2 and patient based systems) that were both developed and use in two developing countries – Ethiopia and Vietnam, Four comparative case studies (across countries and across systems) thus emerged. Social conditions inferred from the SCOT literature that served as the point of departure for the starting point of my analysis included 1) Team structure and organization, 2) Technical capacity, 3) Nature of the sector 4), and Technology introduction process. The study was informed by qualitative methods, and carried out within an action research framework over the period of 2005-2006. Through the analysis process, three main categories of social conditions relevant in shaping the outcomes were identified, which are 1) Technical Infrastructure, 2) Organizational Arrangements, and 3) Development Process. Points of departures between my theoretical inferences and as seen in the traditional SCOT literature were identified such as the level of internet dependence and the dedication and commitment of developers. These conditions were seen to be specifically relevant to the empirical domain characterized by OSS, developing countries and health care. In making these inferences, this thesis makes a contribution to the domain of social analysis of OSS applications. Practically, the research suggests implications in managing OSS projects for health care in developing countries such as emphasizing the role of appropriate team organization and structure, managing the technology introduction process, strengthening the link between development and use, and sensitively cultivating the support from the network.

Key words: OSS, developing countries, SCOT, health care, globally distributed systems development

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF PHOTOS

LIST OF ABBREVIATIONS

- ART Antiretroviral Therapy
- DHIS District Health Information Software
- DOPS Department of Planning and Statistics
- ECA Economic Commission for Africa
- EICTDA Ethiopian Information Communication Technology Development Authority
- EPR Electronic Patient Record
- EU European Union
- GIS Geographic Information System
- GDSD Globally Distributed Software Development
- HIS Health Information System
- HISP Health Information System Programme
- HMIS Health Management Information System
- ICT Information Communication Technology
- IHAMS Integrated HIV/AIDS Management System
- II Information Infrastructure
- IS Information System
- ISD Information Systems Development (ISD)
- IT Information Technology
- MDS Minimum Data Set
- MM Morbidity and Mortality
- MOH Ministry of Health
- NGO Non-Governmental Organization
- OSS Open Source Software
- SMS Short Message Service
- UNDP United Nations Development Programme
- UNICEF The United Nations Children's Fund
- UNPA United Nations Population Fund
- VCT Voluntary Counseling and Testing

CHAPTER 1: INTRODUCTION

1.1 The problem domain: OSS for health care in developing countries

Increasingly, information infrastructure of a country is seen as an essential element to economic growth. As UNCTAD Secretary-General Rubens Ricuperso recently stated: "*Among all the factors that can contribute to the competitiveness of developing countries, I can think of none more important than information technologies*" (Murphy 2004). However, while the potential of Information and Communication Technology (ICTs) no doubt exists, in practice the benefits are not easily realized. Information Systems (IS) researchers empirically examining the implementation of ICTs in different contexts have reported a wide range of cases where systems end up as total or partial failures (Avgerou and Walsham 2000). Major contributing reasons identified have been the structures (for example, funding mechanisms) and processes (for example, training and education methods) around the technology transfer projects. It has been argued by various quarters that the technological disparities between poor and rich countries have worsened as a consequence of the wide adoption of ICTs in developed countries, as evidenced in debates around the notion of "digital divide" (McNamara 2003). This gap proves a major barrier to battling ongoing problems such as poverty, health care (for example, the HIV/AIDS pandemic), environmental degradation etc.

Various arguments have been raised by IS and development studies researchers to establish "south-south" collaborative networks as contrasted with the traditional "north-south" networks. North-south projects, as Heeks (2003) argues often leads to the creation of design-reality gaps reflecting the chasm between the design assumptions inscribed in the technologies by the developers in the North and the practical realities of the ground where the technology is expected to be put in use. Heeks identified the following seven dimensions around which these gaps are prominent:

- **I**nformation

- **T**echnology
- **P**rocesses
- **O**bjectives and values
- **S**taffing and skills
- **M**anagement systems and structures
- **O**ther resources: time and money

(summarized by the ITPOSMO acronym)

A contrasting model, though not empirically tested in depth, can be south-south collaborative links, reflecting networks being constituted primarily by developers and institutions in the South. These networks can be further supported by taking on a structure of south-south-north, where the North playing a more enabling role rather than one of direct design, development and the control of implementation processes. These enabling roles can be in the form of providing specific technical guidance, provision of expertise and funding, and for supporting educational processes. These south-south linkages, also referred to as "counter-networks" (Mosse and Sahay, 2003) are seen to provide the potential to challenge technology imposition and unsustainable practices sometimes seen in north-south transfers, and allow the sharing of experiences, skills, products, and processes between similar contexts. These similarities in structures can thus help to close some of Heek's "design-reality" gaps, and potentially contribute to more effective applications of technology.

One important domain which should be the basis of such a counter-network involving south-south (-north) partnerships concerns Open Source Software (OSS), which provides the developing world with a powerful strategy to leverage the potential of ICTs for supporting local applications. OSS represents fundamentally a model of distributed, shared, open software development, and as a development philosophy gaining momentum in many developing countries. For example, a significant proportion of the Internet infrastructure is based on OSS products including send mail, BIND, Linux and Apache HTTPD, the latter running more than 69% of all active websites (approximately 15 millions) with Microsoft IIS coming a distant second with less than 23% share. OSS is arguably a cost effective and customizable alternative, potentially helping to prevent monopoly, and can be beneficial to the poor nations, where software license costs account for a significant proportion of governments' budgets on ICT infrastructure. For example, Meystre and Muller (2005) use the example of Vietnam to argue that "*the cost of Microsoft Windows XP and Office is higher than the average annual income!* "

OSS offers a wide spectrum of software programs developed not under the lock and key of a single company but through communal efforts, and is increasingly becoming the foundation of the Internet economy, powering much of the global flow of email and more than 70% of all web servers. The OSS movement is helping to provide alternative models, technologies and processes to businesses and governments in developing countries to compete more significantly in the global marketplace, as well as to enhance service quality and efficiencies in the public sector. An example of this is the Linux operating system which has grabbed a significant chunk of the server business once held by Unix, and is also becoming a significant desktop contender, with the increased backing of influential companies. With the rapid spread of Internet access, OSS offers a great potential for significant commercial gains for industries in developing countries, for example, in the domains of saving on licenses for standard business software, for specialized software serving a range of industries, and for contracts to do customizations and to add features. Provided the skills to exploit it are developed locally, OSS offers an excellent foundation for commercial software development serving local or global markets.

Furthermore, OSS provides governments in Asia, Africa, and Latin America with freely available ICT-based tools to address chronic problems of poverty, low productivity, population growth, wide scale unemployment, primary product export dependence, epidemics, and illiteracy. Several European countries are now promoting OSS and setting up international networks and United Nations Development Program (UNDP) has established a network for the Asia-Pacific region[1]. Developing countries are increasingly facing up to the

[1] UNDP website, http://www.apdip.net/

potential of OSS. Examples include South Africa[2] and Vietnam[3] making important policy decisions, Brazil setting up a "Chamber for the Implementation of Software Libre"[4], and Malaysia recently introducing a "Public Sector Open Source Software Masters plan"[5]. Arguments for OSS are also being made forcefully in India, both relating to rural e-governance and for commercial software development, with the Indian president Dr. Abdul Kalam a strong proponent and states forming OSS policies[6].

This increasing momentum due to dynamics in the domains of the industry, markets and politics helps to make a compelling case for a developing country to adopt an OSS driven ICT strategy. OSS should not be viewed as a mere product choice, but more fundamentally as an alternative strategy for building, maintaining and changing the rules that govern information flows in the economy. While OSS represents a real paradigm shift in how organizations approach ICTs (Fitzgerald, 2006), there are several challenges in making them work effectively in practice, for example:

- There are limited institutional mechanisms in place to support sharing of OSS both within and across countries.
- OSS as an ideology, to be effectively implemented on the ground, needs to be supported by alternative organizational forms and business models, for example through pricing mechanisms for training and support.
- The nuances of licensing arrangements are still poorly understood, and policies need to be established in this regard.
- Technical capacity to deal with OSS development and support is still limited, and lies primarily in the domain of computer scientists working on closer to the machine applications (like operating systems and compilers). When we talk of applications within domains like health or education, there is very limited capacity to deal with OSS technologies.
- Technical support mechanisms for OSS applications are a big issue as there are

[2] South Africa OSS, http://www.oss.gov.za/docs/OSS_Strategy_v3.pdf
[3] Vietnam OSS, http://www.oss.gov.vn/index.php?SetLanguage=0&lang=en
[4] Linux Today, http://www.linuxtoday.com/news_story.php3?ltsn=2003-06-13-009-26-OS-LL-PB
[5] Malaysia OSS, http://opensource.mampu.gov.my
[6] Time India, http://timesofindia.indiatimes.com/cms.dll/html/uncomp/articleshow?msid=47799819

often no clear accountabilities within a community based development and support model.

- The capacities of universities and technical institutions need to be strengthened in order to diffuse greater competence amongst the development community to effectively deal with OSS technology.
- Language issues to enable transfer and use of OSS from one context to another are significant.

As the above list indicates, the opportunities of leveraging OSS applications for developing countries are tremendous, but so are the challenges. With the view that we know little of how these applications can be made to work in practice, we focus on trying to develop insights into the micro-level practices surrounding OSS software development process in a developing country context. These micro level processes concern how learning takes place, how developer capacity is created, and how particular software choices are made. The empirical focus is Ethiopia and Vietnam, and the domain of application is health information system (includes both aggregate-based system and patient based system).

While in general, it is challenge to make OSS work in practice, applying OSS for the health care domain in developing countries is more difficult than other sectors because the nature of health care domain are largely different from OSS applications in general which are typically closer to the machine (for example operating systems and compilers). For example, the health care domain requires applications which are more specific and "context dependent" compared to the OSS application in general, second is the technical expertise in health care is much lower than one primarily involving computer scientists (Nhamposa and Sahay 2005).

So far, literature on OSS for health care is quite limited (Muller and Meystre 2005). Most of the published examples on this topic come from experiences based in developed countries, with literally nothing written about OSS in health care in developing countries. Another key theme in these published papers is the key argument made to propose the use of OSS in health care to reduce the cost and dependency to vendors. These papers also explain the barriers which prevent the dissemination of OSS in health care. While many of these barriers are generic in nature and apply to OSS in general, the health sector has certain particularities arising from the conservative nature of the health managers and also the heavy work burdens of the health workers (Chilundo 2004). These particularities are magnified in the context of developing countries, because of further problems such as of fragmentation contributed by donor funding policies aimed at supporting specific health programs (Chilundo 2004). Furthermore, infrastructure constraints such as limited bandwidth and lack of capacity in OSS based technologies (Java, PHP etc) add to the challenges to apply OSS.

In discussions around OSS, it is often that the social context is not emphasized, implying that the same technology can be seamlessly adapted for different contexts in such developing countries (for example, Weber 2001, Weerawarana and Weeratunga 2004). Technology deterministic views have been now explicitly criticized by IS researchers (Smith and Marx 1994). However, these arguments have not been empirically examined in the context of OSS, especially within public health sector in developing countries.

Given these arguments, the thesis aims to understand the nature of social context which shapes the outcomes of applying OSS for health care in developing countries. Furthermore, the thesis will try to answer the question: "How can the process of development and implementation of OSS be organized more effectively?" We now discuss the theoretical basis of this thesis.

1.2 Theoretical basis

To address issues of analysis of the technology and organizational relationships, there are several approaches. One important one concerns the Social Construction of Technology (popularly referred to as SCOT) (Bijker *et al.* 1987). SCOT emphasizes the role of social context in shaping technology related processes, arguing that technology does not define human behavior but the reverse also equally applies, and the only way to understand technology is by analyzing it in its embedded historical-social-political context. Also, there are no universal views of technology in general, but difference with relevant social groups. Since the same technology means differently for various relevant social groups, who have their own ways to interpret it, thus contributing to "interpretive flexibility" (Bijker *et al.*

1984).

The SCOT theory has been widely used to address the question "why the same technology has different outcomes in different social settings" (Barley 1986, Orlikowski 1992, Robey and Sahay 1996, Nhampossa and Sahay 2005). Barley (1986) studied the divergent consequences of the same medical technology (CT scanner) in two neighboring hospitals: called the urban and the suburban. The divergence in consequences was analyzed as arising from different social interpretative processes in the two hospitals. Barley (1986) argued that technology was an "occasion for", not a determinant of social change (Barley 1986, Robey and Sahay 1996). The CASE (Computer Aided Software Engineering) tool example in Orlikowski's study (1993) revealed a diversity of outcomes in different project groups adopting the same tool to support their daily work. Some project groups with higher technical capacity could modify the tool to overcome the design limitations in the tool, while others could not and had to come back to do the work manually. Robey and Sahay (1996) also reported the diversity of outcomes in two county governments adopting the same technology (Geographic Information Systems - GIS). They argued that the new GIS technology was subject to differing interpretations depending upon the social context and the process of implementation. Divergent interpretations of identical technologies were also reported by Nhampossa and Sahay (2005) who studied the customization process of the same District Health Information Software (DHIS) in Mozambique and India. They identified several differences' in social context which led to different results in adapting the same application in India and Mozambique namely - the health sector context, the structure of the team, the nature of the software customization process and the installed base.

As discussed so far, SCOT has been used widely to deal with the problem of "same technology – different outcomes". In the case of OSS for health care in developing countries, SCOT becomes very relevant because of the following reasons:

- Health care sector is inherently composed of diversity, both within and across countries contributed to by different health policies of governments, the capacity of health managers and health workers, variations in infrastructure (such as electricity, computer, etc), the different organization structures, and the varying level of dependencies on foreign donors. This significant diversity in the social setting provides the potential of creating "interpretive flexibility" around the HIS amongst the different "relevant social groups", which then gives rise to the potential for varying outcomes around the same technologies. This scenario sets the ground for the application of SCOT in my thesis.

- In spite of being developing countries, the social contexts in these countries are very different from each other, and vary in their levels of development. For example, Vietnam may have a better ICT (Information Communication Technology) infrastructure as compared to Ethiopia, indicated for example by the rate of people using Internet. Or for example, India has a much stronger pool of software developers than most other developing countries. These varying levels of development shaped by different socio-cultural-historical processes also contribute to the potential of the same technology having different outcomes in varying settings.

- It is often assumed that OSS will function uniformly well in different contexts, including that of developing countries. However, the different conditions of infrastructure, expertise in technologies, human resources capacity create conditions for varying outcomes of OSS applications.

Given the potential of varying outcomes of developing and applying same OSS in different developing countries, this thesis aims to answer the following two questions:

- Why "the same technology but different outcomes" in applying OSS for health care in developing countries?
- How the development and implementation of OSS for health in developing countries can be organized more effectively?

1.3 The empirical basis

An effective HIS can help developing countries improve their pubic health system (AbouZahr and Boerma 2005). Vietnam and Ethiopia are both developing countries which share a lot of common problems in education, health, poverty, environment etc. The two governments have made significant efforts to improve the situation, including in the health sector, by incorporating ICTs. The continuing low level of development of health status, indicates the limited contribution that ICTs have made on the ground. There is in agreement on the need for focused action on the ground to address the problems, including strengthening of the HIS. For example, AbouZahr and Boerma (2005) have argued that:

"Public health decision making is critically dependent on the timely availability of sound data. The role of health information system is to generate, analyze and disseminate such data". (WHO bulletin 2005, p. 578)

However, amongst other things, achieving the above requires effective software solutions, which a country can either buy or develop on their own. Given the high costs of proprietary software, OSS solutions provide a potentially effective alternative (Meystre and Muller 2005) because of lower costs and reduced vendor lock in. However, practically leveraging the benefits of OSS on the ground remains largely an unanswered challenge in the context of developing countries. Trying to contribute to the practical challenge of making OSS work in practice is a key aim of this thesis.

This research work is done within the context of HISP (Health Information Systems Program), a project that has been ongoing for more than 10 years in various countries with the aim to strengthen and build HIS in developing countries (Braa *et al.* 2004). The empirical basis for this research is case studies carried out in two countries (Vietnam and Ethiopia), over two years (2005-2006) involving two different software applications. The first case discusses the process of customization and implementation of DHIS2 (District Health Information System version 2) software in the two countries, while the second focuses on the process of development and implementation of a patient based system in the same two countries. The Patient based system includes several subsystems such as the ART (Antiretroviral Therapy) module for monitoring and managing ART treatment in clinics in Ethiopia, and a flexible system for generating patient based systems that is being piloted in Vietnam..

As the aim of the thesis is to understand "why the same technology (OSS for health care), has different outcomes in different settings" and "how applying OSS for health care in developing countries can be organized better", this research was designed around two cases studies in two developing countries : Ethiopia and Vietnam. In Ethiopia, the field work was done by working with the local HISP team on customizing DHIS2 software and initiating the development of an ART management system which was patient based. For the DHIS2 related work, the HISP Ethiopia team and I downloaded the DHIS2 source code and built it for the Ethiopian needs. After that, we made a conversion tool to migrate data from the previous format (DHIS1.4) to DHIS2. We also attempted to develop additional functionalities such as a report module and a morbidity and mortality module to use locally in Ethiopia. For the patient based system related work, another group of the HISP Ethiopia team and I started to build the system from scratch. During the development process, we used a prototyping approach involving close interaction with the end-users.

Similarly, in Vietnam, I also worked with the local team in both the development of the DHIS2 and another patient based system. Related to the former, we expanded DHIS2 by adding several modules for local and global use such as security module and GIS module. Related to the patient based system, I developed a flexible patient based system based on the ART system developed in Ethiopia. This system was also adapted for several places in Vietnam, however for reasons which I will discuss in the case study description, it had not actually been used.

The field work was thus done around two OSS for health care systems in two developing countries which created the empirical basis for this thesis. SCOT was used as the theoretical lens for analyzing how the differences in social contexts affect the outcomes of designing and applying OSS for health care in developing countries. Understanding the different social settings between Ethiopia and Vietnam helped to highlight its implications on the use of OSS in the two settings. Such an understanding also helped me develop practical implications on how globally distributed development of OSS can be better organized to suit the social settings of developing countries, in a more generalized way, Vietnam and Ethiopia in particular.

This research was framed with the umbrella of an action research project of HISP more generally, and informed by qualitative research methods. These included different data collection methods such as interviews, participant observation, meetings and presentations, workshops, document analysis, etc to understand various facets of OSS phenomenon seen from different perspectives. Interviews were conducted with coordinators of the HISP project, core development team members in Oslo, health managers in research sites, and members of the two local teams. Participant observation was done while participating with the team in actual development work. Meetings took place at different occasions, regularly with the team and occasionally with external partners such as health managers and third parties. Several workshops related to HIS were also a source of empirical data. I also extensively used the archive of the developer mailing list of the project and project websites to collect data relating to the problems being experienced by developers and how they tried to obtain solutions for the same.

This research was carried out over a two year period from 2005 to 2006 in two countries: Ethiopia and Vietnam. However, the development of DHIS2 started in Vietnam in 2005 and lasted until the end of 2006. From 2006, the research involved the development and implementation process of both DHIS2 and patient based system in both Ethiopia and Vietnam.

In summary, in this section, I have presented the empirical basis of this research, and provided some general information related to the field work, the research methods, the research design, and the research timeline. In the next section, I will discuss the expected contributions of the thesis.

1.4 Expected contributions

Theoretically, the aim of thesis is to contribute to the application of the SCOT theory to the analysis and use of OSS for health care in developing countries. This contributes both to the domain of SCOT, and also to the literature around design and development of OSS for health care in developing countries. Practically, I contribute by developing implications on how OSS development involving globally distributed teams can be better organized. Furthermore, by actually supporting the development of the systems, I have made a practical contribution of the development of a working application. In particular, this contribution includes:

- Helping the team in Ethiopia to develop the ART module by giving technical training and support.

- Developing the flexible system (OpenEPR – Open Electronic Patient Record)
- Participating in the development of several modules of DHIS2 such as Data View Module, Data Mart Module etc.

1.5 The structure of the thesis

This thesis is structured around the following chapters:

Chapter 2: Open Source Software (OSS) in developing countries involving Globally Distributed Software Development (GDSD)
This chapter discusses Globally Distributed Software Development (GDSD) and OSS for health care in developing countries.

Chapter 3: Theoretical perspective
This chapter discusses key concepts related to SCOT, Globally Distributed Software Development (GDSD) and OSS for health care in developing countries. I also discuss how SCOT has been applied in prior IS research.

Chapter 4: Research approach and methods
This chapter discusses the methods employed in this research, based on a qualitative approach within an action research framework. The comparative case study design adopted is discussed to enable an analysis of similar systems in different social settings and their associated outcomes. The chapter also describes the processes around data collection and its analysis.

Chapter 5: The research settings
In this chapter, I present the settings in which the research was conducted, including the introduction of the HISP initiative in Vietnam and Ethiopia

Chapter 6: The case of DHIS2
The chapter discusses the overall process of customizing and developing DHIS2 in Ethiopia and Vietnam from the beginning of 2005 till the end of 2006. The chapter describes how

DHIS2 was downloaded, customized and developed, and the process by which additional modules were incorporated to support local needs.

Chapter 7: The case of the patient-based system

Similar to the case of DHSI2, this chapter discusses the process of building and applying the patient based system in the two countries Ethiopia and Vietnam.

Chapter 8: Analysis

Based upon principles drawn from SCOT, this chapter identifies within a comparative framework the relation between social context and the outcomes concerning the design and use of the OSS applications across the two countries. Three categories of contextual differences were identified: technological infrastructure; organization arrangement; and development process.

Chapter 9: Discussions and conclusion

This chapter discusses the answers to the two research questions posed in the introduction of this thesis:

- Why "the same technology but different outcomes" in applying OSS for health care in developing countries?
- "How the development and implementation of OSS for health care in developing countries could be organized more effectively"?

CHAPTER 2: OPEN SOURCE SOFTWARE (OSS) IN DEVELOPING COUNTRIES INVOLVING GLOBALLY DISTRIBUTED SOFTWARE DEVELOPMENT (GDSD)

In this section, there are two main issues I discuss. The first one concerns OSS and the second GSSD. The OSS section is organized as follows. The first subsection discusses OSS in general, including a brief history. In the second subsection, I specifically discuss issues relating to OSS in developing countries. The last subsection is used to discuss OSS for health care in developing countries which is the focus of my thesis.

2.1.1 OSS: An overview

The OSS movement was initiated in the early 1980s by Richard Stallman. The key principles of OSS is described in his article in 2002: "*Why Software Should Not Have Owners*". He argued that the copyright in software restricts also the individual user, and thus argues for the copyleft scheme instead. The case of Linux, representing community work carried out by a Finnish student Linus Torvalds, exemplifies the case of the "mainstream OSS". A book written by Eric S. Raymond "The Cathedral and the Bazaar" in 2001 classifies two models of software development respectively called the Cathedral and the Bazaar. The Cathedral model represents an ordered and well-structured organization while the bazaar model concerns community based development. Raymond goes on to argue for the superiority of the bazaar over the cathedral model. However, this paper has been criticized by several researchers such as Bezroukov (1999) who has argued that OSS contains potential failure and vulnerability.

Bezroukov (1999) criticized that Raymond's bazaar model as it provides a too simplistic view of the OSS development process. To him, the OSS is not a progressive phenomenon (promising a bright future for mankind) with no problems. As for the novelty of distributing source code, Berzukovs argues that. IBM for many years had distributed its mainframe software as open source. He criticized that OSS, in Raymond's view, as just focusing on source code, and ignoring all of the complex infrastructure and implicit knowledge that are used in large software projects. The creation and maintenance of an intellectual environment and an efficient infrastructure for a successful proprietary business is not even discussed. He argued that it's naive to assume that the open source movement is free from the kind of fights, common in the corporate and academic worlds around ownership and status.

However, OSS has evolved fast and attracted more and more attention from the IT industry and now in different domains such as government policy makers. Big software companies like IBM, Sun etc have got involved in OSS and developed several OSS products such as Webphere, Eclipse etc. Fitzgerald (2006) has argued that the characteristics of the current OSS movement is very much different from what it was in the past, and names the current trend as OSS2.0 to distinguish it with the OSS in general in the past. He has argued:

"The open source software phenomenon has metamorphosed into a more mainstream and commercially viable form, which I label as OSS2.0"

(Fitzgerald 2006, p.2)

Drawing upon Tushman and Andersen's (1986) framework, Fitzgerald argues there is a transition from OSS to OSS2.0 because there are changes in both the product and process of making software. In the product set, OSS2.0 is based on business strategies, support and licensing. In the process set, OSS2.0 is based on "purposive planning", is "less bazaar-like" and "developers are being paid".

The argument of Fitzgerald reflects the situation where the current OSS products mostly fall into the generic domain such as operating systems, integrated development environment software, database server etc. The requirements of such software are common and widely understood as they are guided by technical protocols universally understood by computer scientists speaking to each other. In contrast, applications for specific domains such as health care, education or business software are more context-dependent rather than being closer to the machine specific systems. As compared to machine specific applications like operating systems (ie. Windows or Linux) the user needs are rather uniform except maybe around the user interface language. The situation is very different in domain-specific software because of the need to engage with different social contexts, groups, and their inerpretive flexibilities. In these cases, the OSS applications are expected to meet varying requirements of users located in different places. It thus raises a real challenge for making such kind of software because it requires a flexibility in design which can address these varying needs. Then adapting an OSS into local contexts is a process of interpretation of technology, and structuring the development process in particular ways to meet local needs.

Although claimed as open source and free of charge, OSS also has licences involved. An OSS license is a copyright license for computer software that makes the source code available under terms that allow for modification and redistribution without having to pay the original author. Such licenses may have additional restrictions such as a requirement to preserve the name of the authors and the copyright statement within the code. One popular (and sometimes considered normative) set of open source licenses are those approved by the Open Source Initiative (OSI) based on their Open Source Definition (OSD). Licensing is one of the confusing aspects of OSS because of several reasons. First, there are many kinds of licenses. Till the year 2005, OSI had approved some 60 different licenses. There are also other non-approved licenses such as HESSLA, Lemur License Agreement, MAME (source available, but not free software because it forbids commercial use and redistribution), PGP, Ruby License etc. Second, every license has different constraints, rules and regulations which can be difficult to understand. For example, software published in GPL license requires that all software built upon it have to be published as the same license. It means that if one company develops software based on a GPL-licensed software, the source code of their software has to be public. Therefore, GPL is not a business-friendly license. The complexity of OSS license is also a key deterrent for people to use OSS. Every organization or individual before adopting an OSS application has to investigate the licensing arrangements and the implications that has for their own context.

In this section, I have discussed several issues related to OSS in general. They are quite generic and can be applied for various contexts both in developed or developing countries. Applying OSS in developing countries, however shares both similarities and differences with developed countries. In the next section, I discuss OSS within the sepcific context of developing countries.

2.1.2 OSS in developing countries

OSS is attracting more attention of researchers, policy makers and software vendors in various parts of the world, including in developing countries. Weber (2001) describes the emerging picture of OSS as follows:

"Over the past five years governments around the world have begun considering legislation that would require the use of OSFS when it provides a feasible alternative to proprietary software. This phenomenon has been particularly pronounced in the developing world as these nations, struggling with limited information technology budgets, look to the potential welfare gains from deploying OSFS solutions"
(Weber 2001, p.10)

The benefits of OSS are spelled out to include cost savings, economic leverage, security and autonomy, and intellectual property (Weber 2001). The cost of proprietary software is a concern for both developed and developing countries. For example, the Taiwanese Government estimates that it could save nearly $300 Million. in royalty payments through a strategic open source project that encourages research and development in office software and the opening of source code for government agencies and private enterprises as well. In the US, a new bill introduced in Oregon in April 2003, which could be the first bill in the country to encourage the use of OSS by a state government, says that open source options can significantly reduce the state's cost of obtaining and maintaining software (Weber 2001).

In developing countries, the problem of costs to buy licensed software is magnified. For example, the license costs of Windows can be more than a month's salary of people in these countries. A comparison table made by Weerawarana and Weeratunga (2004) states that the number of months salary (based on GDP per capital) for Vietnamese and African citizens to buy license of WindowsXP are 16.33 months and 10.31 months respectively. The vision of the information society will never be achieved in these poor countries if it is based primarily on proprietary software, and the digital divide will only increase, not reduce.

OSS does not only bring potential economic savings but also gives the countries alternatives to select multiple software vendors who are distributed around the globe rather than being specific to a particular country. For example, in the proprietary domain, the most commonly used software are owned by US based companies (Windows, Office, SQL Server, Internet Explorer, Adobe, Macromedia). In the OSS domain, the "owners" of the software are more distributed (Linux – Finland, Ubuntu - South Africa, Mysql - Sweden etc). Therefore, choosing OSS helps the users to not necessarily depend on any particular country or region. Also, the source code of OSS is public, enabling countries to use that and make their own applications, of course if they have the required capacity and infrastructure. For example, based on Linux, China has made Red Flag Linux, Vietnam has made Vietkey Linux and South Africa has created Ubuntu. . However, these examples still primarily concern closer to the machine applications and not user domain specific ones like in health care and education.

The other benefits of adopting OSS in developing countries are related to technology development and capacity building. Camara and Fonseca (2005) argue that:

"Many authors propose that open source software (OSS) is a good strategy to bring information and communication technologies to developing countries. Nevertheless, the use of OSS needs to be more than just adopting Linux as the standard for operating systems. Adoption of OSS is not only a choice of software, but also a means of acquiring knowledge. Developing countries have to use OSS as a way to gain knowledge about the technology itself and as a way of creating technology products that fit their specific needs"
(Camara and Fonseca 2005, p.1)

Open source software is thus considered to have a potential impact on processes of knowledge acquisition by developing nations. Combining free software tools with the technical workforce available in developing countries can enable technology transfer (Camara and Fonseca 2005). This point of view was also supported by Weber (2004) who argues that "….the essence of open source is not the software. It is the process by which software is created" (Weber, 2004 p.56). He expects OSS to have far-reaching effects: "Of course information technology and open source in particular is not a silver bullet for long–

standing development issues; nothing is. But the transformative potential of computing does create new opportunities to make progress on development problems that have been intransigent" (Weber, 2004 p. 254). Clearly, the interrelation between adaptation of OSS and capacity for technology development has implications on broader development processes (James 2002). The adaptation of OSS creates the opportunity for developers in developing countries to experiment the technologies:

"Free software can also have 'a positive multiplier effect in the sense of encouraging the programmer community to explore and experiment with developing applications. As the barriers to entry in the case of proprietary software (high cost, poor availability, lack of online documentation and lack of source code) do not apply to free software, freedom of experimentation is very high"

(James 2002, p.28)

However, while the opportunities of OSS (even for application domains) are highlighted by researchers and policy makers, the challenges and difficulties of developing and applying them in practice have not been adequately discussed. There is a common assumption made by different groups that the adoption of OSS automatically will bring benefits of cost saving, independence, ownership, IT growth and capacity building (for example, Weerawarana and Weeratunga 2004, Weber 2001, James 2002, James 2002, Bokhari and Rehman 1999, Kshetri 2004). In reality, developing countries are struggling hard to make OSS work in practice. For example, in Vietnam, seven years after the decision of the Prime Minister (Decision 235/QĐ-TTg[7]) to boost OSS development and use in the country, Vietnam is still ranked at the top of the list of countries using pirate copies of Windows, making the task of applying OSS in developing countries to represent a complex challenge. James (2002) discusses the constraints of piracy, awareness and path dependency to shape the uptake of OSS in developing countries.

As argued earlier, adopting OSS to developing countries, if successful, can potentially contribute to the socio-economic development of these countries. However, OSS as a kind of technology can not be transferred into local contexts (from North America or Europe) seamlessly with the assumption that they all will work well. The social context, as argued by

[7] http://www.mpt.gov.vn/details_law_e.asp?LawDoc_ID=100138

various SCOT studies and also other researchers that have adopted a socio-technical perspective, will make a difference in how processes of adoption and use will unfold. Nhampossa and Sahay (2005) have pointed to three different aspects of the social context that can make a difference in OSS adaptation and use: the nature and specificities of the domain; the structure and organization of the development and implementation teams; and, the process of adopting the technology itself. .

The problems of "design-reality" gaps (Heeks 2003) between developed and developing countries in the context of ICT projects more generally have been reasonably well documented, even across developing countries despite sharing similarities with respect to poor development levels, low income, inadequate infrastructure etc. However, there are various technological, cultural and political conditions which contribute to large variations in the context. These undoubtedly will have a significant bearing on the processes around applying OSS.

Applying OSS concerns at least two distinct facets of: development; and field implementation and use. Development can either involve adapting the existing OSS solutions to the local context, or developing applications from scratch by using available OSS tools. Either way, both processes are context dependent, coming with their own sets of challenges and opportunities. Implementation and use of OSS is also a context dependent process because it is intimately linked to the end-user's background, organizational structure, training, and implementation mechanisms. A key challenge arises from the fact that OSS often does not have a real owner, which in turn implies that it may be difficult to get effective support. The users of OSS have the tendency to be unwilling to pay for third parties to get the professional support, as they believe the OSS is free.

After this more general discussion around the nature of OSS and its relation with the context of developing countries, I will turn my attention specifically to the context of OSS in the domain of health care, which is the focus of this thesis.

2.1.3 OSS in health care in developing countries

Literature on OSS for health care is relatively limited (Muller and Meystre 2005), and furthermore, most of the published examples on this topic come from experiences based in developed countries, with literally nothing written about OSS in health care in developing countries. For example, Carnall (2000) published a letter advocating the use of OSS in health care. Wright and Murray (2002) have also proposed the creation of the IMIA (International Medical Informatics Association) open source working group.

The key theme in these published papers concerns the key argument to propose the use of OSS in the health care sector to reduce the cost and dependency on vendors. They also explain the barriers which prevent the dissemination of such applications in general for example (Muller and Meystre 2005). While many of these barriers are generic in nature and apply to OSS in general, the health sector has certain particularities arising from the conservative nature of the health managers and also the heavy work burdens of the health workers (Muller and Meystre 2005). These particularities are magnified in the context of developing countries, because of further problems such as of fragmentation contributed by donor funding policies aimed at supporting specific health programs (Chilundo, 2004). Furthermore, infrastructure constraints such as limited band width and lack of capacity in people to work with the newer OSS based technologies adds to the challenges to apply OSS in this context.

Although the literature on OSS for health care in developing counties is limited, there is information about OSS projects for health care published on the Internet. I conducted a search in two popular OSS repositories - Sourceforge.net[8] and freshmeat.net[9] – using the following keywords: health, developing countries, electronic patient record etc. The results of this survey are presented in the Table 1:

Keywords	Number of results found in	Number of results found in

[8] http://sourceforge.net/
[9] http://freshmeat.net/

	sourceforge.net	freshmeat.net
Health	149	58
Patient based	70	10
Developing countries	1321	0
Patient + developing + countries	0	0
Health + developing + countries	0	0

Table 1: A survey on OSS for health care in developing countries

The results of the survey provides several interesting conclusions, summarized as follows:

- There are more than 200 projects targeted for health care, including for health information systems for routine data, patient record and other health related management systems.
- Among them, at least 80 projects are specific for patient record
- There are more than 1321 projects for developing countries but none specific to health
- When I combined the key word: health+developing+countries or patient +developing +countries, the results returned was 0. This finding reinforced my hypothesis: there is a assumption that OSS can be developed once and reused in all other places regardless of the different social conditions.

I now discuss some specific examples of OSS for health care specific in developing countries as obtained from the literature and Internet resources related to OSS.

District Health Information Software (DHIS versions 1.3 and 1.4 – hisp.info)

Braa *et al.*. (2004) have described the development and use of the DHIS software ongoing in several African and Asian countries. The DHIS1.3 and 1.4 is an aggregated based system which allows to capture health data at different levels. The software was first developed in South Africa in 1994. It is considered an OSS since the source code is open, although it is Access-based which is not license free. The system has been over the years rolled out to the whole South Africa, and since 1999, has also been has been transfered to several countries such as Mozambique, India, Tanzania, Ethiopia, Vietnam etc. However, the development of the software which was centered in South Africa, made the process of transition to other countries difficult for various reasons, including the centralization of knowledge about the system. This also impeded the sustainability of the implementation of DHIS in some of these countries. The authors pointed out several approaches to address the issues of sustainability such as the flexible design of DHIS, associated training and education process etc. The authors have also argued for the need to strengthen the network between developing countries to support processes of sustainability and scalability.

Integrated hospital management system (Care2x – care2x.org)

In contrast to DHIS, Care2x is an OSS specifically for integrated hospital management. While DHIS is working with aggregated data, Care2x is concerned on individual patients. Care2x was first developed in Germany, but there have been subsequent attempts to adapt it to some non-Western contexts. The mission of Care2x as stated in their website (http://care2x..org) is to strive to develop the most useful and practical integrated health care information system which is open for others to develop further. It has been implemented in several countries, both developed and developing, such as Germany, Italy, Malaysia, Vietnam, Costa Rica, Brazil etc. Care2x provides an integrated solution for hospital management which includes outpatient, inpatient, laboratory, finance etc.

Although, it is a complete solution for hospital management, Care2x appears to be inappropriate for other situations of health care management. For example, when we tried to adapt Care2x for ART management in Ethiopia, we found it so difficult to customize it. The weakness of Care2x comes from its hardcoded design meaning that there is no way for end-users to customize it easily. All customization works need to be done by either interfering with the source code or making plug-ins. Therefore, applying Care2x is quite a difficult process and requires lots of technical support from professional agencies. In general, Care2x development can be considered as a successful case of OSS development because of two reasons. First, the Care2x has a big development team currently comprising of more than 100 members with different skills and backgrounds, coming from more than 20 nations, including from Europe, Asia, and South America. Second, Care2x has been used in many countries in Europe, Asia, South America.

Amongst the African countries, Care2x has received the attention of Tanzania OSS community. The research wing of the University Research Center has started customizing Care2x for Hindu Mandal Hospital and UDSM Health Centre[10]. Actual details of the status of implementation is not easily available.

Open medical record system (OpenMRS[11])

OpenMRS is another OSS which also deals with individual patients. The contrast between Care2x and OpenMRS is that the latter allows end-users without programming knowledge to also be able to customize the system as per their needs. The development of OpenMRS has been managed by a group in South Africa, and has been implemented in various places such as Kenya, Malawi, Rwanda, South Africa etc. On February 14 2007, OpenMRS celebrated the one year anniversary of its implementation in Eldoret, Kenya. To date, the system has stored close to 10 million patient-level measurements on 43,000 patients who have accumulated ~450,000 visits. (openmrs.org). OpenMRS is developed in Java using several modern frameworks such as Spring, Hibernate, etc. In general, OpenMRS can be used to generate any kind of patient based applications. However, OpenMRS still has some technical limitations, making it hard to be a fully integrated hospital management like Care2x. For example, OpenMRS requires that other flexible forms must be linked up with the patient form, a feature which prevents OpenMRS from generating rather complex systems with many relationships between different forms. The second limitation of OpenMRS is that it stores data of patient and historical treatment in a special data model which makes it difficult to retrieve historical data for reporting and analysis purposes. The third limitation of OpenMRS is that it relies on InfoPath[12], a commercial tool, to design and edit forms.

[10] http://www.tanzaniagateway.org/news/news/article.asp?ID=101
[11] http://openmrs.org
[12] http://office.microsoft.com/en-us/infopath/FX100487661033.aspx

The above discussion about OSS for health care in developing countries is now summarized in the Table 2:

	DHIS 1.3, 1.4	Care2x	OpenMRS
Technology	MS Visual Basic, Access	PHP, Mysql	Java, any DBMSs
Kind of system	Aggregate based system	Patient based	Patient based
Development	Center in South Africa	Distributed (100 nations)	A team of 6 members
Design	Flexible in defining data elements, indicators	Fixed design for data entry forms	Flexible in designing new forms
Implementation sites	Africa, Asia	Europe, Asia, South America, Africa	Mainly in African countries, extends to South America

Table 2: The summary of typical OSS projects for health care in developing countries.

In this section, I have discussed OSS in general and also specifically for health care in developing countries. In the next section, I discuss processes of global software development and how this relates to OSS.

2.1.4 The global distributed software development (GDSD) around OSS

Along with the rapid globalization of companies, the globalization of software development has also become a reality. Many software projects are now distributed in diverse sites across the globe. The distance between these sites creates several problems that did not exist previously for collocated teams. There have been reported problems with the coordination of the activities, as well as with the communication between team members (Redmiles *et al.* 2007). The potential benefits of the globally-distributed development model include reduced costs, closer proximity to customers, enhanced global presence. However, proven methods for successful distributed development have not yet been formulated, especially in the domain of OSS. Software has several defining characteristics which includes the ability to be replicated, transmitted, amended and even used across almost unlimited distances. However, it is also subject to change, intangible and can be extremely complex (Fitzgerald 2006).

When discussing the challenges in adopting OSS in developing countries, many of the more generic challenges of distributed work apply, such as the seperation of time and space (Sarker and Sahay 2004; Krishna *et al.* 2004) and the social-cultural differences between team members (Walsham 2002; Nicholson and Sahay 2004; Sahay 2003).
They are now discussed as follows.

Cultural differences

A key challenge in GDSD reported in the literature concerns cultural differences. Although globalization, and the related spread of ICTs, implies that the world is becoming more homogeneous, the differences between organizations and societies will not disappear. For example, Robertson (1992) discussed the way in which imported themes are indigenized, in particular some societies and cultures constrain the receptivity to some ideas more than others, and adapt them in specific ways. While accepting the idea of time-space compression facilitated by ICTs, Robertson argued that one of its main consequences is an exacerbation of collisions between global, societal, and communal attitudes. Similarly, Appadurai (1997), coming from a non-Western background, argued against the global homogenization thesis on the grounds that different societies will appropriate the "materials of modernity" differently depending on their specific geographies, histories, and languages. Walsham (2002) developed a related argument, with a specific focus on the role of ICTs, concluding that global diversity needs to be a key focus when developing and using such technologies. Then working with ICTs in and across different cultures should prove to be problematic, in that there will be different views of the relevance, applicability, and value of particular modes of working and use of ICTs which may produce conflict. For example, technology transfer from one society to another involves the importing of that technology into an "alien" cultural context where its value may not be perceived in a similar way to that in its original host culture.

OSS development, both traditional and the new model, actually needs to address similar challenges related to cultural differences. Indeed, OSS development also gathers developers coming from different countries and regions around the globe with diversities in culture. The potential of collisions and conflicts in OSS development because of cultural differences is also visible. Bezroukov (1999) argued that it's naive to assume that the OSS movement is free from the kind of fights common in the corporate and academic worlds (Bezroukov 1999).

Embedded knowledge

Software development involves a variety of cognitive and organizational issues concerning the communication and co-ordination of knowledge relating to the program, the methodologies to be used, the domain area and various organizational practices such as reporting relationships within the project team. Managing these processes by which knowledge is acquired, shared, and integrated between these various individuals, teams and organizations is a crucial task in the process of software development (Nicholson and Sahay 2004). GDSD is different from co-located development for three reasons: the diversity of organizations and cultures co-involved in the development process, the nature of work; and the processual nature of the offshore relationship. These characteristics contribute to the challenge of embeddedness of knowledge (Nicholson and Sahay 2004). For example, Imsland and Sahay (2004) in the case study of Rusian and Norwegian software outsourcing project point out several issues of embedded knowledge:

- The system of Norwegian tax and salary rules
- Limited access to English language education from the Russian side.
- The Russian organizational practice where hierarchy was emphasized while Norwegian relatively flat and egalitarian style founded largely on trust.

These kinds of embedness knowledge led to delays of the project and other consequencses such as the system being specified incorrectly etc.

OSS development also shares the challenges of the embedded knowledge because it gets involved different individuals and organizations around the world with very different backgrounds, skills, and work practices. First, the embedded knowledge is related to the languages used as communication means in OSS projects. As developers of OSS projects are from different regions of the world, English is often used as the main communication language. In some parts of the world where English is not widely used, the developers will have problems exchanging information with the global teams. Second, the technology related embedded knowledge has a big affection towards OSS as every developer coming from different backgrounds tends to use different tools for the development work. Thus, it raises the complexity in managing the projects. Third, the challenge of embedded knowledge becomes more serious in the case of OSS applications for specific domains such as health care, education etc because the development of these applications requires more situated and domain specific understanding and knowledge.

Distributed physical location

One of the problems related to distributed physical location are the lack of human connection because the physical separation does not allow physical situatedness, an important precondition for meaningful face-to-face human interactions (Giddens, 1984; Sarker and Sahay, 2004). For examle, one team does not know whether the remote team pays attention to what they say; or can not judge the reactions of the other teams. Also, the seperation makes it difficult to establish inter-personal relationships and social glues.

The other problem of distributed physical location is ineffective communication because norms of turn-taking in conversation and presence that are usually well-established among individuals in a face-to-face context are not applicable when interactions, synchronous or asynchronous, occur in a virtual medium (Sarker and Sahay, 2004). At least 5 among 8 teams, in the research made by Sarker and Sahay (2004), felt it was difficult to communicate over electronic means as stated in the following quote:

"When the project first started, both sides would post messages and then it would be days before anyone would respond to those messages. So nobody was sure if the other side was checking WebCT, or was ignoring messages that were posted"

(Sarker and Sahay, 2004, p.10)

The distributed physical location also caused the problem of suspicion, since one could not verify the actions of the remote members because of the absence of visual observation (Weisband, 2002; Armstrong & Cole,2002; Sarker and Sahay, 2004). The example which is taken from the case study of Sarker and Sahay (2004) concludes that there were 7 among 8 teams having the sense of suspicion, as illustrated in the following quote:

"We have been reading your design document, and we are confused about some of the contents. Have you really been reading our systems requirement document?"
(Sarker and Sahay, 2004, p.10)

OSS presented as 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 by means of email and bulletin boards (Mockus, 2000). The challenges of distributed physical location will therefore exist.

Technology related diversity

Every team in different locations, due to their background, tends to use several technologies, frameworks which can be very different from those used by other teams. Some of the difficulties that arise include:

- Difficulty in collaborating due to mismatch in Information Systems Development (ISD) philosophies and approaches, and areas of expertise:
- Difficulty in collaborating due to differences in skills and training:
- Difficulty in collaborating due to mismatch in ISD language:
- Difficulty in collaborating due to mismatch in IT infrastructure:
- Rejecting the use of a communication channel due to differences in the economic drivers of technology use:

(Sarker and Sahay 2004)

Some very interesting examples were extracted from the case study of the US based and Norwegian based team by Sarker and Sahay (2004). For example, the US based team thought all computers in the world used Windows while Unix was more popular in Norwegian universities. What was even more funny was that the team in US interpreted that "prototype is the first release of a working system" while the Norwegian team thought a prototype was just a series of screens without any functionalities. The difference in IT infrastructure also caused difficulties in converting documents among different formats. For example, one team preferred to use pdf[13] over doc[14].

Technological frameworks related to OSS present a much more diverse picture than proprietary software. A search on sourceforge.net to find out tools and frameworks which supports OSS development returned with 19150 results[15]. Developers in OSS, therefore, have more choices than in the proprietary domain. This, however, contributes to more challenges related to technological diversity, and the task of coordinating them.

Time differences

Various challenges related to time difference are discussed in the literature such as the problems to work in parallel, unproductive waiting periods while receiving a reply from the other side, etc. Sarker and Sahay (2004) also proposed some strategies to sort out the challenges of time difference. For example, using the clock in one place as a standard; or relocating time by logging the content of discussion so that other team members can follow if they were absent.

Because developers of OSS projects are distributed in different places with different time zones, time differences also challenge the OSS development process. Also, since OSS developers are usually voluntary, often working only in their spare time, it may be difficult to enforce coordination mechanisms as may be the case in commercial projects.

[13] Portable document format - http://www.adobe.com/
[14] Microsoft Word format - http://office.microsoft.com/en-us/word/FX100487981033.aspx
[15] This search was done at http://sourceforge.net with the key word 'oss tool framwork '

Table 3 summarizes the similarities and differences challenged by GDSD and OSS. I also separate OSS into two categories: traditional OSS and the new OSS (OSS2.0), based on the classification proposed by Fitzgerald (2006).

Challenges	GDSD	Traditional OSS	The new OSS
Cultural difference	North-south cultural differences between software owners (usually developed countries) and the outsourcing suppliers (East Europe, India, Asia).	OSS software developers come from different places and bring the diversities in culture and access to infrastructure.	The same as the two previous models.
Embedded knowledge	Not easy for the suppliers to understand practices which are not existing in their contexts.	Mostly, products in the traditional OSS are generic enough to be easily understood by developers worldwide. However, the link between development and usage is not well established.	Same as GDSD.
Distributed physical location	Several co-located work periods (ie. on site) help to reduce the distance – geographic and cultural.	Chance for developers to meet is very rare.	Same as traditional OSS.
Technology related diversity	Technological choices are typically	Choices are determined typically by the various	Varied

	made by the owner of the project.	development communities engaged in the project.	
Time difference	Separation influences the project, but mechanisms to deal with it are determined typically by the owner.	No formal mechanisms are established typically to deal with the time differences.	Same as GDSD.

Table 3: Comparison among three GDSD models

In this section, I have discussed the challenges of GDSD and contrasted them with those experienced in the OSS domain. Arguably, these gaps are more magnified in the context of developing countries, making it more difficult to involve south-south networks in collaborative OSS development. The nature of such challenges within the context of OSS for health care in developing countries remains an open and empirical question. In this thesis, I explore these issues using the theoretical lens of SCOT, which I now discuss in the context of my research problem.

CHAPTER 3: THEORETICAL PERSPECTIVE

Introducing Open Source Software (OSS) to developing countries is the process of bringing technology to the social context and embedding it within. This process can be either effective or not as effective, since technology is accepted or rejected by the context, and is a process which develops over time. The thesis studies two cases of OSS application development and implementation in different set of contexts provided by two developing countries – Ethiopia and Vietnam. The aim of the thesis is to examine how the same technology gets shaped in varying ways by the respective social contexts, and analytically examine the relationship between the processes of implementation and development with the characteristics of the social context. Such an analysis can help to identify patterns in the similarities and differences in contexts, and how they shape the processes in varying ways. Such an analysis is especially urgent within the current context of globalization, where it is a crying challenge to understand how to take the same technology and related practices to different countries and also within the same country.

For the analysis of the relationship between the social context and the process of development and implementation, and more specifically how the same technology has varying outcomes in different contexts, I draw upon the theoretical perspective of Social Construction of Technology, popularly called SCOT. The SCOT approach has its roots in the theoretical perspective of sociology of technology, and has since the nineties been drawn upon by various researchers to examine the relationship between technology and the context within which it is developed, implemented and used. In the next section, I examine the basic tenets of the SCOT approach, and then I develop a perspective which can be drawn upon to examine the phenomenon under study in my case of – OSS application development and implementation in Vietnam and Ethiopia. Specifically, I examine how the same OSS gets adapted in different contexts. Given the extreme urgency in exploiting the possibility and potential that OSS provides to developing countries in supporting their attempts to effectively apply ICT for development, the analysis of how the social context shapes the adoption process, becomes a crucial challenge for IS research. However, before presenting my theoretical perspective, I first introduce SCOT and how it has been drawn upon in prior IS research.

3.1 The Social Construction of Technology perspective and IS research

The SCOT theoretical perspective was first formally proposed in 1987 by Wiebe Bijker, Trevor Pinch and Thomas P. Hughes. (Bijker *et al.* 1987). According to the SCOT approach, the technology by itself does not determine human behavior but in fact it is equally important to understand how human intention and behavior, and the characteristics of the social context, shape the processes by which the technology is developed and embedded in the social context. Some of the key arguments of the SCOT approach can be summarized as follows:

- Social causation: This principle suggests that those who are looking for the explanation for the success and failure of technologies should analyze the social context of technology. This principle is concerned with how the technology gets defined and by whom, and so also similar issues concerning the establishment of criteria for measuring the success of the technology.
- Symmetry: This principle suggests that all arguments explained for the acceptance or rejection of technology (social, culture, political, economics as well as technical) should be treated equally. The same criteria should be equally used to explain both the success and failure of the technology.
- Interpretative flexibility: This principle allows for a flexible interpretation of technology in varying ways. Groups of people, called “relevant social groups” in the SCOT literature, based on their background and interests, can interpret the same technology in different ways. For example, an automobile can be seen as a means of transportation by some or as freedom by others or even as an environmentally destructive machine by others. These different interpretations can lead to varying outcomes of the same technology in different settings.
- Design flexibility: Just as technologies will have differing meanings to different social groups, there are also different means and views on how to construct technologies.

In other words, there exists design flexibility towards the selection and implementation of technical solutions.

- Closure: Given that there are different interpretations of the same technology, there is also bound to be conflicts around whose interpretation "wins." Closure is a concept proposed by SCOT to examine these processes of multiple interpretations, surrounding controversies, and how these can be resolved and understood. The closure state can be achieved through different mechanisms:

1. **Rhetorical Closure**: When social groups see the problem as being solved, they will begin to talk about the problem being solved. This is often found in advertising. For example, Bijker *et al.* (1987) used the example of bicycle advertisement to demonstrate this kind of closure:

"Bicyclist! Why risk your limbs and lives on high Machine when for road work a 40 or 42 inches Facile gives all the advantages of other, together with almost absolutely safe "
(Bijker *et al.* 1987, p.44)

Bijker *et al.* (1987) argued that the claim of being "almost absolutely safe" was a rhetorical move when it presented the problem of safety when riding in a high bicycle.

2. **Redefinition of the Problem:** Closure sometimes is achieved through a redefining of the problem. Bijker *et al.* (1987) used the example of the air tyre which was invented by Dunlop. To illustrate this, when it first came, engineers thought it was a monstrosity and public saw it as an awful accessory. Sport cyclists did not think that it was an anti vibration device because they used the high wheel bicycles. However, when it outpaced all rivals in racing, there was only astonishment left amongst the people. In this example, closure was reached when the sports cyclist and public started to respect the air tyre, but not by convincing people about the feasibility of it as an anti vibration device but based

on the problem of speed. In other words, a new problem concerning how to go as fast as possible was defined.

Closure is not permanent, and will evolve over time. New social groups will form and re-introduce new interpretative and design flexibilities contributing to the creation of new controversies and requiring redefined approaches of closure ((Bijker *et al.* 1987).

SCOT is an influential theory. Over time, it has been used to describe the importance of social alliances and control, social groups and their frames of meaning, the use of heuristics and various organizational issues (Nhampossa and Sahay 2005). In the field of information system research, several concepts such as relevant social groups, interpretative flexibility, and unintended consequences of technology have been developed and applied. Initial work with SCOT can arguably be said to have provided the basis for the subsequent application of Actor Network Theory and later the information infrastructure theory to study large networked and complex systems (Nhampossa and Sahay 2005). In the next section, I discuss how SCOT has been used in IS research by presenting some examples.

3.2 How SCOT has been applied in IS research

Various IS researchers, for example, Barley's analysis of the introduction of CT scanners at hospitals in the US, Orlikowski's ethnography of CASE tools (1992), Nhampossa and Sahay's (2005) study of software customization for health care in Mozambique and India, and Robey and Sahay's (1996) analysis of Geographic Information Systems (GIS) implementation in two counties in the US, have all drawn upon the SCOT approach for their analysis. These above cases are now discussed in more detail below so as to reflect upon how SCOT was drawn upon, and the associated analytical leverage the theory provides. Following this, I will turn to understanding some of the critiques of the SCOT approach.

Barley (1986) investigated the use of CT scanners in the radiology departments in two hospitals in Massachusetts, USA, one called suburban and the other urban. The radiology department in the suburban hospital started using the CT scanner technology by hiring 2 experienced technologists, who were then grouped with 2 inexperienced technologists and a newly hired experienced radiologist to bring the scanner online. The structuring process in the department was described by Barley to take place in two phases.

Phase 1: Negotiation of discretion

In the first phase of structuring at suburban, since there were no standard procedures, the interaction in the first weeks centered on clarifying roles and duties of and between the technologists and radiologists. Barley identified the following patterns of discussion between the two social groups: unsought validation; anticipatory questioning; and, referencing stating.

Unsought validation: The CT technologists frequently acted without inquiring with the experienced radiologists whether their actions were desirable. The radiologists commented, usually positively, about these actions.

Anticipatory questioning: This pattern was inferred when technologists inquired about the actions they perceived to be imminent.

Preference stating: This pattern concerned when radiologists stated their preferences for scan flow and operations. By outlining their grounds for preference, the radiologists established their credibility and competence while treating the technologists as if they deserved reasoned explanations.

Phase 2: Usurping autonomy

In the fourth week, the radiologists decided to rotate the CT duty on a weekly basis. This led to the generation of new scripts as the inexperienced radiologists worked with the experienced technologists: The following social interactions were observed:

Clandestine teaching – The technologists tried to teach radiologists without appearing to do so. This threatened the institutionalized roles of the two jobs.

Role reversal – The radiologists asked the technologists directly whether a scan evidences pathology. This generated anxiety for both the technologists and radiologists. Barley (1986) reported that one technologist "nervously" said of such an incident, "'*It's not my job to tell them how to do their job*'" (Barley 1986, p.93)

Blame the technologist – The radiologists put the mistakes around the machine problems on the incompetence of the technologists. Unaccustomed to having their knowledge perceived as inadequate, anxious that they might make a serious mistake and baffled by the computer technology, the radiologists began to express hostility toward the technologists.

Eventually, the technologists began to take responsibility for routine decisions without consulting the radiologists as in the past, while the radiologists started to withdraw from the operations around the scanner to save face.

The radiology department in the urban hospital mobilized to introduce the new scanner by hiring a young radiologist who had previously specialized in CT scanning. A second radiologist, who was a long-time member of the department, had followed the scanning literature but lacked the practical experience with the technology. To complete the staff set for operating the scanner, the department assembled a group of eight technologists from different areas. Later on, the experienced radiologists were assigned to work with the inexperienced technologists. The structuring process was described by Barley to take place in 4 phases:

- Negotiating dependence

  In this phase, radiologists appeared dominant and created a work environment that the technologists perceived as arbitrary.
- Constructing and ensuring ineptitude

  In this phase, the radiologists stayed in the office to break the dependence of the technologists. But after 6 weeks, the radiologist dropped the pretense of staying in the office and stayed in the control room.
- Towards independence

  After the sixteenth week, four technologists were permanently assigned to the head scanner, and the radiologists began regular rotation, so that the inexperienced ones could also begin to do scanning as well. Two different patterns of interactions were observed: technical consultation in which the new radiologists made no pretence of understanding technology, directly consulted with technologists, and mutual execution in which the radiologists consulted technologists for technical information, but handled the pathology side on their own. The technologists began to exercise more discretion and radiologists loosened their control over the day-to-day operations.

The results of the research showed that in the introduction of CT scanner, the structuring process of the organizations took place in the two hospitals in very different ways. However, "*technologies do influence organizational structures in orderly ways, but their influence depends on the specific historical process in which they are embedded*" (Barley 1986, p.107).

In summary, we can see how the different social contexts of the two hospitals led to very different outcomes related to the adoption of the CT scanners. As the principle of social causality states, the result of applying technology (either success or failure) was affected by the social context in which the technology was introduced to. Whereas in suburban, the inexperienced technologists worked with the experienced radiologists, in urban, the experienced technologists worked with the experienced radiologists. The different outcomes in the two radiology departments were seen to be due to differences in the structure and organization of the two departments, and also the complexity of the CT scanner technology itself. The interpretative flexibility of the CT scanner was also demonstrated. The technologists and radiologists were two different social groups who then interpreted the CT scanner technology in different ways. The situation prior to the introduction of the CT scanner technology was that the radiologists knew what the technologists knew, but not the reverse. In other words, most interactions followed the pattern where the radiologists gave orders, which the technologists followed. When the CT scanner was introduced, the radiologists still interpreted that they could master it as they did with the previous technologies but in fact they could not.

In the study of CASE tools told by Orlikowski (1992), the SCOT approach was used to reconstruct the concept of technology and proposed a model for investigating the relationship between technology and the organizations, referred to as the "duality of technology" or the Structurational Model of Technology. This conceptual model was used by Orlikowski to analyze the use of CASE tools in a large multi-national software consulting firm (anonymously called as Beta Corporation), having 13,000 consultants, 200 offices in 50 countries. Consultants in the company were grouped in to different temporary project groups. She analyzed in depth five project groups throughout different phases of software customization: analysis, design, etc. The introduction of CASE tool in Beta Corporation was described to take place in 3 phases:

- Initiating the technology

The technical consultant's actions were influenced by Beta's institutional properties – the existing institutionalized knowledge and norms of Beta's systems development methodology.

- Instituting the use of the technology

Tools were deployed in project teams. As a medium of human action, the tools were seen to both constrain as well as facilitate the activities of consultants. Once the tool use was taken-for-granted, they became institutionalized and their use influenced Beta's institutional structure.

- Ongoing interactions with technology

Most consultants in Beta had relatively low technical competence but some had the ability to modify their interactions with the tool, typically perceived to be the purview of the technical consultants – and through this they assumed some control over their task execution. As a result, they had forced a change in the technology which gave them a little more discretion in how and when they used the tools (Orlikowski, 1992). This analysis indicates how technology cannot be conceived as a fixed object at any stage during its deployment; its features and implementation patterns can and do change over time through human intervention (Orlikowski, 1992).

The SCOT principles of social causality and symmetry suggest that the social context should be analyzed and treated equally with the technology when looking for the explanation of the success or failure of technology. In the Beta case study, the technical competence of consultants in the different project groups varied from very good to very poor. Orlikowski (1992) emphasized this contextual difference as an important cause of the varying outcomes of different project groups, as important as the technology itself.

Also, the notion of interpretative flexibility helped Orlikowski to explain the varying reactions of the different users (consultants) during the development and use phases of technology. In the case of Beta Corporation, although the CASE tools could help to increase productivity and reduce manual work in some groups, in some other cases it did not work. In such cases, consultants with good technical competence could modify the tool but the ones with low technical competence came back to do the work manually.

In another study, Nhampossa and Sahay (2005) used the SCOT approach to describe a comparative case study between Mozambique and India in building the same health information software called DHIS (District Health Information System) in these two countries. Their analysis specifically focused on how the process of software customization was socially organized differently, leading to varying outcomes in the two countries.

Mozambique started attempting to customize and implement the DHIS in 1999 in selected pilot districts of three provinces (Braa and Hedberg 2002, Nhampossa and Sahay 2005). In Mozambique, the team comprised of university faculty (informatics and medical) who were responsible for both the project work and their individual PhDs, faced the challenge of trying to match the two quite different and demanding activities. The customization process included the translation of language (from English to Portuguese), defining datasets, data elements, and indicators, and migrating the legacy data (of 1997-2002) from prior existing systems into the DHIS.

India started the customization process of the DHIS in the state of Andhra Pradesh in December 2000. The team initially comprised one coordinator and 1 system developer which later increased to a total of 11 staff members. HISP India staff spent a long time (about 8 months) to make a situation analysis of the manual information flows before starting on the software customization process, which similar to Mozambique involved defining data sets, data elements, and making local reports. However, the team in India went further than the Mozambique team by adding other modules to the DHIS such as the web-view enabled module, and linking up with the mother and child health care module. The implementation process in India was financially supported by the state government, while the Mozambique implementation was primarily donor funded.

Although the attempt was to adopt the same technology (DHIS) in the two countries, the outcomes were very different. While India could develop additional modules to support the localization of DHIS2, Mozambique still struggled with very basic tasks such as of translation and data migration. And while HISP India got the support from the state government to roll out the system, HISP Mozambique till date has not obtained the acceptance from the government.

The different outcomes of the same software customization in the two countries were explained by the authors in terms of variations in social contexts. Specifically, they identified 4 features (summarized in the Table 4 below): the health sector; the organization of the team, the nature of the customization process; and, the installed base of the reporting systems.

Aspects of social context	Mozambique	India	Outcomes
Health sector	Dependence on foreign donors.	Largely state supported.	Customization process more under local control in India.
Organization of the team	Primarily doctoral and masters students from the university.	Paid staff working full time on implementation.	Paid staff responsible so fully engaged with the customization work while the students had their thesis work also to contend with.
Customization process	The primary aim was to duplicate the paper system.	Re-defining the paper system and adding more supportive modules.	Duplicating paper system made the customization harder, as there was the

			legacy systems to contend with.
Installed base	A legacy system called SysProg was in place – with strong technical and institutional legacies.	No prior existing system.	Without a strong legacy system to contend with, the Indians could focus more directly on the customization.

Table 4: Difference in social aspects of the context and their outcomes in the case study of Mozambique and India.

Based on this analysis of the influence of social context on the customization and implementation process, the authors concluded that: "*the process of software customization itself, which is extremely context dependent, can influence the content of the software*" (Nhampossa and Sahay 2005, p.2). The interpretative flexibility of the software in the two contexts was quite different, as in India success in the customization process was perceived by the implementation team to be a precondition for the continuation of the project, while in Mozambique working with the software was seen for the purposes of primarily supporting the respective PhD studies. These different interpretive flexibilities contributed to quite varying levels of engagement of the respective teams with the customization process and thus contributing to varying outcomes.

In another case study of GIS implementation in two counties in the United States, Robey and Sahay (1996) emphasized the role of the background of team members in shaping the consequences of GIS technology. The authors conducted a comparative case study between two county government organizations (called North and South county) that were in the process of implementing the same GIS technology (Arc/Info). In each organization, one technical support group and two user departments were included in to the research to make relevant comparisons. In each research site, the authors investigated in depth the social context, the relevant social groups and five organizational processes surrounding the GIS implementation: initiation, transition, deployment, spread of knowledge and consequences of technology. The summary of the analysis made by the authors is summarized in the Table 5 as follows:

Basis for comparison	North County	South County
Configuration	Organizational structure was unified, with decentralized management of information technology.	Organizational structure was differentiated with centralized management of information technology.
Capability	High capability to absorb GIS due to continuous experience with earlier versions of mapping systems.	Low capability to absorb GIS due to Office of Computer Services' traditional orientation as data processing department.
Initiation	GIS was positioned as an organization-wide response to shared needs. Initiating group was geographers with holistic view of organization and application systems.	GIS was positioned as revenue-adding service provided by Office of Computer Services. Initiating group was focused on marketing service to user departments.
Transition	Transition to GIS was smooth and continuous due to existing capability.	Transition to GIS was abrupt and discontinuous due to lack of prior capability with mapping systems.
Deployment	GIS was deployed in a distributed technical configuration supported by a central group. Users had primary responsibility for developing and maintaining applications.	GIS was deployed in a centralized technical configuration, controlled by the Office of Computer Services. Data processing personnel had responsibility for developing applications for users.
Spread of knowledge	Conceptual knowledge about GIS was spread among users through training conducted by the initiating group of geographers.	Procedural knowledge was conveyed to users through training conducted by the Office of Computer Services.
Consequences	GIS was widely acknowledged as responsible for transforming the conduct of work and the structure of North County's organization.	GIS was slowly implemented with little consequence for users. Minor increases in staffing in Office of Computer Services were attributed to GIS.

(Taken from Robey and Sahay 1996, p.275)

Table 5: Summary of the analysis in two counties

Based on the contrasts between two sites, the authors formed specific inferences in three areas:

- The relationship between structure and initiation, deployment and spread of knowledge

  Organizational structure was identified as a major point of contrast between the two sites. The differences in organizational structure were related to three aspects of the implementation process: initiation, deployment, and spread of knowledge. For example, the unified organizational structure enabled greater commonality in the social interpretations of the new technology and the creation of a single vision in one county (South) as contrasted with the other. Also, as a result, information could be and was more widely shared in the unified organization than in the differentiated structure.

- The relationship between capability and transition, deployment, and spread of knowledge

  The greater prior capacity in South county helped them to absorb the new technology and produce a smoother transition to the new systems. The distributed deployment of information technology in this county helped to increase the capability of the user community, whereas the centralized deployment in the North restricted user capabilities to develop. The capacity to understand technological innovations in the South was enhanced when the knowledge about them was spread amongst the user community, and knowledge spread faster since the user capability was high.

- Organizational consequences of GIS

  The organizational consequences of information technology were conditioned by the interplay between social interpretations, context and organizational processes that occurred during implementation. The processes of implementation were affected by the contextual conditions of the organizational configuration. And the organizational consequences were affected by the processes of implementation. Technology's organizational consequences were described to arise out of a complex interplay of the technological frames of different relevant social groups within the context and processes of implementation.

Robey and Sahay (1996) argued that the new GIS technology was subject to differing interpretations depending upon the social context and the process of implementation.

Examining how different social groups interpreted technology yielded them valuable insights for understanding the "actual patterns" of implementation. Each context is different, and it is expected that contextual elements (i.e. social groups, background, structure etc) interacted to produce different consequences. .

In the three cases described above, several aspects of social contexts which contributed to the varying outcomes of the technology initiatives were identified including team and organization structure, prior experience of team members, the nature of the application domain (sector), and the interplay between context and processes. In the Table 6 below, the key features of this SCOT inspired analysis is summarized. This sample analysis is a reflection of the extensive body of work in which SCOT has been used in IS research to deal with the issue of why "the same technology has different outcomes"

Features of the social context	Characteristics	Examples
Team structure and organization	Relationship between team members	Barley (1987) - the relationship between the radiologists and the technologists Robey and Sahay (1996) – cartographers and computer scientists had different familiarities with the GIS
History	Prior experience	Barley (1987) - the prior experience of the radiologists and the technologists in using CT scanners.
Capability	Team members' capabilities	Orlikowski (1992) – the variation of the technical competence of consultants in different project groups.
Team members' backgrounds	Prior background of team members	Nhampossa and Sahay (2005) – different backgrounds between Mozambique and

		India team Robey and Sahay (1996) – cartographers and computer scientists have different orientations to maps and GIS
Nature of the sector	Difference in social context of the sector	Nhampossa and Sahay (2005) – health sector in Mozambique was dominant by foreign donors, while in India the process was state supported.
Technology introduction process	Differences in the process of introducing technology	Nhampossa and Sahay (2005) – different processes were adopted in Mozambique and India. Mozambique team tried to duplicate the paper reporting system while India team first tried to standardize the reporting system.

Table 6: The key features of this SCOT inspired analysis

Although SCOT is an influential theory and has been widely used, it has also received various critiques from researchers. For example, Winner (1993) has emphasized the following limitations of SCOT:

- It ignores the consequences of technology. SCOT is a social theory and method to study how technologies arise and how they are shaped through social interactions, but not their consequences. According to Winner, the reasons for that were partly from the belief that the consequences of technological change had already been extensively studied by other researchers. Another reason was that the SCOT researchers regard technology as a less important field of inquiry.
- SCOT tends to forget the existence of the "irrelevant social group" (in contrast with relevant social group), which raises the annoying question for political pluralism: Who says what are the relevant social groups and their respective social interests? What about

those groups that have no voice but will be affected by the result of technological changes? According to Winner, SCOT offers an account of politics and society which is implicitly conservative, and the analytical lens through which SCOT models the relationship between social interests and technological innovation tends to conceal as much as they reveal.

- SCOT disregards that technological change involves dynamics beyond those revealed by studying the characteristics and actions of relevant social groups, such as deeper cultural, intellectual, or social origins of social choices about technology, and the autonomous properties of technology. Winner criticizes SCOT researchers to simply overlook aspects of philosophical discussion around the autonomy technology and that does not fit their conceptual straw man of technological determinism.

Despite these criticisms, which authors have tried to address in more recent studies, I argue that SCOT provides a potentially rich approach to study the social shaping of the processes of OSS development and implementation. In the discussions part of my thesis, I will also argue how I have tried to address some of these criticisms against SCOT. In the next section, I first provide an overview of OSS technology before discussing how I apply SCOT for its analysis in the specific context of my cases.

3.3 Relevance of SCOT to my analysis and proposed theoretical perspective

3.3.1 Relevance of SCOT to my analysis

Several reasons make SCOT an appropriate theoretical approach for analyzing OSS development and application in health care settings of developing countries. The thesis aims to see how the same technology is shaped in different social context. While the common assumption is that the same OSS can be effectively used in different social context and country setting, the reality is often not the case. The thesis then attempts to analyze how development and user processes are socially constructed differently in varying settings, and why this is the case. SCOT offers a set of tools to examine these questions.

The social context of health care sector varies significantly both within and between countries. Different aspects of social context relevant for my analysis include technological infrastructure, competence of development groups, and political support. Each of these issues is now discussed.

Technological infrastructure

Although classified as developing countries, the technological infrastructure between regions of a country and between countries can be very different. For example, Lagebo and Mekonnen (2005) in the comparative case study between Addis Ababa and Oromia region in Ethiopia argued that the infrastructure in Oromia region is very much poorer compared to Addis Ababa which made scaling difficult. Also, Nhampossa and Sahay (2005) described the difference in terms of infrastructure between India and Mozambique as follows:

"Indian rural areas may be more stable than Mozambique, for example, related to public transport which influences the transmission of reports from one facility to the other" (Nhampossa and Sahay 2005, p.7)

The difference in technological infrastructure can shape the outcomes of an OSS for health care application significantly. For example,

- With the health care sector, there are differences in the organization, levels and structure in different countries. Also, in different countries, there are different emphasis on particular diseases. For example, South Africa on HIV/AIDS, Mozambique on Malaria, India on Family Planning due to its big population problem etc.
- OSS is unlike other Microsoft based technologies. For example, it involves the use of new tools and frameworks such as Spring, Hibernate, Symfony[16], Mysql etc. And the competencies to use these tools are not well developed in different countries. Although countries like India may have more programming expertise, such expertise

[16] http://www.symfony-project.com/

may not be available in countries like Ethiopia or Mozambique. Also, Internet bandwidth, an important facility for OSS development, varies significantly across countries.

Competency of development groups

In the health care sector, the team for the software adaptation will comprise of people from different backgrounds. For example, Nhampossa and Sahay (2005) discuss this difference in the constitution of the two teams in Mozambique and India. One team consisted of merely researchers and students (Mozambique), the other team consisted of paid workers (India). Countries have different expertise in development groups. For example, expertise in Java technology can vary because of different education systems and different level of development of the IT industry. So India with a more developed IT industry, has relatively more Java developers as compared to other countries.

As OSS is a relatively new technology, it requires different kinds of development approaches and expertise. However, people in most of developing countries do not have such expertise. In some countries, the situation is worse than others.

Political support

The health care sector in different countries is constituted of varying political contexts. The nature and extent of donor involvement in particular can shape the level of support to the health sector. For example, Nhampossa and Sahay (2005) describe the differences in the level of donor-dependence between Mozambique and India as follows:

While both countries are dependent to a large extent on donor funds, in Mozambique where 80% of the health sector budget is comprised of external support, in India, the figure is much lower. This dependence shapes the nature of donor influence on HIS design and use.
(Nhampossa and Sahay 2005, p.7)

The policies of the governments in fostering OSS development and promoting public health are also different between countries. For example, in Vietnam, OSS has been politically stated to be the strategy forward thus making it easier to get commitment for OSS technology compared to Ethiopia where no stated strategy for OSS exists. These issues discussed above can potentially shape processes of both OSS development and use. SCOT potentially provides a powerful tool to make an analysis of how different aspects of the social context can shape this development and use processes.

In addition to developing theoretical insights, SCOT can also help to potentially develop various practical implications. For example, the principle of interpretative flexibility of SCOT can guide a manager of a OSS project to understand that there are different stakeholders in a project, and it should not be assumed that all of them share the same interpretations, and these divergent meanings can shape the implementation processes in trajectories that were unintended. SCOT also emphasizes that these interpretations are shaped through historical processes and the socio-cultural context of the social group, and are thus complex and time consuming to change. Just having one time training sessions may not be adequate, and instead the efforts should be on longer term education processes. A health manager may not see OSS as a superior technical artifact as compared to proprietary software like a technologist may, but instead will focus on how the tool helps him to develop a better analysis of the region which he is responsible for. A health manager thus has to try and balance these divergent interpretations, find reasonable closures, so that the potential negative consequences arising out of these divergences may be avoided.

My proposed perspective is now presented.

3.3.2 Proposed theoretical perspective

Based on the literature review presented and the Table 6, the following conditions of social context have been identified by SCOT research to be relevant in shaping IS development and use:

- Team structure and organization
- History
- Capability
- Team members' backgrounds

- Nature of the sector
- Technology introduction process

The there aspects which are history, capability and background are relatively related, so I grouped them into one aspect namely technical capacity. The Figure 1 visualizes these conditions in relation to the social context:

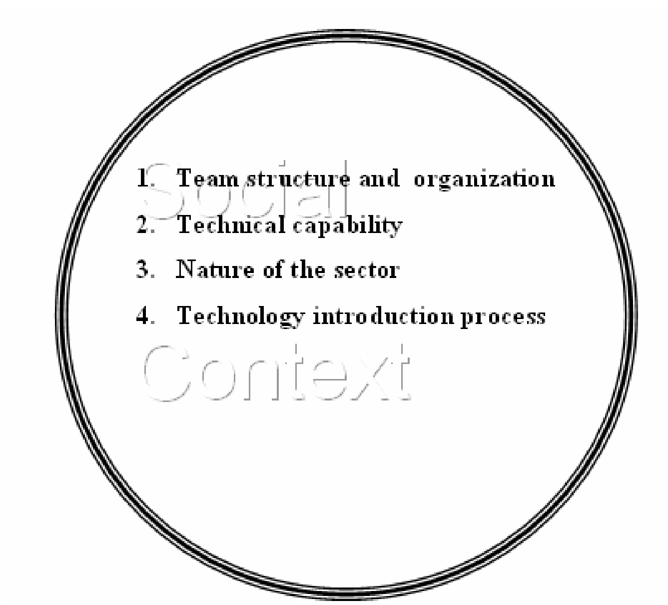


Figure 1: Social conditions shaping the IS development and use identified by SCOT research

This framework will serve as the starting point for analysis. I will find out whether these conditions of social context are matched with the empirical work and are there other conditions which also influence the OSS development and use for health care sector. Therefore, I propose a theoretical perspective based on SCOT research for my analysis process in the Figure 2:

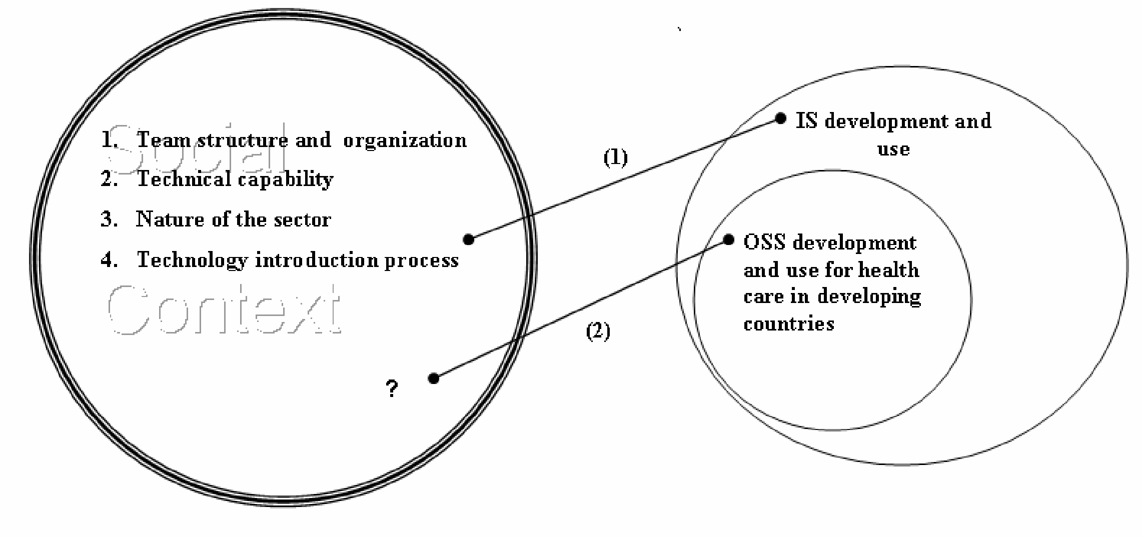


Figure 2: The proposed theoretical perspective for the analysis

The figure demonstrates the relationship between social contexts and IS development and use, presented in the line (1). In fact, OSS for health care in developing countries is a subset of IS, presented by a smaller circle inside. The analysis's aims to explore the line (2) to find out whether the social conditions in IS in general are matched with OSS for health care and additional conditions which may influence OSS for health care in particular.

CHAPTER 4: RESEARCH APPROACH AND METHODS

In this chapter, I present the research approach adopted and details of the settings, methods of data collections, and analysis. Comparative case studies were used to understand the difference between the research settings of Ethiopia and Vietnam and also to analyze different outcomes of applying OSS in these two countries. The study is conducted within an action research framework where, in addition to understanding, the aim was to also influence the outcomes, primarily through the process of software development and capacity building. This chapter is organized as follows:

In the first section, I briefly describe the two research teams in Ethiopia and Vietnam. The second section presents the research approach, while section 4.3 describes the data collection methods. Section 4.4 discusses how data was analyzed and interpreted. The last two sections discuss the ethical considerations and the limitations of the research respectively.

4.1 The research teams

This research was conducted in the two countries of Ethiopia and Vietnam during the period from February 2005 to December 2006. The author of this thesis collaborated with a larger team in both countries, and also with the development team in Oslo where the core development of DHIS2.0 was being carried out.

Ethiopia

The team in Ethiopia consisted of the author of this thesis and four Masters students from the University of Oslo who were originally from Ethiopia, and one Ethiopian doctor who was taking a second degree in Informatics in Addis Ababa University. This group was in turn subdivided into two teams. One sub team worked on DHIS2 customization, comprising of two Ethiopian Masters students, and the other team worked on the patient based system, comprising of the doctor and the two other Masters students.

Vietnam

During the period from January 2005 – August 2005, the team in Vietnam consisted of the author of this thesis and 4 interns from ComA Company. From January 2006 – January 2007, the 4 interns were replaced by 3 paid developers, supported by the Oslo HISP project. The Table 7 describes the types of work carried out by each of the teams:

Type of systems	Description of tasks	Ethiopia	Vietnam
The patient-based system	The term "Patient based system" in this thesis refers to a set of systems including: The ART module The flexible system called OpenEPR (Open Electronic Patient Record) The Symfony-OpenMRS based flexible patient system	Developed and implemented the patient based system for HIV/AIDS management and later ported the system to the Symfony framework and made it more flexible and shareable (The ART module and The Symfony-OpenMRS based flexible patient system).	Expand the patient based system to be more generic so that it can support all different requirements of Patient based systems without interfering with the source code (The flexible system called OpenEPR).
DHIS 2	DHIS2 core functionalities are actually developed by the Oslo team. In Ethiopia and Vietnam, the two local teams adapted it for local requirements by defining the database, developing the report and some other additional modules.	Adapt DHIS2 for local use. Work included preparing the local database, developing local reports and a special Morbidity and Mortality module.	Adapt DHIS2 for local use. Work included preparing the local database, and developing local reports.

Table 7: Summary of work done by the two teams

The time frame of the research

The time frame for this study lasted from early 2005 to the end of 2006 mainly in the two countries Ethiopia and Vietnam. From early 2005 to August 2005, I was also involved in the implementation of DHIS1.3 (an earlier version of DHIS2.0) in Ho Chi Minh City (Vietnam) and the development of DHIS2 to some extent. In this period, I worked with a Vietnamese team comprised of four students, who did their internships in a company which had a collaboration with HISP, and several other visiting developers from Oslo (Norway). This team customized the DHIS1.3 for Ho Chi Minh City by defining datasets, data elements and local reports, and also joined in the global development of DHIS version 2.0. The visiting developers from Oslo helped in orienting me to the new technologies used in DHIS2.0.

In August 2005, I went to Oslo for the first semester of the International Masters program. In Oslo, I worked on DHIS2 development with the core team in Oslo. I was assigned to work on web GIS module with two other Masters students who took the open source course. I also joined in technical meetings with the core team to discuss the functional design of the system, and contributed by doing some development.

In January 2006, I finished my first semester of the Masters course in Oslo. A Professor in the Department of Informatics, also a coordinator of HISP global and HISP India, suggested that I should go to India to get more exposure to the problem domain. During this two-week trip I had chance to meet most key members of HISP India. These face-to-face meetings helped to build collaboration between me personally and HISP India. I also gave introductory training on using Webwork[17] (a framework used to build MVC – Model View Controller - architecture of DHIS2) for developers of HISP India team.

Then I went to Ethiopia in February 2006 to study my second semester of the Masters program. In a meeting to orient the field work of Masters students, two groups of students were formed to work on DHIS2 customization and the patient based system. I was assigned to work with both the teams, specifically to provide technical support. During my period in Ethiopia, I was extensively involved in software development with both the local teams. The

[17] http://www.opensymphony.com/webwork/

link between the Ethiopian team and me continued even after I left Ethiopia in April 2006, mostly through electronic means.

I came back to Vietnam in April 2006 and joined the efforts of the local team in Vietnam to customize DHIS2 for Ho Chi Minh City. The same time, I continued developing the flexible patient based system in close collaboration with the team in Ethiopia.

The research time frame is summarized in the Figure 3:

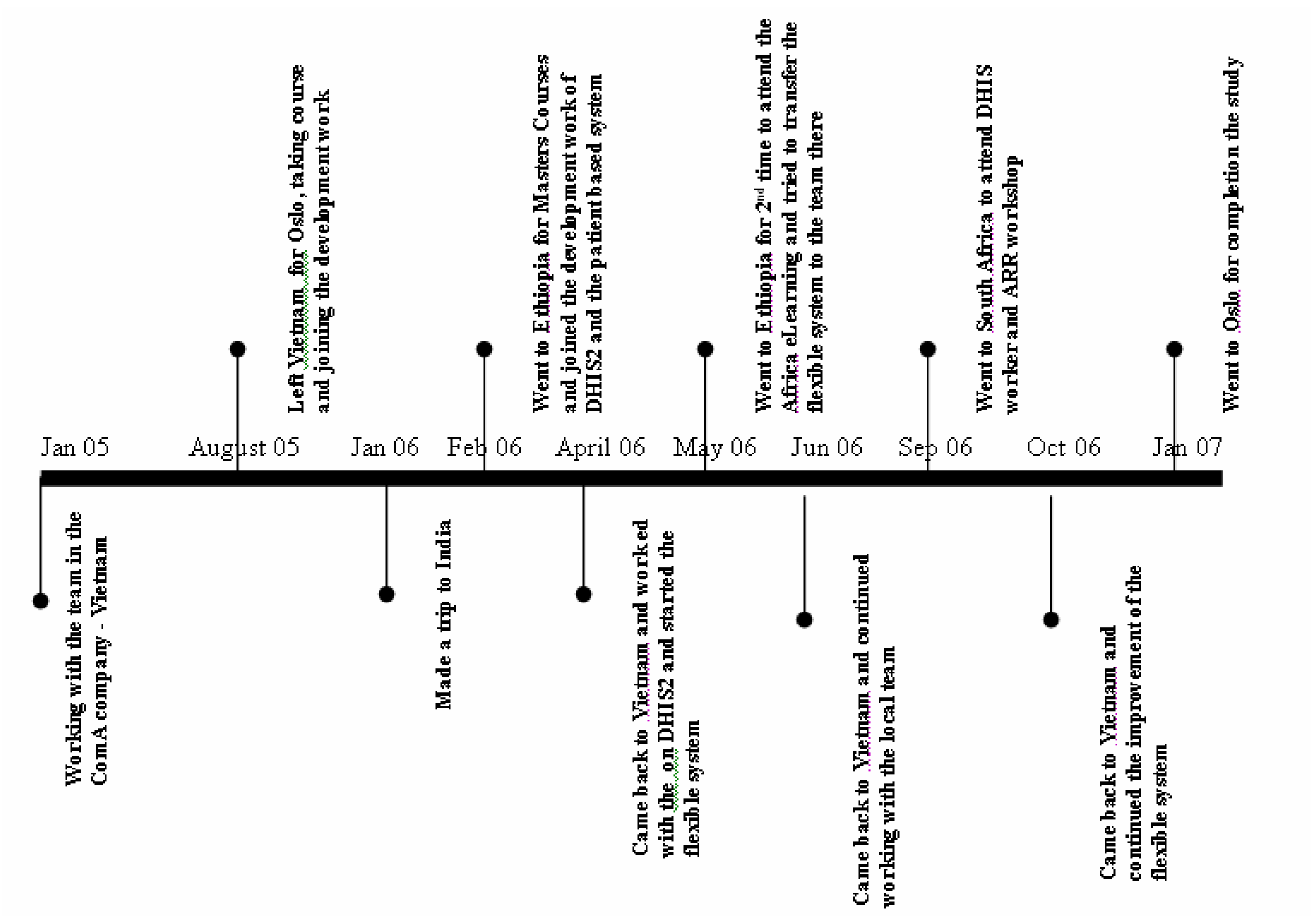


Figure 3: The research time frame

In this section, I have discussed the research setting and provided and overview of the time frame for my different fieldwork activities. In the next section, I discuss the research approach employed in this thesis.

4.2 Research Approach

This research is situated within an action research framework as I aimed to understand how the differences in societal contexts affects the outcome of applying the same technology and practically contribute to improving the development and use processes. As the aim of the thesis was to understand how social settings influence outcomes of technologies, qualitative research methods were employed. These methods are designed to help researchers understand people and the social and cultural contexts within which they live. The goal of understanding a phenomenon from the point of view of the participants and its particular social and institutional context is largely lost when textual data are quantified (Myers 1997). This research has two research objectives: *a) to understand why the same technologies (OSS for health care) produced different outcomes in different settings (two developing countries) b) to explore the approaches to help organize the process of OSS development and implementation more effectively.* Qualitative research methods were thus used to understand the problem of "the same technology, different outcomes" (Barley 1986, Orlikowski 1992, Robey and Sahay 1996). .

My empirical work was conducted within an interpretive research approach as the aim was to understand the phenomena through the meanings that people assigned to them. In IS research, interpretive methods have been used by researchers to produce an understanding in the context of the information system, and whereby the process it influences and is influenced by the context. For example, Hanseth *et al.* (2006) present a study of the development of a standard EPR system and its implementation in the National Hospital in Oslo, Norway. The research began with a focus on the implementation process, and in particular, on understanding <u>how</u> existing information systems and work practices influenced the implementation process, <u>how</u> difficulties emerged, what were the sources of such difficulties, and what roles the various actors involved in the process played. In my study, the aim was to understand how differences in social contexts shape varying outcomes, and how OSS for health care in developing countries can be organized better. The focus on understanding the social context and the objectives in answering the "How questions" makes the interpretive approach an appropriate base for my study.

Action research framework:

As is the tradition within the HISP framework, where the focus is on making improvements in HIS related systems, an action research approach framework was employed. Action research as a research approach originated in socio-psychological studies of social and work life issues, and has been argued to be well suited to the study of technology in its context (Baskerville and Wood-Harper, 2002). Action research is often uniquely identified by its dual goal of both improving the organization and at the same time generating new knowledge – both theoretical and practical. Accomplishing these goals requires the active collaboration of the researcher and client, and thus it stresses the importance of co-learning as a primary aspect of the research process (Baskerville and Wood-Harper, 2002).

Action research was carried out through cyclical processes of problem definition, action taking, and reflecting on the outcomes that were generated. This cyclical process, I argue, was fundamental in creating the knowledge on which this thesis is based, and also in making practical improvements on the ground. Baskerville and Wood-Harper (2002) describe these cyclical processes in the following five identifiable phases:

- diagnosing
- action planning
- action taking
- evaluation
- specifying learning.

I adapted this cyclical process as depicted in the Figure 4:

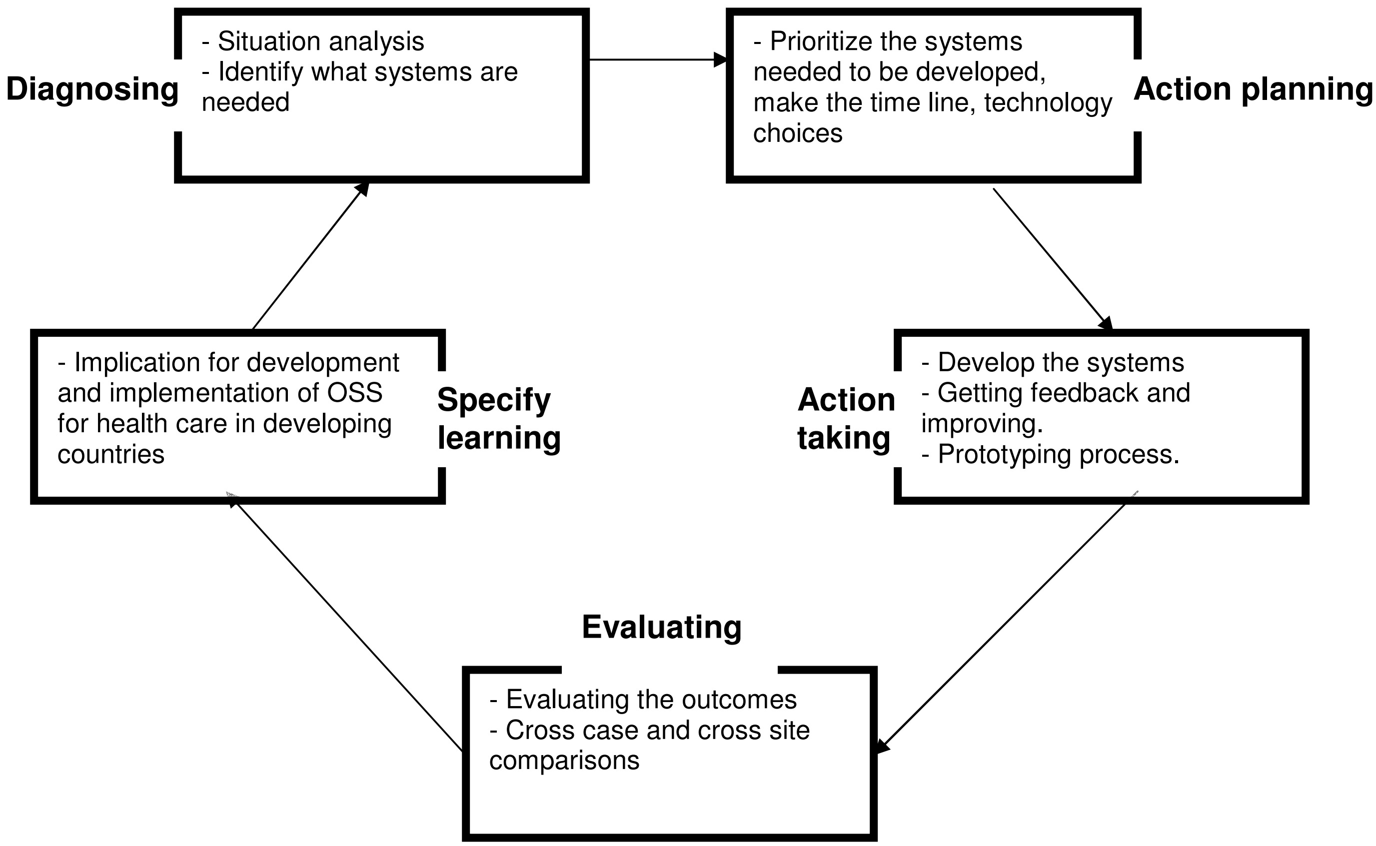


Figure 4: Action research cycle

Comparative case study

A case study is an in-depth exploration of one situation (Cornford and Smithson 1996), which allows the assessment of a phenomenon in its natural setting (Benbasat *et al.*. 1987) and provides an opportunity to directly observe the unfolding events over time (Walsham 1993). Yin (2002) defines a case study as an empirical inquiry that investigates a contemporary phenomenon within its real-life context, especially when the boundaries between phenomenon and context are not clearly evident.

Case study research is a commonly used qualitative method (Orlikowski and Baroudi, 1991; Alavi and Carlson, 1992) because it is particularly well-suited to IS research, since the object of the discipline is the study of information systems in organizations, and interest has shifted to organizational rather than technical issues (Benbasat *et al.*. 1987, Myers 1997).

Yin (1981) suggests the use of single-case studies where the cases are revelatory, critical or unique, while a comparative case study design is a powerful way to study the same phenomenon in different settings, and to analyze how different contextual conditions shape the phenomenon. In my study, one of the aims is to understand how differences in social context affect outcomes around OSS development and use in the health care setting of two developing countries. Although both countries examined in my study are classified as developing countries, the difference in technological infrastructure, technical capacity and culture are quite significant and thus contribute to varying outcomes. This thesis did the comparative case study on two systems (DHIS2 and Patient based system) in two developing countries (Ethiopia and Vietnam). I made comparisons in two ways: across systems and across countries. According to that, the first two comparisons were made between two countries, one in DHIS2 and the other across the patient based system. Within Ethiopia, there was another comparison made between DHIS2 and the Patient based system, and similarly in Vietnam. The Table 8 describes the research analysis design:

	DHIS2		**Patient based system**
Ethiopia	**1** Comparison of DHIS2 between Ethiopia and Vietnam	**3** Comparison between two systems in Ethiopia	**2** Comparison of Patient based system between Ethiopia and Vietnam
Vietnam		**4** Comparison between two systems in Vietnam	

Table 8: Cross-system and cross country comparisons

In this section, I have briefly discussed the research approach and research design employed in my study. In the next section, I discuss data collection methods used in this study.

4.3 Data Collection Methods

To conduct this study, I used qualitative research methods including interviews, observation, and document analysis (Myers and Avison, 2002) to collect data. These methods are now discussed as follows:

4.3.1 Observation

Observations are used in fieldwork for gathering information on the behavior and activities of individuals in the research site (Creswell, 2003). In this study, three types of observations were employed:

- **Participant observation (team work):**

The real working interaction among members in the two teams was a key source of empirical data. Working closely with the team in the two countries gave me good opportunities to observe thoroughly the process of OSS development in the two settings. Facts and events were recorded by taking notes everyday in note books and these were later summarized in electronic format at least on a weekly basis. The contents of these notes covered information about project progress, reaction and ideas of the team members, difficulties the team faced and the workarounds they employed. For example, the notes which were recorded on March 25th in Ethiopia were shown in the Figure 5:

March 25th 2006, Addis Ababa

ART team

- *The ART team could design forms and make table in the database.*
- *They preferred to learn a lot of programming techniques than every small piece and put them into practice.*
- *Convinced them that they should be patient and do learning in the same time with doing real development work.*

DHIS2 team

- *Was struggling with design and making reports*
- *Integrating JSP based reports into DHIS2 created difficulties for them as it was so complicated (5 different steps they had to follow)*

Figure 5: Field notes in Ethiopia

Such participation helped me to gain an overall understanding of team behavior, their communication styles, the difficulties and the challenges they faced in the development and implementation processes. There were also slight differences between the observations in Ethiopia and Vietnam. In Ethiopia, due to the low technical capacity of the team, support was given more frequently and more detailed, leading to the more detailed observations as compared to Vietnam.

- **Observations during field visits:**

Observations were made during field visits during this study to get understandings of the nature of the health care sector in the research sites, the work related to data collection of health workers, and the responses of the health workers after the software was installed. Observations were done in Ethiopia by visiting the ART clinics in Addis Ababa to understand the paper based system and to train the end-users on how to use the ART system. In Vietnam, several observations were done by field visits to the ward clinic, the district health center and the Health Services in Ho Chi Minh City. Details of the number of field visits are summarized in the Table 9:

Visit to	Number of observation	Duration
ART clinic in Addis Ababa – Ethiopia	2	1 hour each
Ward clinics in Ho Chi Minh city – Vietnam	3	1 hour each
District health center in Ho Chi Minh city - Vietnam	2	1 hour each
Hospital	1	45 minutes each
Mother and Child Health Care center in Ho Chi Minh city – Vietnam	2	2 hours each

Table 9: Summary of the observations done during the research

In Ethiopia, the field visits were made very rarely as I felt the presence of a foreigner in the ART clinics may be seen as negative, due to issues relating to sensitivity of HIV/AIDS data. Even in the two visits I made, I only had a general discussion with health workers; and health managers did not look at the registration book or other related documents. However, in Vietnam, I had more freedom to do such observations in the field visits. I could look at the documents, take photographs, and question the health workers about the meaning of numbers.

4.3.2 Observation in presentations and meetings:

Besides observations during the field visits to the health facilities, other observations were made in meetings and presentations. In Ethiopia, I frequently met with the team to discuss problems related to the project. The meetings were held wherever it was most convenient. For example, sometimes we had meetings in the project office in Addis Ababa University and some times in class rooms. In addition, we often had informal meetings when we went for coffee in the university's canteen or in the restaurant. In the meetings, we discussed the strategy/plan for implementation and development of DHIS2 and the patient based system.

In the Africa eLearning conference, organized in Addis Ababa in May 2006, the Ethiopian team and I gave several presentations about the work we had done. I also had the opportunity to attend the DHIS2 workshop organized in South Africa in September 2006. The meetings and discussions with HIS developers and health managers were very useful for me to understand the project domain from a broader perspective. Feedback and discussion during such presentations and meetings became a very valuable source of data for the project work and also for my thesis. With respect to the project work, I received various comments on how to improve the system, and as a result many design improvements were made. For example, in the DHIS2 workshop in South Africa, after I gave the presentation of the openEPR system, one DHIS developer suggested that the entry forms should have real form layouts rather than row-by-row layouts. Based on that comment, I redesigned the system so that it could generate data entry forms with grid layouts.

The Table 10 summarizes the number of meeting and presentation carried out during this research:

Type of meeting and presentation	Quantity
Regular and irregular meetings with the team in Ethiopia	7
Regular and irregular meetings with the team in Vietnam	12
Presentations in HIS workshops (Africa e-Learning in Addis Ababa – Ethiopia – May 2006, International DHIS in Cape Town and East London – South Africa – September 2006, HIS workshop in Pretoria – South Africa – October 2006)	6

Table 10: Summary of meetings and presentations done during the research

4.3.3 Interviews

Interviewing is one of the most common and powerful ways to understand human being since it gives room to both the interviewer and interviewee to clarify opinions and points of view through interactions (Creswell, 2003). Interviews were done with members of the core team in Oslo, Ethiopia and Vietnam to find out issues around the development process and the technologies used in the project. Several interviews were also done with health managers in Vietnam to understand their needs for health information systems and find out the nature of the health sector contexts and their perceptions towards OSS for health care. Interviews made with coordinators of DHIS2 development in Oslo was to find out the initial phase of the development process and to understand how technologies and frameworks used in DHIS2 were selected.

Every interview lasted approximately one hour. Questions used in the interviews varied depending on the respondents. For example, questions given to the Oslo team members related to issues around how technological choices were made, and specifics about the development process. Questions to the health managers in Ho Chi Minh city were around their perceptions in comparing DHIS2 and DHIS1.3, their views toward OSS, and their perceived needs for a patient based system.
The Table 11 summarizes interviews conducted in this research:

	Number of interviews
The core team in Oslo	4
The team in Ethiopia	2
The team in Vietnam	3
Health manager in Ho Chi Minh city – Vietnam	2
DHIS2 development coordinator	1

Table 11: Summary of interviews done

4.3.4 Document analysis

Document analysis involves making sense out of the text and image data. It helps to get useful information related to the researcher's interest and supplement other forms of data collected (Creswell, 2003). Document analysis played an important role in my study, to gain additional information on various issues such as the nature of the health sector, the DHIS2 development, and other patient based projects. Different types of documents studied are described in the Table 9:

Type of documents	Relevant to my study
Mailing list archive and other email discussion	• Understand the ongoing global efforts in making DHIS2 and patient based system • Understand the perspectives of the two teams in system development • Understand the challenges and difficulties the different teams faced during the development and implementation process. For example, the developer mailing list is the place to post questions and answers related to technical issues.
HISP project website	Getting overview over design and architecture of DHIS2 system
Documents on health care sector, paper forms, reports, health care software documents	Getting an understanding of contexts of health care sectors in Ethiopia and Vietnam
Other project websites of patient based system such as OpenMRS, Care2x	Getting an understanding of other similar OSSs for health care

Table 12 Summary of types of documents studied

The two main kinds of documents which are email discussion and HISP project wiki, are now discussed.

Mailing list and other email discussion

During the research, I used the DHIS2 developer mailing list very frequently to collect data. This mailing list was a huge source of data because in a short period of time (less than 1 year, more than 3000 mails were sent). The discussion in the mailing list reflects the progress of the project over time. I also drew heavily from various texts in the mailing list as sources in my writing. This mailing list provided rich data that can be sorted by time and subject, helping to understand the evolution of HISP project. I created a folder of DHIS2 emails in a mail box for archive searching. For example, the following mail reflected the challenge of the team in Vietnam in developing local report modules for child health program, sent by a developer in Vietnam to the list, as showed in the Figure 6:

"Yes, data elements are input into Monthly report should not input in Quarterly report at same Organization unit, they should be aggregated. To create datasets as Mr. A said is okay and easy for our work, but how can we explain for users? They will be confused when go to Data Entry window to input data for Quarter and see only 20 data elements when actually is 30. Are there any ways to show all data elements in Data Entry form but the data elements aggregated from month cannot be entered value?

Another problem is, in Quarterly report there are two types of data, some are routine data; some are semi-permanent. But when create dataset we can choose only one, routine or semi. Then I have to create two datasets. This will trouble users because to input data for one report they have to do a lot of action" (DHIS2 mailing list, 2006)

Figure 6: One email discussion related to the DHIS2 in Vietnam

This challenge occurred due to the mismatch between the standard design of DHIS2 and the local reporting system in Vietnam, which did not separate routine data and semi permanent data, creating difficulties for using DHIS2.

In relation to the patient based system development, there was no mailing list setup but a list of email addresses separated by comma was used instead. Email addresses of those involved in the discussion were put in the To or CC section of the email we sent. And in replying, we used the function of Reply to All. This email archive became useful for me in doing this study because it contained valuable information about the development of patient based system. For example, the following email discussed the progress and the future plan of the patient based system after the first prototype of ART was released, sent by a ART team member to a group of people having concerns in the Patient based system within HISP, as showed in the Figure 7:

> *"today, the system (version 1.0) prototype is deployed in one ART clinic A, B (with fantastic support and effort from C) have made it happen lots of plan follow, about functional expansion (PMTCT, VCT etc) and also to other facilities (Oromia region also wants), so also Pretoria. Another student of Eric is coming to India, and we will be taking this system and customizing for Kerala...but also, try to generalize the logic to enable customization for other patient specific follow up diseases like TB .is going to be linked with dhis2 and also GIS the system has been built on care2x, and A, B and C will give you more details on accessing the set up file, and the process of customization"*
> (Group email discussion, 2006)

Figure 7: One email discussion related to ART module

Email and mailing list discussion reflects a diverse picture of activities taken place within the project. The richness of data provided by these email archives makes email and mailing list discussion become an important data collection method in this study.

HISP project wiki

The HISP official website (http://hisp.info) was another source of data used in my thesis. This website contains information about the project such as the modular architecture of DHIS2 software, road map and milestones of software releases, as well as technical explanations.

This website is also used to provide information of other versions of DHIS such as .1.3 and 1.4. There are also several parts of the site providing implementation information. For example, the implementation of DHIS1.4 in Nigeria and DHIS2 in Vietnam are reported in this site.

4.3.5 Prototyping

As one of the aims of this research was to develop a Patient based system and customize DHIS2 for implementation in Ethiopia and Vietnam, prototyping was one important data collection method. Prototying approach in close interaction with users gave a rich source of data for analysis. Through the prototyping approach, the cycle of "build-show-feedback-revise" was repeated many times until the users felt satisfied with the system. In the case of the Patient based system, prototyping was used in developing the ART module and the flexible system. For example, in Ethiopia, the ART module was brought into the ART clinics every week to demonstrate its functionalities to the users, also other improvements based on the previous feedback. In the DHIS2 case, the team in Vietnam also installed the software in several districts to get feedback from users before rolling it out to the whole city.

4.3.6 Electronic communication means

Besides above-described methods, other electronic communication means (i.e. chat, voice telephone etc) were also used in my research to collect empirical data. I used Yahoo Messenger[18] for text chatting and Skype[19] for both text and voice chatting, depending upon my counterpart's preference. For example, this method was used mostly to get in touch with the team in Ethiopia after I left the country in April 2006. By using chat, I could receive immediate responses from my counterparts without delay as compared to the use of emails.

[18] http://messenger.yahoo.com/
[19] http://www.skype.net

For example, the following discussion was extracted from one of discussion between the Ethiopian team member and me, as showed in the Figure 8:

> *A: Hi there!!!*
> *thanhnew2001: hi*
> *thanhnew2001: u have problem with dhis?*
> *A: how did you suspect???? yes*
> *A: are you using M6?*
> *thanhnew2001: yes*
> *A: have you got any problem like "cannot connect with underlying database" while running jetty?*
> *thanhnew2001: never*
> *thanhnew2001: please check first the hibernate.properties*
> *A: yeah it's there with create-drop property and it's not working*
> *thanhnew2001: there are 3 places to keep that file: USER_HOME, web-portal*
> *A: USER HOME*
> *thanhnew2001: the file in USER_HOME will be checked first, and then web-portal*
>
> (Instant Message Chat, 2006)

Figure 8: Communication via Instant Message Chat

4.4 Data Analysis

The method for analyzing the interview data was conceived as a process of continuing refinement, moving from the raw transcribed text toward more general theoretical inferences (Robey and Sahay, 1996). The fist step of the analysis process was to split the empirical data into coded categories. After this, the split data with similar or related meanings was grouped to form themes. Based on such themes, the theoretical inferences were derived. These steps of data analysis are discussed below in greater detail:

Data splitting

As I planed to do a comparative case study on two systems in two countries, four categories were formed to assign empirical data into them:

- DHIS2 in Vietnam
- DHIS2 in Ethiopia

- The patient based system in Vietnam
- The patient based system in Ethiopia

Themes grouping

All aspects around social-technical conditions in the two research sites were considered to identify the ones which shaped the outcomes. For a systematic analysis, these aspects were grouped into initial themes, which are constructed based on literature about SCOT (Barley 1986, Orlikowski 1992, Robey and Sahay 1996). These themes served as the point of departure for the analyzing process, listed as follows:

- **Team structure and organization**

  This refers to the structure of the team and the organization of the development process. For example, whether the team members were student or paid worker, their background (Medical or Informatics), the decision on what technologies would be used, and how to distribute the development.

- **Technical capacity (history, capability, team members' backgrounds)**

  This refers to the technical competence of the team in dealing with OSS technologies such as whether they had developed software before, their experience on system and database design, and the use of programming languages.

- **Nature of the sector**

  This refers to the characteristics of the health care and OSS sector, such as the involvement of donors in national health structure, the stated policy of government on OSS etc.

- **Technology introduction process**

  This refers to how the technology was introduced, the process of adaptation, and the approach to deal with the technology.

While these above themes based on literature served as starting points for the analysis, they were redefined based on the empirical analysis.

Inter-sites and inter software comparisons

After completing the theme grouping process, inter-sites comparisons were made to identify the similarities and differences between two research sites. The results of this process helped to examine the interrelation between the social context and the outcomes of the two sites, and how these relate to issues identified in the literature.

Drawing theoretical inference

Based on the results of the comparisons, several theoretical inferences were drawn upon, which served to answer the two research questions. The process is schematically depicted in the Figure 9 below:

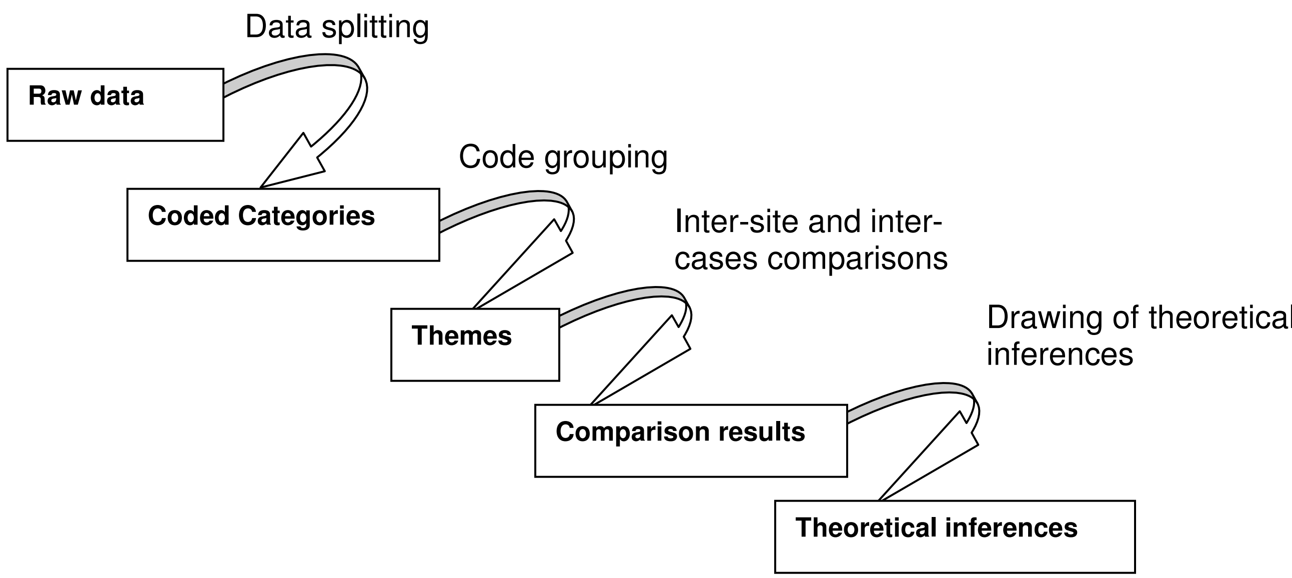


Figure 9: The process of data analysis

4.5 Limitations of the study

Some of the limitations of the study identified were:

- First, concerns the selection of the case. Due to time constraints, I did not include the case of India which can be described as a successful case in applying DHIS2. The comparison in the DHIS2 system between Ethiopia and Vietnam did not reflect the great difference in outcomes in applying the same technology as the Indian case would have been.

- With respect to the patient based system team, I was the only developer which impeded the development and implementation processes.
- The time line of the research was short, and outcomes which take long to develop, could not be easily ascertained. So, tendencies are discussed, rather than concrete changes.

4.6 Ethical considerations

The permission to get access to the field in Vietnam was granted to me by the Ho Chi Minh City Health Service since I had worked as a DHIS facilitator of the HISP project in 2004. At that time, I was responsible for the implementation of DHIS1.3 in Ho Chi Minh city and Thua Thien Hue province. In Ethiopia, I first came as a Masters student to take course in the Masters program. However, I got involved with the HISP team as a technical resource person. I thus was not directly involved with field level issues, but worked more with the other students. In my thesis, I did not use actual names of the respondents, thus respecting their confidentiality.

In this chapter, I have discussed several issues related to the research approach and methods such as research design, the time frame, the data collection and analysis methods, and research limitations. In next chapter, I describe the research setting, which will be followed by the case description.

CHAPTER 5: THE RESEARCH SETTING

This chapter is structured as follows. The first section discusses the background of HISP (Health Information System Program) project and related several systems developed within this project. Section 2 and 3 discuss the research settings in Ethiopia and Vietnam respectively.

5.1 HISP and the HISP software

5.1.1 HISP

This research is designed in the framework of the HISP which aims to design, develop and implement free and open source application to contribute to building "*sustainable health IS in developing countries*" (http://hisp.info). HISP was initiated in South Africa in 1994, and has slowly expanded and is now running in a number of countries such as Ethiopia, Malawi, Tanzania and Mozambique in Africa, and Mongolia, India and Vietnam in Asia. The aim of this thesis is to empirically establish “networks of action and support” between two of the nodes in the South – Vietnam and Ethiopia – and also to establish the enabling role of Oslo in the reality. This network seeks to enable the design, development and implementation of the FOSS called District Health Information Software (DHIS) and another patient based system.

5.1.2 The HISP software

This section provides explanations to two systems developed within the HISP project DHIS and DHIS version 2.0, and there patient based system called ART, IHAMS and the flexible system.

- **DHIS:**

DHIS is the software for collecting data (routine data, population data, and infrastructure data) from health centers, hospitals, clinics etc. This can be grouped into two broad categories: routine and semi-permanent.

Routine data are data about the delivery of health services. For example, number of women giving birth in a certain period (month, quarter etc), or number of children fully immunized. The health workers collect these data by counting patient by patient and aggregate by period and geography. For example, the number of women giving birth in January, February and March can be aggregated to the number of women giving birth in the 1st quarter. This is the period aggregation. If there are two hospitals in a certain district, the number of women giving birth in those two hospitals individually can be aggregated to the number of women giving birth in that district. In short, routine data are numbers which are aggregated by both period and geography to be used for planning and management.

Semi-permanent data contains numbers which do not change on a monthly basis. The frequency of this change could be 6 months or a year. Semi permanent data usually refers to population data and infrastructure data. For example, the number of doctors in a clinic is semi-permanent data. Like routine data, semi-permanent data can also be aggregated by geography, but not by period. For example, the number of doctors in a certain hospital in January, February and March is not aggregated because it does not mean anything. This is an important feature to differentiate routine and semi permanent data.

In many countries, similar systems to DHIS are widely available. For example, in Mozambique, there is a system developed by the Ministry of Health called SysProg. In Vietnam, there is a system called HMIS. However, DHIS allows end-users to define elements and datasets as per their needs. This "flexible" feature of DHIS2 makes it become a system which can be adapted to any contexts and any reporting systems. However, implementing it requires engaging with political brokering.

- **DHIS2**

DHIS2 is software developed by the HISP project. It took one year to select the technologies; another year for technology learning and a third year for actual development. DHIS2 is not built from scratch, and is the newer version of DHIS1.4 and DHIS1.3. DHIS2 seeks to meet the two following requirements:

- Making a web-based, open source version of DHIS1.4; and
- Distributing the development in the HISP network

- **The Patient based systems (Care2x, OpenMRS, ART, IHAMS and the flexible system)**

In contrast to DHIS, the patient based system does not deal with the aggregated numbers but with the individual patient. The typical functionalities of a patient based system include:

1. Managing the administrative information of each patient
2. Recording the history of medical treatment of each patient. Therefore, sometimes it is referred to as Electronic Patient Record[20] (EPR). The history of the patient is a set of events related to the patient. An event can be the visit of the patient, the admission to the ward etc. In each event, depending on the system, there is more information captured in detail such as surgery, X-ray, lab test etc.

The data from the patient based system could be aggregated to feed the aggregate based system. If the two systems are implemented at the same time, aggregated reports can be produced automatically by retrieving aggregated value from the patient based system.

In this thesis, I dealt with two kinds of patient based system which are now briefly discussed.

IHAMS (Integrated HIV/AIDS Management System)

First, the ART module was developed to support ART treatment management in ART clinics in Ethiopia. ART module was a system which was hard coded with forms and reports in ART clinics. The system would be more effective if it was linked to other HIV/AIDS related programs such as VCT (Voluntary Counseling Testing), PMTCT (Preventing Mother to Child Transmission) etc. Therefore, a plan to extend the ART module to cover all aspects

[20] There is a different name referred to this system: Electronic Health Record.

of HIV/AIDS management was made. The goal was to have an Integrated HIV/AIDS Management System.

The flexible system (called OpenEPR - Open Electronic Patient Record)

The flexible system allows defining data entry forms and reports for patient based management without requiring programming knowledge. This system is developed in PHP/Mysql. The architecture of the system is similar to IHAMS. Key functionalities of OpenEPR:

- OpenEPR gives the power to the end-users who can define the system themselves without knowledge about programming.
- OpenEPR allows to make any forms with a variety number of control types (text, combo box, check box etc).
- OpenEPR supports standard reports and allows to integrate predefined reports.

This system was developed in Vietnam by me.

Now I discuss the two countries where the fieldwork was carried out.

5.2 Ethiopia

The content of this section is mostly taken from several Masters Thesis related to HISP Ethiopia such as Damitew and Gebreyesus (2005), Lagebo and Mekonnen (2005) etc, and also based on my relatively brief stay there.

5.2.1 Country Profile[21]

Ethiopia is situated in the Horn of Africa between 3 and 15 degrees north latitude and 33and 48 degrees east longitude (MoH, 2002; The World Factbook, 2004). The total area of the country is approximately 1,127,127 square kilometers. Ethiopia shares borders with Djibouti, Eritrea, Sudan, Kenya and Somalia. Unique among African countries, the ancient Ethiopian monarchy maintained its freedom from colonial rule, with the exception of the Italian

[21] Taken from Damitew and Gebreyesus (2005)

occupation from 1936-1941. Ethiopia is the oldest independent country in Africa, and one of the oldest in the world (ibid.). The three main colors of her flag (Green, Yellow and Red) were often adopted by other African countries upon independence (The World Factbook, 2004). Currently Ethiopia has a Federal Democratic Republic government with nine ethnically-based Regions and two city administrations. Each Region has its own parliament and is responsible for legislative and administrative functions except for foreign affairs and defense.

Demography

The total population of Ethiopia was about 67,851,281 by July 2004 as estimated from the 1994 census (The World Factbook, 2004). Of the total 49.8% and 50.2% are females and males respectively. The average population density is 52.2 per square km, with great variation among different regions. Population densities are highest in the highland Regions and lowest in the eastern and southern lowlands. Most of the districts along the borders of the country have densities of less than 10 persons per square km. Higher densities are found in the highland areas, mostly above the 1,500m contour line. 23.2% of the population is concentrated in 9% of the total land area. The settlement pattern of the population and its density greatly affect the provision of health care including the accessibility and utilization of existing health care facilities. In Ethiopia the age structure of the population is typical of many developing countries, with *0-14 years:* 44.7% (male 15,189,921; female 15,109,870) *15-64 years:* 52.5% (male 17,857,758; female 17,767,411) *65 years and over:* 2.8% (male 855,103; female 1,071,218) and the population growth rate is 1.89% (The World Factbook, 2004). The main religions are Orthodox Christian and Muslim. Many languages are spoken in the country including Amharic, Tigrigna, Oromigna, Guragigna, and Somaligna. English is the major foreign language taught in schools (Ibid.).

Economy

Almost 80% of the national economy is based on agriculture. The agricultural sector suffers from frequent droughts and poor cultivation practices. These problems of drought and under nourishment place further pressure on the health services of the country. Under Ethiopia's land tenure system, the government owns all land and provides long-term leases to the tenants; the system continues to hamper growth in the industrial sector as entrepreneurs are unable to use land as collateral for loans. Coffee is critical to the Ethiopian economy with exports of some $156 million in 2002 (MoH, 2002; The World Factbook, 2004). Widespread poverty along with the generally low income levels of the population, low education levels, inadequate access to water and sanitation facilities, a high rate of migration, and poor access to health services contribute to the high burden of ill health in the country. This situation is further aggravated by high population growth (MoH, 2002).

5.2.2 ICT development[22]

Computers were introduced into Ethiopia in the early 1960's by some organizations like Ethiopian Airlines, Economic Commission for Africa (ECA), Ethio-Djibouti Railway Company, Imperial Board of Telecommunications, and the Haile Selasie I University. The Ethiopian Science and Technology Commission and Ethiopian Telecommunications Corporation contributed significantly to introduce ICTs and create public awareness in the country.

Initially, the National Computer Committee, and then the National Computer Centre, that were hosted in the Ethiopian Science and Technology Commission, played a leading role in enabling the imports of IT products by providing evaluation reports in relation to the context of the country. In addition to that, the National Computer and Information Centre (NCIC) played a leading role to develop the national ICT policy which describes the development of a national ICT infrastructure as one of the national strategic components to make changes and improve the determinants of national socio-economic performance. As of July 2003, the Ethiopian Information Communication Technology Development Authority (EICTDA) was established by Proclamation No. 360/2003 with the objective of utilizing ICT for socio-economic development and the building of democracy and good governance in the country (Federal Negarit Gazeta, 2003). The authority designed various projects to develop the ICT infrastructure at the three tiers of the government system (Federal, Region and Wereda). Out of these, the Wereda-Net and the Content and Application Development Projects are the major ones. The Wereda net project is aimed to develop ICT infrastructure in 571 weredas of the country (EICTDA, December 2004). The Content and Application Project is also working in identifying the prioritised sector organizations' (Health, Agriculture

[22] Taken from Lagebo and Mekonnen (2005)

and Education) content to develop shareable applications and to utilize the established infrastructure. The Ministry of Health (MoH) is member to a steering committee in EICTDA to utilize the established Wereda-Net infrastructure for the health sector. Currently, the MoH has identified four content areas; HIS, human resources, integrated financial management, and property management to develop applications that will be a shared resource amongst the health institutions at various levels. Currently, computers are being used in the health sector mostly for word processing activities. To make the best use of ICT, there are some efforts and initiatives that are ongoing to implement telemedicine in collaboration with different governmental and international agencies since 2002.

5.2.3 Ethiopia health information systems[23]

HMIS in Ethiopia has a poor status. At the national level, the HMIS has been established for receiving summarized data from all the Regional Health Bureaus each quarter of the year. At the end of each year, the report is issued in a Ministry of Health (MoH) publication which includes health and health related indicators (MoH, 2002; Awala, 2003). The major concerns regarding the current HMIS in Ethiopia refer to certain gaps in coverage, particularly drug-related and financial management indicators. Lack of timeliness and completeness of reporting remains a weakness of the HMIS, and such delays contribute to the failure (at all levels) to use data as the basis for informed decision-making in planning and management. In addition, parallel reporting mechanisms persist, with program-based and donor-supported initiatives resulting in multiple reporting formats and an increased administrative workload (Tigray Health Profile, 2003; MoH, 2002; WHO, 1999).

5.2.4 The current situation of HISP in Ethiopia

HISP in Ethiopia was initiated in 2003 with the base in Addis Ababa University, the national university of Ethiopia. HISP Ethiopia has a large team including 5 PhD students, 2 graduate Masters students, and 5 currently enrolled Masters students. Like India, in every region, there are system facilitators who are supporting the implementation process.

[23] Taken from Damitew and Gebreyesus (2005)

HISP Ethiopia also has experienced a process of evolution, starting in the Addis Ababa region, and currently is ongoing in 5 regions. In Addis, DHIS 1.3 version was fully scaled up in all the health facilities. DHIS version 1.3 is Access based, and transition plans are being made to DHIS2. In the four other regions (Oromia, Tigray, Bengshangul and Amhara), similar transition plans are being planned.

As a country affected seriously by HIV/AIDS, the need for a HIV/AIDS management system is very urgent. A group of 2 Masters students from Oslo, supported by myself, and a local medical doctor have established a patient based system for ART management and related programs. This system is currently under acceptance and testing.

The Vietnam research setting will be presented in the next section.

5.3 Vietnam

Here I briefly provide a description of the research setting of Vietnam.

5.3.1 Country Profile

Vietnam is located in Southeast part of Asia. It is a small country in terms of area (approximately 330.000 km2) but a big country in terms of population (80 million). Vietnam was known widely in the world through its war with America (1960-1975). When I traveled to other countries, everybody asked me about the war, and made me realize how the war has shaped people's perceptions about Vietnam. But Vietnam has been changing significantly since 1986 both with respect to economy and social changes. Currently, Vietnam has one of the fastest growth rates in the world (7-9% every year since 1994). The ICT industry is also rapidly developing at the rate of 35% per year, including an increasing penetration of computers and Internet in every field of social-and economic life.

The Table 13 below presents the key facts of country profile (http://news.bbc.co.uk/2/hi/asia-pacific/country_profiles/1243338.stm):

Full name:	Socialist Republic of Vietnam

Population:	83.6 million (UN, 2005)
Capital:	Hanoi
Largest city:	Ho Chi Minh City
Area:	329,247 sq km (127,123 sq miles)
Major language:	Vietnamese
Major religion:	Buddhism
Life expectancy:	68 years (men), 72 years (women) (UN)
Monetary unit:	1 dong = 100 xu
Main exports:	Petroleum, rice, coffee, clothing, fish
GNI per capita:	US $620 (World Bank, 2006)
Internet domain:	.vn
International dialing code:	+84

Table 13: The key facts of the country profile

5.3.2 ICT development

The Table 14 below summarizes key indicators about ICT in Vietnam (website of Asian Pacific Development Information Program, http://www.apdip.net/projects/dig-rev/info/vn/).

Total population	80 million
Rural population as a percentage of total population	75%
Key economic sectors	Oil and gas, agricultural products, fishery
Literacy in the national language(s)	95%
Computer ownership per 100 inhabitants	1.17
Telephone lines per 100 inhabitants	2.44 (Working) and 3 (Installed)
Internet hosts per 10,000 inhabitants	0.06
Internet cafés/telecentres per 10,000 inhabitants	6.2 (5,000 Internet cafés)

Internet users per 100 inhabitants	1.3
Cell phone subscribers per 100 inhabitants	1.9
National bandwidth within the country	156 Mbps (north-south)
National bandwidth to and from the country	106 Mbps
Ratio of incoming to outgoing Internet traffic volume	25:75

Table 14: Summary of ICT in Vietnam

At the national level, ICT development is strongly backed by the state government and many policies have been formulated. For example, on the 2nd March 2004 the Prime minister signed Decision No. 235/QD-TTg to foster OSS development and use. Vietnamese people are strong in Mathematics, eager to learn new things, especially IT. Also, since Vietnam has quite a large number of universities, colleges, and institutes which provide degree programs in IT, the human resources in IT is vibrant. And since, Vietnam also has a strong IT industry, there is a good link between the university and industry sectors, contributing to make Vietnam one of the top 15 leading outsourcing countries in the world.

5.3.3 Vietnam health information systems

5.3.3.1 The reporting system

Paper based reporting systems in Vietnam are very heterogeneous and complicated. Beside the national reporting system, there are other reporting systems used at the provincial and district level. There are also reporting systems managed by national health program. Besides an aggregate reporting system, there is also a need for patient based reporting system. Different departments in MoH are responsible for various reporting systems. In brief, the reporting system in Vietnam can be grouped into four main categories:

1. The 15-form reporting system (aggregate based, Department of Planning and

Statistic)

2. The treatment reporting system (patient based, Department of Treatment)
3. Health program reporting systems (National health program such as Tuberculosis, HIV/AIDS, Malaria etc)
4. Provincial level and other internal reporting systems.

I, in the following section, take a closer look at these reporting systems

5.3.3.2 The 15-form reporting system (Aggregate based, Department of Planning and Statistic)

This reporting system is under control of the Department of Planning and Statistic (DOPS), Ministry of Health. In 2003, DOPS reformed the reporting system under a project sponsored by UNPA and UNICEF. The approach used was based on the "minimal data set" as suggested by the WHO. The national reporting system which is used for all provinces and districts contains 15 forms for different health management sector and health programs such as health staff, budget, child health, mother health program etc. Each form has approximately 20-30 data elements. The last part of the report uses ICD (International Classify Disease) codes. In total, there are about 400-500 data elements which need to be collected within the overall reporting system.

The frequency of reporting in the 15-form system is not purely quarter based but 3, 6, 9, and 12 months. For example, 6 month report will be the summary number from January to June. Semi permanent data for 6 months will be the number in June. The following table shows how to calculate the aggregated numbers:

Report	Routine data	Semi permanent
3 months	Sum (Jan-March)	Number at the end of March
6 months	Sum (Jan-June)	Number at the end of June
9 months	Sum (Jan-September)	Number at the end of September
12 months	Sum (Jan-December)	Number at the end of December

Table 15 Summary of data aggregation rules

The number of data element was significantly reduced after the 15 form system was introduced. However, there were some inconsistencies in this reporting system which makes it difficult to move to a computerized HIS. This reporting system does not separate the semi-permanent and routine data, and one report may contain both types of data. The health workers have to differentiate the two types of data and use the correct method to collect individual data, and unnecessarily need to collect population data and infrastructure data every 3 months

DOPS also published a book called "National Health Indicators" which describes more than 100 indicators with detailed formula and calculation method.

To computerize this reporting system, the IT office also developed a software called HMIS supported by several international organizations. This software has been piloted in some provinces. The HMIS has some limitations though:

- Does not allow defining organization hierarchy and data element. All are fixed in advance.
- No indicator calculation engine
- No validation rules existing in the software

However, HMIS is still in development stage and the functionality is expected to be expanded.

5.3.3.3 The treatment reporting system (patient based, Department of Treatment)

Unlike the 15-form reporting system, the treatment reporting system requires reports to be based on individual patients. This system is owned by Department of Treatment. All hospitals have to send the report containing basis information about the treatment process of patient to the MoH. This reporting system has been computerized by a local company with the software called Medisoft[24], and has been decided as the official software to be used in all hospitals in Vietnam. This decision was signed by the Minister. Medisoft is a desktop based software using .NET technology, and is not open source and web based and relatively hard coded.

[24] http://medisoft.vn

Here are several functionalities of Medisoft:

- Allows managing inpatient, outpatient data
- Allows keeping history of treatment of patients
- Allows recording surgery during the treatment
- Supports different kind of aggregated reports
- Exchanges data using HL7 standard

Figure 10 presents the reception form of Medisoft:

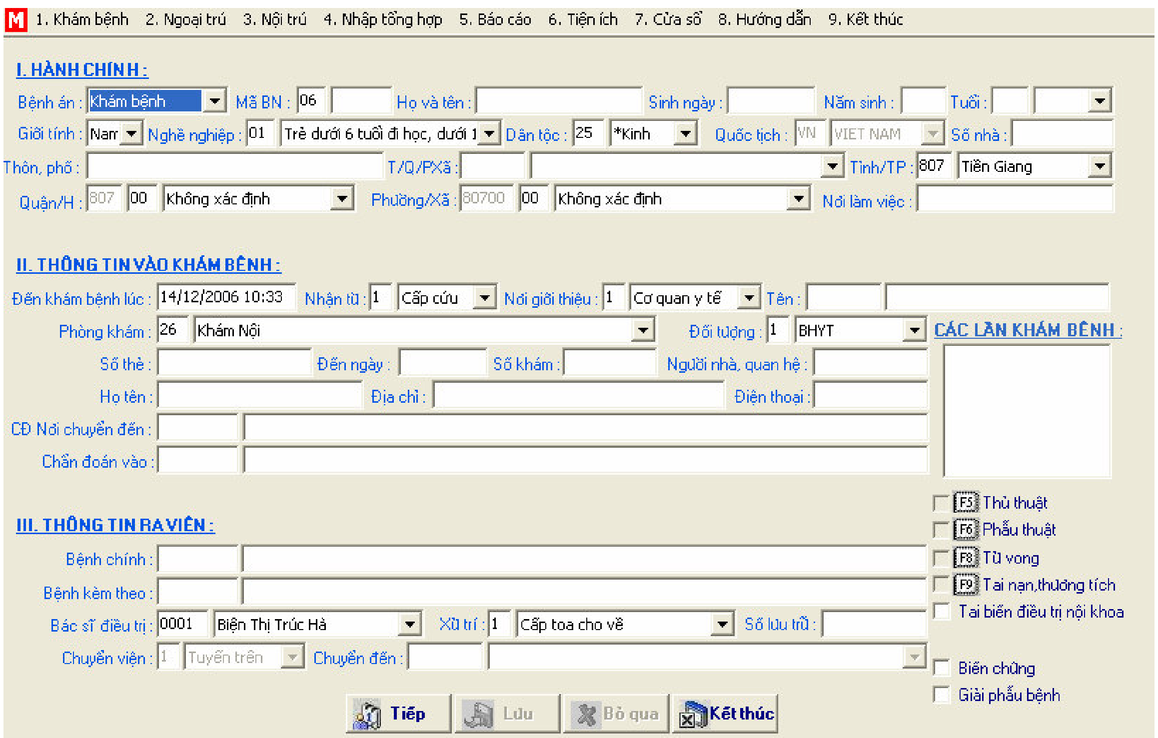


Figure 10: The reception form of Medisoft software

The Medisoft team has also developed the functionality allowing capturing the 15-form reporting system.

5.3.3.4 The vertical health program reporting system

This reporting system is run vertically from the national level down to the ward level. There are approximately 27 vertical health programs. Some of the well-know programs are mother and child, child nutrition, tuberculosis, and HIV/AIDS. Vertical health programs collect data for their own use and do not share it with the other health programs.

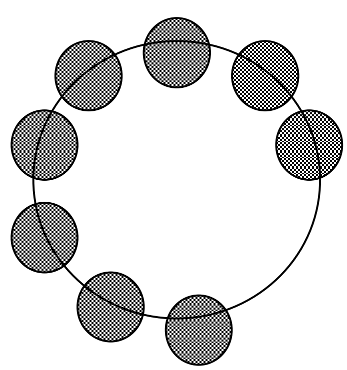
Here is the list of the existing vertical health programs:

- Mother and Child health care program
- Immunization program
- Tuberculosis program
- Malaria program
- HIV/AIDS program
- Mal-nutrition program

...

However, there is a relationship between the vertical health program and the 15-form reporting system. Actually, one health program can be one form in the 15-form system. Typically, the number of data elements in one health program is greater than those in the form in the 15-form system. This is the reason why the health program manager said that they could not use the 15-form because it is not detailed enough. It lacks some of the data elements they need.

The relationship between the vertical health programs and 15-form can be visualized in the Figure 11:



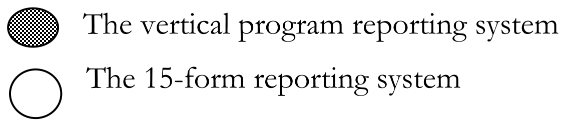


Figure 11: Relationship between vertical program reporting system and 15-form reporting system

5.3.4 The current situation of HISP in Vietnam

Currently, the project has 3 full time developers, 1 facilitator, 1 advisor and 1 Masters student who is completing the study (myself), and 1 coordinator who is responsible for the whole project in Vietnam. The HISP Vietnam team has been involved in the development of the DHIS 2 and the Patient based system and the implementation in Ho Chi Minh City and also other provinces.

The case study tells the story about development and implementation of health information system software within the HISP project, involving the following nodes in North-South-South network: mainly Oslo, Vietnam and Ethiopia and to some extent India. The case study is built upon from experience the period of February 2005 to December 2006. The case study includes two cases: development and implementation OSS for health care: aggregate based software (DHIS2) and patient based software (ART and the flexible system). The following table summaries the two cases in two countries:

Type of software/Place	*Ethiopia*	*Vietnam*
DHIS2	***x***	***x***
Patient based software	***x***	***x***

Table 16: The existence of two systems in two countries

The case of DHIS2 and the Patient based system are discussed in the two chapters 6 and 7 respectively.

CHAPTER 6: THE CASE OF DHIS2 SOFTWARE

DHIS2 development has a long and fascinating history. It is a multilayered story taken place over a vast geography during a long time. This chapter is organized as follows. The first section provides general information about DHIS2 development and implementation, including details of the project initiative, the management and execution of the project. Sections 6.2 and 6.3 discuss in depth the cases of DHIS2 in Ethiopia and Vietnam respectively.

6.1 General DHIS2 background

6.1.1 Project start-up

The DHIS2 story started in the first half of 2004. At that time, DHIS version 1.3 had become a national system in South Africa. And since 1999, the efforts to spread the success in South Africa to other countries had commenced. The countries where the project was being carried out were India (2000), Mozambique (1999), Ethiopia (2003), Vietnam (2004) etc.

DHIS1.3 was criticized by health authorities in several places such as Mozambique and India due to its outdated technology (Visual Basic and Access). For example, the Ministry of Health in Mozambique after conducting an evaluation concluded that: "*DHIS technology is outdated by a decade, it should be migrated to SQL server, tiers should be split, and it should be web-enabled*" (Braa *et al.*, 2006). The weakness in the technology and design of DHIS.13 was also realized by one of the HISP members:

"*When I joined the project, at first, I was very surprised about the design of DHIS and technology used to build it*"

In August 2005, HISP got funding from EU (European Union) to build a new software for capturing, processing routine and semi-permanent data. The software would help to improve the decision making process in the primary health care sector (BEANISH project). According to the report sent to EU, the tool was needed urgently because of the following reasons:

- *The need to rapidly design data collection tools for a variety of purposes.*
- *The need to efficiently capture or import/collate this variety of data in an "integrated" manner, then to monitor the processing and flow of this data, and finally to communicate with various stakeholders in the process.*
- *The need to analyze relatively large and complicated data sets quickly and efficiently. Another way of saying the same is that the term "hybrid" denotes that the DHIS actually needs to combine the typical properties of a flexible TRANSACTION database with the typical properties of a DATA WAREHOUSE and some properties of a COMMUNICATION TOOL.*

The management team of HISP project, which comprised of Professors and researchers at the Department of Informatics (University of Oslo), had taken this chance to build a new version of DHIS which was based on a better and powerful platform (Java and its frameworks), and open source.

6.1.2 The choice of technology

Java or PHP or Ruby or ASP.NET?

The first question the DHIS2 coordinators faced when they started the project was which programming language will be used to develop the software. There were many discussions to find the answer for that question, mostly done through electronic means (i.e. email, website) or face-to-face meetings.

First, one coordinator suggested to use PHP (an interpretive server scripting language). His argument was that PHP is easy to use and also very powerful. However, the issue was that no one neither in India nor Oslo knew PHP. Also, PHP did not support any MVC (Model View Controller) frameworks at that time.

Another technology which was mentioned in the discussion was Ruby in Rails. At that point of time, it too was a new technology. Microsoft's technology, ASP.NET, was also considered because the team in India had experience in developing some modules using ASP.NET. However, ASP.NET is not an open source technology, and the idea was thus dropped.

Eventually, Java was selected to build DHIS2. Below is the email in the mailing list on the final decision for using Java:

"*During the first half of 2004, discussions and workshops were carried out between South Africa and Oslo. South Africa had already selected Java based tools for the development of the new data dictionary, and since this was also a preference in Oslo, lightweight Java containers /frameworks were eventually selected as the development platform*".
(DHIS2 mailing list, 2004)

And what Java frameworks?

After Java was selected, another question raised was which Java frameworks should be used? Java was famous for EJB (Enterprise Java Bean) technology. But at that time, there was another approach called lightweight container framework (Spring) advocated by Rod Johnson (2004) (also the author of Spring) in his book: "*J2EE Development without EJB*[25]". The main argument he made in his book is that EJB is too heavy and complicated, and sometimes unnecessary. This book influenced the decision of the team to choose between heavyweight or lightweight container:

"*I have been reading in the recently published book {J2EE Development without EJB}. It promotes the strategy we are planning to use, namely using a lightweight framework for our business layer. This will reduce complexity compared to using EJB. This strategy work best for web application, and I would thing for our*

[25] http://www.amazon.com/exec/obidos/tg/detail/-/0764558315/ref=pd_bxgy_img_2/103-3711788-4875069?v=glance&s=books

application as well. But I think we must keep in mind that we are not firstly developing a web application, but that web enabling comes as an added value".
(Mailing list archive, 2004)

Spring therefore was selected as a bean container instead of EJB.

The database management system for DHIS2 suggested was either Mysql or another open source DBMS:

"*One thing addressed is the choice of database engine, with Mysql seeming like a good choice at this time. However, there are serious alternative contenders*".
(*DHIS2 mailing list*, 2004)

But at later stage, when DHIS2 was designed by layers, Hibernate was selected to make the persistent layer become more generic. It would enable DHIS2 to be run on all kinds of database servers such as Mysql, PostgresSQL, etc.

The selection of GUI (Graphic User Interface) was also discussed in the early days of DHIS2:

"For HTML interfaces, there are now some very interesting open source Java web frameworks, the primary ones being Tapestry (and the newer, simpler Wicket) or Webwork (used for Confluence). Lately, an alternative framework, Ruby on Rails has received a lot of attention".
(*DHIS2 mailing list*, 2004)

One of the criteria to port DHIS to a new framework is that the new DHIS must use a *layer design,* meaning that the code and the user interface are separated. At that time, Struts[26] was very popular and used widely. However, Webwork was selected instead of Struts. The reason for this was varied. One of the key developers of DHIS2 explained that the development of

[26] http://struts.apache.org/

Struts had been stopped so selecting it could be a risk. That decision can be seen to be right because now we are hearing that Struts merged with Webwork to become Struts2.

At this time, HISP attracted two good Java developers who were also Masters students at the University of Oslo. Later, these developers played the role of technical architects. The choice of other tools and frameworks was suggested by these developers. For example, Maven was selected as a build tool instead of Ant which was more famous and mature at that time. The fact was that one of these developers was also a developer of the Maven project. This developer convinced the whole team that Maven is better than Ant. The selection of Subversion instead of CVS was also based on his suggestion. To assure the quality of code, JUnit was selected as a unit testing framework. According to that, every single method must have a test to check its correctness and to be sure it functions as expected. A website for the development of the project was also set up at http://hisp.info. To increase discussion and community contribution, the project website used the "wiki" approach so that all registered users could join in editing the content of the website. A famous wiki tool called Confluence was selected. This is not an OSS but HISP could get a free license as it is a non-profit organization. The reason for selecting Confluence was that it was built by using frameworks which are similar to DHIS2 frameworks. For bug tracking, JIRA was selected, which was also a well-known Java based OSS tool.

Table 17 summarizes frameworks used in DHIS2:

No	Name of frameworks selected	Instead of	Underlying reasons	Argued for by
0	Java	PHP, ASP.NET, Ruby on Rails	Java is mature to build such big enterprise applications	The coordinators of DHIS2 development, who were also doing PhD in IFI, UiO.
1	Spring	EJB	A lightweight container framework	

2	Hibernate	JDBC, ODBC	A Object-Relational mapping tool	Two Masters students who joined the HISP project
3	Webwork	Struts	MVC (Model View Controller) framework	
	Maven	Ant	A tool to manage libraries and build project	
4	JUnit		Unit Testing	
5	Subversion	CVS	Concurrent version control tool	
6	Confluence		For project information	
7	Jira		For bug tracking	

Table 17: Summary of frameworks used in DHIS2

In this section, I have discussed the tool and framework selection process in the early phase of DHIS2 development. In the next section, I will discuss how DHIS2 development was managed.

6.1.3 Project management:

It was decided by the coordinators of HISP that DHIS2 would be open source, web-based and importantly based on a distributed development model. The HISP coordinators planned to distribute the development of DHIS2 to solve the limitation of the DHIS1.3 development model. For example, DHIS1.3 was developed in South Africa so that when it was spread to other countries, there was lack of capacity and understanding to customize the software. Furthermore, getting support from only one node could create a risk. The development should be distributed to build the local capacity to sustain and maintain the system at a later stage. Also, it reduced the dependency on one country. The distribution of the development work was attempted by using mailing list, project website, and the issue tracker, described as follows by DHIS2 coordinators:

"*Like is common in distributed open source efforts, various net based communication tools have been used to try to develop this global community including mailing lists, a collaborative web site (wiki), and the use of a bug/issue tracker*".
(Braa *et al.*, 2006)

There were also several discussions around whether HISP should setup its own OSS development services (i.e. mailing list, wiki, and issue tracker) or those provided by Sourceforge instead. At the beginning, some discussions were done through a mailing list provided by Sourceforge. Later some members argued that HISP should have its own services because Sourceforge provides services using very different technologies compared to the services HISP wanted to use. For example, Sourceforge wiki is PHP-based while HISP wanted to use a Java-based wiki; Sourceforge uses CVS while HISP wanted to use Subversion.

Eventually, the official website of HISP project was setup at http://hisp.info. A mailing list for developer discussions was also provided at dev@dhis.hisp.info. However, maintaining such services required a lot of time and effort. There was also a plan to migrate the mailing list to a public service like Google with the argument: "*It removes the hassle of administration*", a suggestion was made by one of HISP members who was responsible to maintain the website:

"This is my main concern - because of the lack of resources in the project, and for the long time viability of DHIS 2. We've experienced problems with the mailing lists and with Confluence, though both seem to be running OK at the moment".
(*DHIS2 mailing list*, 2006)

Although the mailing list, project wiki and central repository were vital for distributing the development work, they were only used widely in one or two counties: "*The use of these tools has however remained highly variable across different development nodes*" (Braa *et al.* 2006).

6.2 Project execution:

The development of DHIS2 finally commenced in the beginning of 2005. There were one group of students in Oslo who were taking the OSS course and another group in Vietnam who did their internships in one local software outsourcing company (ComA). One Masters student from Oslo was "commissioned" to Vietnam. His responsibility was to build up a development team in Vietnam based on the student group there (for more, please read 1.1.5).

The students in Oslo during the course were assigned into different project groups. Each group was responsible for the development of one module. There were approximately 10 different modules, covering almost all the functionalities of DHIS2, for example, data dictionary, organization hierarchy, data entry, import-export etc. However, at the end of the semester, most of the modules were not completed as planned because of several reasons. First, the "developers" of these modules were students with different levels of competence and experience. In a short period of time, they had to learn about the system and related technologies, and so the actual time given for development was very limited. Second, the architecture of the core module based on which other modules were built was not stable and changed over time. The change in other modules was not done in time to reflect the changes in the core module and so the modules developed became out-dated and could not be used. Third, due to time limits, the "student developers" did not have enough exposure to the actual use of the system, so the systems were not tailored to the users' needs.

In the fall semester of 2005, another OSS course was organized. At this time, HISP attracted two more relatively good Java developers to replace the earlier ones who had left the project. These two developers were students who took the OSS course in the previous semester. Like in the previous semester, the students in the class were also divided into groups. Each group either continued the work of the students from the previous semester or started building new module from scratch. However, similar to the previous semester, none of these modules were completely finished, leading to delays in releasing the first version of DHIS2.

Photo 1: The Oslo team and HISP India team in Kerala, India early 2006

This delay in releasing of DHIS2 made the Indian team become impatient. They needed an OSS version of DHIS for local implementation. They had waited for DHIS2 for almost a year but it had not come. They decided to build their own system which was web-based and Java-based with the support of one Indian developer who had worked in Norway. The Indian team with two Java developers worked intensively to release the "Indian version" of DHIS2 in two months, covering basic functionalities such as organization management, dataset management and data entry. However, this action was criticized by the Oslo team with the argument that the Indian team should join in the global development of DHIS2 instead of making a new system. A meeting was organized at the end of 2005 to find out how to align the efforts in India and Oslo. In that meeting, the decision to build a core version of DHIS2 was made, which would cover only basic functionalities of DHIS such as data entry, organization management, data set management etc. The deadline set for this core version to be released was February 2006. One developer from Oslo flew to India and one to Vietnam to work with the teams in both the places. Eventually, the first version of DHIS2 was completed with the core features: organization management, data element management and data entry. The success of the core version of DHIS2 helped to inspire the whole development team. A road map for DHIS2 development was set up for the first time, consisting of a list of milestones which were due on every 15th of each month. Every milestone in turn described a list of functionalities which would be included. The first three milestones were achieved relatively smoothly. However, since the release of milestone 4, the development process progressed slowly because of several reasons. First, the members of the core team became busy with their studies. Second, the relatively easy developed modules were completed, the remaining modules were more complicated. Third, the core team, after a long time of working intensively, needed some time to rest. However, the implementation in India, Ethiopia and Vietnam continuously requested more and more features from DHIS2. The consequence was that lots of functionalities such as indicator engine, data mart, and import-export etc requested by the team in India, Ethiopia and Vietnam were not implemented in time. The urgent needs for further development were reflected in one of emails sent by one India team member over time:

21.4.2006

First the India team member was keen to remind the development team about the needs for India:

"The milestone 4 shows that nothing has been included which we have asked for: The immediate requirement is sort order of data elements while defining dataset. This was mentioned in my previous mail also, but nothing has been done…" (*DHIS2 mailing list*, 2006)

3.6.2006

And when the milestone was released, he became disappointed because the features requested by his team did not come:

"Sorry for my late feedback. Yesterday I was going through the DHIS2 with latest milestone that we have downloaded and built. I didn't see any functionality as such from the previous version except the username and password" (*DHIS2 mailing list*, 2006)

23.6.2006

And in the following email, he totally lost his patience:

"I have been writing to dev list many things and some of the things are not even acknowledged whether you have read it or not, whether my suggestion and requirements are taken into consideration. In that case we just have to assume that the dev's are ignoring those suggestions/requirements" (*DHIS2 mailing list*, 2006)

The delay in releasing features needed for countries where the implementation was going on affected greatly to the implementation processes in the three countries.

In this section, I have discussed the execution of DHIS2 development by presenting related activities which had taken place in Norway and other countries. In the next section, I will discuss the DHIS2 development and implementation in Vietnam.

6.3 DHIS 2 in Vietnam

The development and implementation of DHIS2 in Vietnam can be divided into two periods: January 2005 – August 2005 and January 2006 – December 2006. This section is organized as follows. The first subsection discusses the DHIS2 development process in Vietnam during the period from January 2005 to August 2005. In the second subsection, I discuss the development and implementation of DHIS2 from January 2006 – December 2006.

6.3.1 January 2005-August 2005

The first action to distribute the DHIS2 development to developing countries was through the visit of a team from University of Oslo to Vietnam in January 2005. The team was composed of one professor, one PhD student, and one Masters student, who was also a Java developer. The visit served several purposes: to attend one OSS conference organized by Vietnam MOST (Ministry of Science and Technology), and set up the basis for collaboration of local partners. The timing was right for the involvement of HISP in Vietnam since the government was promoting OSS. In the conference, the professor gave a presentation about DHIS2 as an important OSS project for health care.

After the conference, the delegates visited Hue to set up another development node besides the one in Ho Chi Minh City. At that time, Hue had already been a pilot site of DHIS1.3 as per the agreement signed between Oslo and Thua Thien Hue Health Services signed in August 2004. The implementation of DHIS1.3 was supported by a state owned software company which based in Hue City (refered as ComB) during the period from August 2004 to December 2004.

The first meeting with a representative from ComB was not successful as they did not want to continue the collaboration with Oslo due to bad coordination and misunderstanding between the company and the Health Services. For example, ComB complained that some implementation work such as the class room used for training etc should be done by the Health Services rather than them.

The Director of ComA company in Ho Chi Minh City was informed the result of the meeting. In fact, he was the one who introduced HISP to ComB in Hue City and made all arrangements through his personal relationships. He made a call to the Director of the ComB and tried to arrange another meeting. It was a dinner in a riverside restaurant. In this meeting, the Oslo team tried to convince them by mentioning about the superior technologies being used in DHIS2, and argued that by joining the project, the staff from ComB could gain competence in advanced and current technologies which would be vital for strengthening their out sourcing business.

The ComB director was convinced by this argument and decided to continue to support the project. Moreover, ComB was willing to provide infrastructure and resources needed for the training. The PhD student and the Masters student (who was also a Java developer) stayed in Hue for two weeks to train the ComB team on DHIS2 technologies. The professor flew to Ho Chi Minh City to have a short meeting with the director of ComA. After that he went back to Norway. Two weeks later, the PhD student and the Masters student left Hue to go to Ho Chi Minh City. The Masters student stayed in HCMC for 6 months to help build a local team working with DHIS2.

Photo 2: Technical training given in ComA by Oslo Masters student

ComA only provided infrastructure for the development team but not staff. HISP had to use students who were interns in the company for the development work. There were four interns assigned to work for the project, divided into two groups, each responsible for one module. The two modules assigned to the Vietnamese teams were: Ward Patient Module (WPM) and Report Portal Module (RPM). The ward patient group was later supplemented by one developer from ComB. This new developer had rich experience in software development, and was also the project manager of the R&D (Research and Development) office in ComB.

The development process was started by the training given to the local team. The instructor was the Masters student from Oslo. Many training sessions initially were organized formally, in a class room. Gradually they were shifted to an on-job training. Although these four interns were students in a good local university they did not have any prior Java experience, which was problematic since DHIS2 was totally Java-based. This disadvantage impeded the progress of the project. The Oslo instructor gave a quick introduction to Spring, Hibernate, Maven etc, all which were very new to the interns. Exercises were assigned to the interns so that they could practice what they had learned. However, it was really difficult for the team to follow the training as they did not have any knowledge of Java.

After one month, the progress of the development was still focused on learning and doing exercises. No actual development work was done. The Oslo Masters student became impatient. He requested the team to make a plan for the module development: WPM for the first group and RPM for the second group. Since they were taking part in OSS project, they were requested to make the development plan and place it in Jira and project wiki. During this period of time, I was working for ComA and was responsible for the implementation of HISP in Ho Chi Minh City. I was also assigned to co-supervise the interns. When I saw the interns being confused in making the plan, I tried to help them. I guided them to use MS Project to make a plan which outlined the software development process, consisting of four different phases: Analysis, Design, Coding and Testing. The Oslo Masters student was not very pleased when he discovered that the team used a commercial tool like MS Project. He wanted to completely separate the OSS development from the conventional development model. However, as a compromise, the team used both approaches. They placed the project plan in both JIRA and DHIS2 wiki but, they still used the MS Project based plan to track their local work because they felt it was easier to visualize than JIRA. And during the development process, they adopted various practices which were in the conventional model. For example, they used ERD (Entity Relationship Diagram) to design the database schema; and used Dreamweaver® and MS Frontpage® to design web forms. So the development of DHIS2 in Vietnam in this period was a hybrid between the conventional and OSS models.

In April 2005, the Oslo Masters student made a trip to India to also train and build a team to work with DHIS2 there. In this trip, he gave training to a group of students from one local institute which had a collaboration with HISP India. However, these students left after finishing their studies and the knowledge was gone with them. Also what they had done was not usable because it was not specific to the implementation demands and there was no real database to test it.

The Norwegian Masters student came back to Vietnam after the 3 week-trip, and continued supporting the team in Ho Chi Minh City to fix bugs and solve other technical issues. At the same time, the request from the project group in Oslo to have a module which supported transaction management, Hibernate and Unit Testing became very urgent. As this module was vital for the whole architecture of DHIS2, the Oslo student took responsibility for it and spent more time completing his own work. He started to lose his patience with supporting the local team, arguing that he couldn't support such a big team of 4 students at the same time. He suggested restructuring the team so that only the two best students continued with the development work and that he would support only them. The other two should focus on field implementation i.e. doing training and user support of version 1.3.

This decision caused conflicts between HISP and the manager in ComA who was responsible for the interns. The manager tried to defend the interns by saying that the development work was very important to their thesis. In the context of education in Vietnam, graduation and certificates are very important to students. After this event, the Oslo student stopped supporting the two interns who were assigned to do the implementation work, and instead they started doing development on their own.

In June 2005, there was a meeting between ComA and the HISP coordinator to discuss hiring several interns to work with HISP after their study. After which, the company did not show much interest in working with HISP. They asked for a management fee at a rate which HISP could not afford. Therefore, the collaboration ended without achieving anything for HISP, as the interns were trained but did not work for HISP. Later, there interns worked for ComA and one worked for another company in Ho Chi Minh City.

In August 2005, I left Vietnam to join the Masters program in Oslo. The period from August 2005 – December 2005 was a really interesting time. There were three set of different activities taking place in various locations around DHIS2:

- In Oslo, the development of DHIS2 was still in the R&D process, no module had yet been completed.
- In Vietnam, students in Nong Lam University were supervised by the Dean of IT

department to do a report module of DHIS2 for Vietnam.

- In India, a group of two developers who developed a JSP-based version of DHIS2 were hired.

6.3.2 January 2006 - December 2006

The DHIS2 development in Vietnam was re-activated in early 2006 by recruiting 3 students from Nong Lam University to work for the project. The development of DHIS2 during this period can be divided into two smaller periods. The first one was from January – May 2006 and the other was May – December 2006. In the first period, the development took place in close collaboration with the Oslo team. The second period was a relatively independent one for the Vietnam team.

In the first period, with the support of another Masters student from Oslo, the team produced several modules which were needed for the local implementation. For example, the multi-lingual module for translating the user interface from English to Vietnamese; the report module for generating local reports, and the installation module for making the installation package. With support from the Norwegian Masters student, the team was able to complete these modules easily. Also, the development of these modules did not require too much of the overall understanding of the whole DHIS2.

In May 2006, the Norwegian Masters student left Vietnam leaving the team to work independently. As discussed earlier, after milestone 4, the team in Oslo reduced their focus on the development. It opened chances for the Vietnamese team to have more space to work independently but also brought with it other challenges. The team had to define tasks and work independently. In this period, the Oslo team did not involve much in the development process in Vietnam. They only gave technical support instead of driving the development. Therefore, the development in Vietnam at this stage was driven by the local implementation. The team developed and extended several modules to support the implementation. The development of some of these modules is now presented.

The user management module (UMM)

The team started to extend the UMM when DHIS2 was in the stage of milestone 5 (M5). M5 had UMM to control the access to the system. However, the module was not complete, lacking the functionality to identify the roles of each user. The consequence was that every user had equal privilege in accessing the whole system. DHIS2 is a web-based system, with the ideal case being that it is deployed on the Internet so that every facility with an Internet connection could use it online. This could reduce the burden involved in exporting and physically sending report files to upper levels. Several units such as big hospitals requested this security feature but we could not satisfy them because of the limitations of DHIS2 M5. Once deployed on the Internet, the confidentiality of data must be maintained, requiring full security functionality. Therefore, the need to improve the UMM was very urgent. When I suggested the team to do this improvement, they expressed their readiness to do so. This was very different from what happened before, when I had asked to do some improvements, the team had said: *"I will post a message to the mailing list. The developer [A, B, C for example] who had developed it will do it for us"*.

Or

"I am not very sure about this. The technology is so complex. I am afraid of it very much. Let other developers [Oslo team] do it".

The architecture of DHIS2 is modular; every module is more or less independent from other modules. One developer working on one module might not know anything about other modules. However, it would become difficult for one developer working on this module to shift to other modules because it would require learning the other modules.

We needed to use the organization tree in security-user module so that every user can only have access to a part of the tree according to their privilege. However, using the tree in other modules was a very complex task. The developer assigned to do the enhancement of the module said: *"Every time, somebody mentions the tree, I feel very scared"*.

But I persuaded him by saying that: "*It is not that difficult. Please take a look at it and you will realize that you can do it*". He got convinced and started to learn how the organization tree was used in DHIS2. We re-initiated the communication with the mailing list to get support for technical bugs which we faced during the development process. The developer from Oslo who made the organization tree module helped us solve them.

After several rounds of building and testing, lasting about three weeks, we completed the improvement of the security – user management module. The improvement made using DHIS2 over the Internet possible because the data was secured by privileges. Using DHIS2 over the Internet in turn could potentially help to scale up the implementation rapidly as the need for field support could decrease. After the success of making the improvement on the security-user management module, the team gained confidence at some level, and became ready for further development.

Photo 3: The HISP team in Ho Chi Minh City. The field visits of Norwegian Masters students helped to build capacity locally. But more important was the need to build the confidence of the local team.

The data mart module

The second example I use to demonstrate the transition of development from Oslo to Vietnam was the case of the data mart module. It is used to calculate all aggregated values and indicator values and storing it in several tables. After the storage is done, all requests for data value or indicator value could be made by accessing these tables. This approach reduced the time of re-calculation and allowed the analysis to run faster. By using the model of objects, calculation of processed values is done at the object level and after that it maps into the database level through Hibernate, making the calculation of data mart very slow. For example, in the M5 release, the calculation of the amount of data consisting 300 units, 100 data elements in 3 months took several hours. And of course, no user is satisfied with such a performance, and will refuse to use it. Without the data mart, the analysis part in DHIS2 which was claimed as its powerful feature could not be materialized.

At the same time, the HISP Indian team also faced the similar problem with this module. Because of the pressure from the implementation, they decided to develop their own way to calculate aggregated value for their report module. We were not surprised with this decision of the Indian team who had very competent developers that had already developed the first open source version of DHIS based on Java and JSP, even before the first milestone of DHIS2 was released in February 2006.

The problem of the data mart module was one of the most discussed topics in the DHIS2 developer mailing list. Some developers suggested using another "pragmatic way" to solve the problem of performance, such as using SQL queries to access database directly. This suggestion was also supported by one of the coordinator. But the "veteran" developer who was also the architect of DHIS2 responded very strongly:

"Please don't try short term solutions but try other ways around such as optimize parameters in Hibernate." (DHSI2 mailing list, 2006)

The team in Vietnam also decided to change this module in their own way, using an approach commonly used in accounting systems: trigger. This model allowed calculating immediately aggregated and indicator values every time one raw number is stored or modified. The time for data mart calculation in this method is reduced to 0 (compared to hours used in the previous method). It means that the data became fresh, live and ready for analysis after every data entry.

We tried to rework the application level (business logic). We made some changes in the data entry module so that every time one number is input, it evokes a function to update the data mart tables. We hoped that this small change could bring in larger efficiencies. But we failed several times as the calculations of data mart never came. Moreover, the change made the data entry module stop working. We guessed this was due to the use of Hibernate which we had not understood deeply enough.

Because of the failure with the change in application level, we tried another approach which interfered at the database level. Actually, the trigger is not a new concept in the database system. It is widely supported by both commercial and open source database management system (DBMS) such as Microsoft SQL Server, Oracle, PostgresSQL etc. We believed the trigger at the DBMS level can help to calculate the data mart value quickly and effectively. After 3 days, we successfully implemented this feature. But this trigger could not be shared over to the HISP network because of two limitations:

- First, it was developed in psql language only supported in PostgresSQL while the design of DHIS2 is database independent. So the trigger would not work with other DBMS such as Mysql, DB2 etc
- Second, it has difficulty in calculating indicator values which are based on complex formulas. Psql language does not support parsing expression while this calculation can be done in Java very easily.

But for the implementation in Vietnam where the role of indicators is still not emphasized the current approach using data mart at the database level functions adequately for supporting local needs.

The example of deploying GIS – KIDS

This example shows a different view of the autonomy of the team in Vietnam. Before M5, the analysis functionalities in DHIS2 had not been developed. From the M5, DHIS2 started to have a function which supported viewing the history of data of every unit. CSV export was available but still in its early stage. Pivot table was also added but it was not popular as compared to DHIS1.3 and DHIS1.4 because of several reasons:

- It depended on MS Excel (the support of pivot in OpenOffice was not experimented);
- It is a desktop based component which is reversed to the whole web architecture of DHIS2;
- It was complicated to set up; and,
- It uses data from data mart module which was not completed.

Before discussing the deployment of KIDS as a GIS module for DHIS2, we briefly reviewed the history of the GIS module within the project, which was given a lot of attention by DHIS2 development coordinators during its initial phase. This might be explained by the fact that one coordinator of the project had a strong research interest in GIS, a technology which is seen as an important tool for spatial analysis in the health care sector. Several prototypes of GISs were made by students involved in HISP projects, but none of them satisfied even the basic requirements. There were several reasons explained for this failure, such as the selected frameworks, the commitment of student developers etc. For example, in the fall semester 2005, there were two groups of students working on GIS: one for desktop GIS, other for web GIS. The desktop GIS group used Udig to develop a Java based and open source module but it was not able to connect to DHIS2 database to show the health data properly. Even if it could do that, it still raised the issue of integration with DHIS2 which is web-based system. The web GIS group used a system called MapServer, however, the final version of this module was far from the needs of DHIS2. For example, it did not allow linking with DHIS2 dynamically. The failure of several efforts showed that the Java based GIS open source was still at its infant stage.

In the Spring semester 2006, a group of students in Ethiopia (including me and four other from Ethiopia) started with the system called CartoWeb, a web-based and PHP GIS system. We spent several weeks to learn and tried to customize it for Addis Ababa region. It worked fine with Addis maps but we soon realized that it is very difficult to continue because of the following reasons:

- It requires many steps of manual configurations
- It is not very stable, sometimes, some parts of the map are missing
- It requires lots of work to adapt for DHIS2

At this time, we heard that the team in India decided to use a local GIS system developed by a domestic company. This is not an open source and free system but it was provided nearly free to the HISP project because the policy of the company to develop momentum for their product. As a result, the India team gave up the experiment with CartoWeb even though they had spent 3 months in trying to customize it. CartoWeb is a PHP system, which makes it difficult to integrate with DHIS2 which is based on Java.

The development of the GIS module only had a breakthrough when we got acquainted to KIDS[27] (Key Indicator Data System). The presentation made by one HISP member in Cape Town[28] persuaded us to join hands in customizing KIDS for our use. KIDS is an open source, Java based system developed by FAO (Farm and Agriculture Organization) as an analysis tool for the agriculture sector in developing countries. Actually, there is similarity between agriculture and health sector indicators. For example, in agriculture, there is also the need to visualize indicators by geography and time. Indeed, KIDS provides functionalities which fit our requirements. Therefore, we decided to choose KIDS as the solution for our GIS module. The following figure demonstrates the use of KIDS to view statistics of BMI (Body Measurement Index) in the world (http://www.who.int/bmi/index.jsp):

[27] Kids.org

[28] Within the DHIS workshop and DHIS conference in South Africa

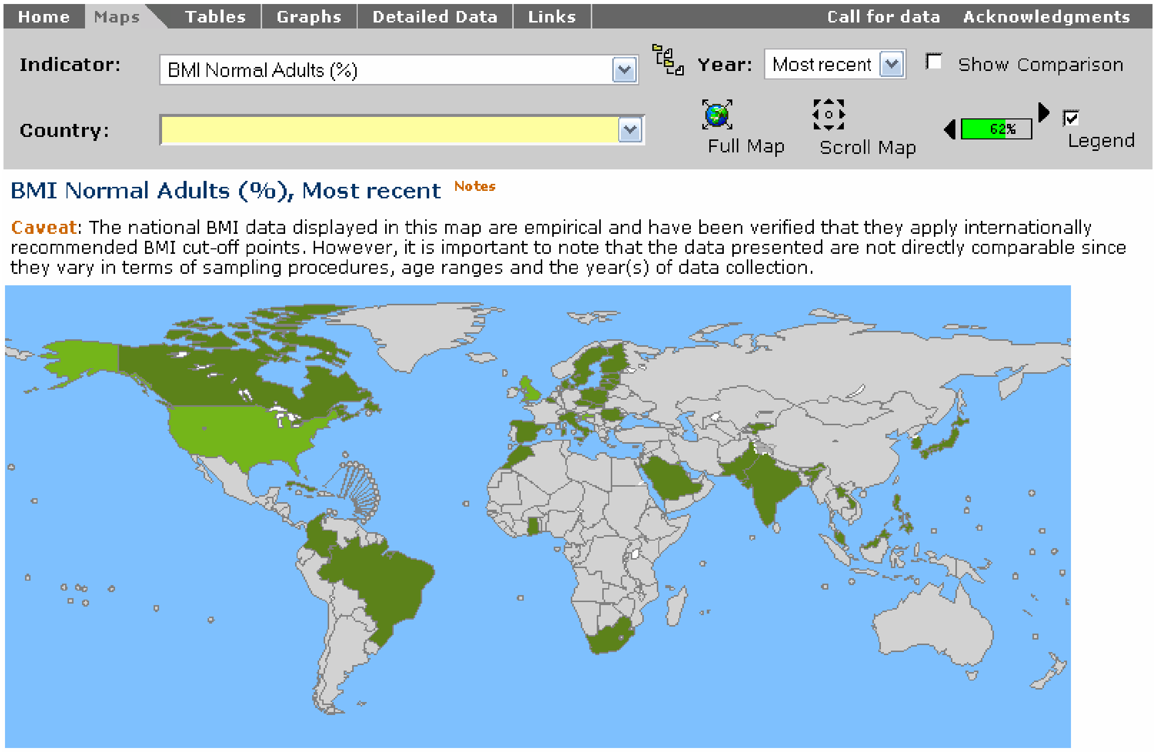


Figure 12: The main screen of KIDS for BMI distribution

Adapting Kids in Ho Chi Minh City

We decided to go with KIDS in for several reasons. First the data collection process had been taking place in Ho Chi Minh City for 3, 4 months so data collected was of a significant amount. We realized that it was the correct time to convert the raw data into useful information which would help enrolling the interests of the decision making process. Second, as we mentioned above, there were very few analysis support functionalities in DHIS2. KIDS could potentially help to fill up that gap. KIDS not only allowed users to view the distribution of data geographically but also provided some parts of pivot view, graphs and charts. With these built in features, KIDS provides an alternative to the analysis module of DHIS2.

Quick introduction of the team working on KIDS

My prior exposure to GIS came through the OSS course in Oslo, the GIS course in Addis Ababa and the GIS sessions in the DHIS conference in South Africa. First, I gave the team a quick introduction and demonstrated how KIDS works. Although GIS is a very popular concept, the Vietnamese team had only heard its name. Most of them did not understand what actually GIS was. After looking at the demo, one member commented: "*GIS is merely one way to analyze data by geography*".

While the comment was still very simple, it showed that the team was exposed to the concept of GIS. After that, I explained them how GIS could help to improve decision making in the health sector by making some basic spatial analysis. They were very convinced and eager to start with KIDS.

Preparing spatial and health data

The spatial data of Ho Chi Minh City was available with many different layers: street, river, health center, hospital, administration boundaries etc. However, the map we had did not contain district boundaries, while health managers at the city level needed to compare districts with each other. So then we had to find a way to make the district boundary layer based on ward boundaries. We sent a mail regarding this to the dhis2 mailing list and received support very quickly from one GIS expert in Oslo, who advised us to use a function called Union. He also suggested that he could help do this quickly because he was aware of the pressure on the implementation in Vietnam. We sent him the shape files of ward boundaries. He converted it into district boundaries and sent it back to us after one day.

After receiving the spatial data, we had another challenge, that of using non-spatial health data. KIDS accepts both kinds of data from the database and in CSV format, but when we tried, only the latter worked. There were two ways to prepare health data in CSV format: the data mart module and the CSV exporter module. Both having their limitations. Data mart module ran very slowly as discussed earlier. CSV exporter was too rigid, and only allowed exporting the unit with the full name[29]. This was a real problem in the case of Vietnamese language. KIDS did not understand Vietnamese Unit code. Every time we tried to import a CSV file into KIDS, it caused an error saying that there is no unit with that name in the

[29] In DHIS2, every unit has several names such as: long name, short name, alternative name etc

shape file. We tried another way to create CSV directly from data mart tables but we faced another problem because the current DBMS we used (PostgresSQL) did not have any utilities to do the CSV export.

We then had to encourage the team to make some small changes in the CSV exporter module so that it could support the use of short names, which would not contain Vietnamese characters. The developer who was assigned to do this task decided to post a request to the developer list.

His message:

"A problem is Vietnamese font. When I used kids application by choosing DISTRICT field or NAME field is KEY COLUMN that couldn't match with name of units in CSV file. Here, I also want to have short name in CSV for easiness.
About CSV export that also should to have criterion as exporting follow level and period. So, when using kids you can compare data following level at every period. I also to know structure of CSV file will be used for kids too?
Just some my though when trying with kids. Let's make me clear if can."
(DHSI2 mailing list, 2006)

The messages in reply from Oslo were:

"The standard CSV or XML format is primarily used to transfer data between two instances of DHIS 2. When it comes to other formats and other requirements, I'd suggest that you write exactly what you need and in what order, and we can implement a separate exporter for KIDS.
The good thing about the current export system is that we can quickly and easily add new export formats. So, what do you need?"
(DHSI2 mailing list, 2006)

The discussion took place over several rounds but did not reach a conclusion. The Vietnamese developers were not keen to follow the discussion because of other pressures from the implementation side. Therefore, the development of GIS module in Vietnam was halted.

The example of the report module

The Report module was developed in Vietnam using the Jasper report engine. It worked well with the reporting system in the Mother Health Program but when moved to Child Health Program it had a problem. This program report required that some semi-permanent data elements be collected by 3, 6, 9 and 12 months. While generating reports for 6 months, the system returned the number of months 1-2-3 plus the number of months 4-5-6. This gave a wrong value, for example:

- Number of malnutrition children under 2 years in the 1st quarter is 10
- In the 2nd quarter, there are 2 more children discovered with malnutrition so that the number of malnutrition children under 2 years in the 2nd quarter is 12
- A health worker gives 12 as the number of malnutrition children under 2 years in 6 months. DHIS2 was programmed to return the number for 6 months as 10+12 = 22. This was a wrong value. The correct value should have be 12.

The developer assigned to work on this module started communicating within the list. There were several solutions suggested, but none of them were concrete and actionable. For example, one solution was to add one more operator for data elements beside SUM and AVG. Another solution suggested was to use two periods in one report, which appeared unrealistic because it could increase the complexity of the module and require restructuring other modules.

I asked one team member to get involved in development to find a solution but she said:

"*I chatted with Mr A [a coordinator of HISP in Oslo] today. He understood our issues clearly. He promised to solve it for us soon. So let's wait for his reply"*.
(DHSI2 mailing list, 2006)

The reply she received was a message sent to the DHIS2 mailing list as follows:

"This is a big issue and very strange. I copy the message to our expert so that they can say something about it… Maybe we should ask the users to change the practice".
(DHSI2 mailing list, 2006)

After this message, everything became quiet. The team in Vietnam also postponed the implementation to wait until a solution came. As the date for the training was very near (in 2 weeks), I had to request her to complete it as soon as possible. Here was the dialog between us:

- *"Please tell me the status of customizing the reports?",* I asked
- *"I don't know how to do so I am still looking for a solution", she replied*
- *"What are your solutions? Can you tell me? "Why you have to look for a solution when the solution is existing there?" I* made it very strong.

Then I suggested her one approach to make some changes in the report generator module so that instead of returning the sum of quarter 1 and quarter 2, it returned only quarter 2. After that I asked her by when she could complete it, for which she said within this week. At the end, surprisingly, she could finish 1 day before the due date.

This "workaround" approach helped us to continue our implementation in the child health program without having any more problems. This solution in turn helped to address the "period" issue in the national reporting system because this reporting system was designed with a similar concept as that of the child health program.

6.4 DHIS2 in Ethiopia (February 2006 – December 2006)

The context of implementation of DHIS2 in Ethiopia was provided by the University of Oslo Masters program, as a part of which students spent one semester in Oslo, and the second semester at Addis Ababa University. This teaching took place in the Spring of 2006, and got together the Oslo faculty mentioned earlier who was deeply engaged in the DHIS2 initiative in India, 4 Ethiopian Masters students, the two Ethiopian Masters graduates who were now currently employed full time in HISP Ethiopia, and myself. The semester in Addis was spent both in attending two courses (Health Information Systems and Geographic Information Systems), and the students identifying their thesis research topics and initiating empirical work on their projects. Two pairs of student groups were created to work on these projects, one on the DHIS2 customization and the other on the Patient based system. Both these groups had nearly zero prior exposure on Java or PHP technologies, on which the DHIS 2 and the Patient based system were based respectively.

Photo 4: The author with HISP team in Ethiopia (March 2006). The "social glue" helped to make the communication become more effective.

We started with DHIS2 when it was at the stage of milestone 2 release. In this milestone, DHIS2 only provided basic functionalities for organization hierarchy, data element and data entry. It was still in an infant stage and immature. However, the Indian team had decided to implement DHIS2 in Kerala using this milestone, as the state had required an open source HIS.

We also started downloading the DHIS2 milestone 2, compiling it and customizing it. The DHIS2 report module made by the team in India was taken as the point of departure for the customization process in Ethiopia. We used M2 plus the "Indian report module" to start our customization process. The customization process included changing the report template and developing the new module called Mortality and Morbidity which is actually used to collect data according to ICD (International Classification Decease) format.

Exposure to technologies

First, we had to download the source code of DHIS2 from the central repository located in a server placed in Oslo, Norway. The Addis Ababa University network policy prevented us from checking out the source code using subversion. Subversion uses SSH protocol to work but the university policy did not allow its use. We then had to ask one of the team members in Oslo to download and send the source code to us via email. The first challenge was solved but another arose. We could not compile the source code because additional libraries were not included with the source code. The build tool used in DHIS2 (Maven) would download these libraries from Internet at compiling time. Again, the networking rules in Addis Ababa University did not allow downloading such libraries from the Internet. We also tried to change the configuration in Maven but we could not make it work. We then had to ask the team in Oslo to send us all libraries needed by email. And in this way, eventually, we could build the DHIS2 successfully.

In the Ethiopian context, we realized that DHIS is very Internet-dependent. Internet is something like a novelty in Ethiopia and maybe also in some other developing countries. India and Vietnam are different, as they have rather good Internet infrastructure but still very far from the Oslo standards.

After building DHIS2 and making it run on our computer, we started the customization process. We only tried to make additional modules which were needed for local use in Ethiopia because we had very limited human resources (two Masters students) and they both had very limited Java competence. We believed that quickly getting something to work could help develop the critical mass and this in turn would foster the development. The first module which we thought was needed for the implementation was the report module, which would retrieve data from the database and generate reports, replicating the paper reports. Usually, paper formats in the health care sector are different from country to country, and even among regions of one country. We started to develop the report module by designing reports for Addis Ababa region, consisting of 40 report templates.

To quickly get something done, we decided to develop the report module by using plain JSP (Java Server Page). Each report is one JSP file where the source code and report view are mixed in one file. This was different from the overall DHIS2 architecture which uses MVC (Model View Controller) approach. We decided to use JSP partly because we wanted to reuse the report module developed by the Indian team and partly because of the poor technical competence of the local team.

Learning by doing

The team had no experience and knowledge with Java and web design before. I started to guide them on designing reports in HTML format and inserting Java code into it. Getting them accustomed to DHIS2 was a difficult process because DHIS2 was like a cooking-pot of various technologies. Besides the knowledge needs of Java, the developers had to learn additional tools and frameworks such as Subversion, Maven, Jetty (a web server used in DHIS2) etc. During the learning cycle, I only provided guidance and tried not to interfere with their work. This approach was selected to help the team build technical capacity and also develop problem solving skills.

Photo 5: The HISP team in Ethiopia (March 2006). On the wall behind the team, there are lots of blue sheets used for explaining the technology.

During this time, HISP signed two agreements with two regions in Ethiopia to implement DHIS2 in May and June 2006. So the team had to complete the customization of DHIS2 for the two regions in a very short time. Due to this pressure for having something up and running quickly, I decided to get the team involved in the real development work and bypass the technological learning period. I also tried to hide the technological complexity which I thought was unnecessary to the development work. To help the team absorb the technical knowledge easily, I just tried to give them a small piece every day and by practicing it in real work they could gradually expand.

When I avoided telling the team about technology details, they complained and requested me to explain. Some times, they questioned why I asked them to do a particular thing instead of another, wanting to know clearly what they were doing. But I had to persuade them to "please do what we tell you to do and you will understand it later". And in reality, this approach was effective. Black boxing them from the technical complexity helped. For example, to make one report they had to know how to use a web page design tool to make it in the HTML format. After that, they must know how to use Java class to connect to the database to retrieve and show data in the web page. They also needed to understand the database model of DHIS2 to write SQL queries to get the expected data. It could take months for them to learn all the knowledge, which appeared unreasonable given the time constraints.

As discussed earlier, the DHIS2 project has an official website (located at http://hisp.info), using Confluence wiki, and a mailing list to facilitate the development. However we used it very little. The reasons were varying. It could explicitly be due to the slow Internet connectivity, and also that the team did not find it useful for their needs, as it took so much time to get the information they wanted. For example, the reply of the team in Oslo on the mailing list often was not specific at all, and instead of giving precise directions, they suggested a link to an online article or requested to review a previous email. That was why after trying several times, the team stopped using the wiki.

Follow-up

After some weeks, the report module was completed with 30 key report templates. Ethiopia still uses ICD (International Code Disease) version 5 from 1950. In DHIS 1.3 and DHIS 1.4, there is also the morbidity and mortality module. This is a very specific module and only used in Ethiopia. That was why we decided to build this module locally. The developer who worked on this module was a paid part time worker for HISP Ethiopia, who had also worked with this module in version 1.3 and 1.4.

One good thing for HISP in Ethiopia was that in some regions they had installed DHIS 1.3, 1.4 before. We thought that we could reuse a lot of what had been done in the previous version such as data set, organization unit, routine data etc. To speed up the customization process, I decided to build a small tool which was just one Java class to migrate all the data from the previous version to the new database structure in DHIS2. The tool helped to get the system ready for installing very quickly because we did not have to refine data sets and the organization unit hierarchy.

Sustaining the network

In the middle of April, I came back to Vietnam and left the team alone to do their work. This period was the real test to see the result of the local capacity building process. At this time, the new milestone of DHIS2 was released with more functionality and less bugs. We decided to upgrade to the new version, but in Vietnam where the connectivity is much better. I downloaded the code, built it completely and sent it back to the team in Ethiopia by email. It surprised me that the team could use the package we sent, to further develop it independently.

After two months working independently, I was informed by the team that they had completed the customization DHIS2 for Benishangul-Gumuz region but there were some remaining bugs especially with the MM module. At the end of May, we came back to Addis for 2 weeks to attend the Africa E-learning conference. During this second period of intensive work, we tried to solve problems but we and the team were very busy with the conference. Therefore, the problems were not solved properly. After I left Addis, the team could not fix the bugs in the MM module and integrate it to DHIS2.

At that time, the team requested help from Oslo. One developer and the coordinator came to Addis in August 2006 to train and work with the team. The Oslo team suggested giving up the MM module on JSP and developing a new one based on the MVC model (the same architecture as DHIS2). However, the MM module was more complex than any other modules, and the Ethiopia team was not capable of dealing with it. In September 2006, the Ethiopia team brought the MM module to the DHIS2 workshop in Cape Town to continue working on it. But the module was not completed.

The problem with the MM module was only solved when HISP Ethiopia employed a Java expert who was a lecturer at the Faculty of Engineering at Addis Ababa University to work part-time for the project. With a strong technical background in Java, he completed the MM module based on Webwork in a short time. This event opened up a chance for the implementation process to be re-activated. Early 2007, a HISP Ethiopia staff who had already gotten admitted to a PhD program at the University of Oslo, came to Oslo for her studies. She joined in the core development team and helped to translate DHIS2 to Amharic[30]. With the support from the core team, Amharic has been successfully added into DHIS2, this would help in convincing the Ministry of Health to approve the system.

[30] Main language used in Ethiopia

CHAPTER 7: THE CASE OF THE PATIENT BASED SYSTEM (ART, IHAMS AND THE FLEXIBLE SYSTEM)

This chapter discusses the case of the Patient based system including its general background (section 7.1) and detailed activities in Ethiopia and Vietnam (section 7.2 and 7.3).

7.1 General background of patient based system

7.1.1 Project start up:

The context for the development of patient based system in Ethiopia shared similarities and differences with DHIS2 described above.

In Ethiopia, like in many other developing countries, there is an urgent need for patient-based health management system, in addition to the aggregate based system like DHIS. A starting point for building such a system in Ethiopia, was an Open source course conducted in Oslo in fall 2006, where one of the student groups conducted a survey on the available open source application for patient based systems. A recommendation from this group was to explore the potential of Care2x to meet these patient specific reporting needs as an "Integrated hospital management system".

The basis for the Oslo group's recommendation for the use of Care2x for hospitals in developing countries was:

- Care2x is a fully integrated hospital system; and,
- Care2x is web-based and open source.

Care2x is described as a complete system for hospital, and has strong inscriptions of Western style hospital management, since it was first initiated in Germany for German hospitals. For example, all the detailed information during the hospitalization process of a patient needs to be captured. In developing countries, one doctor has to give diagnosis often for more than one hundred patients a day, requesting them to fill up all these details is next to impossible. Next, the network infrastructure in hospitals in developing countries is often not very good, and is not amenable to heavily resource intensive applications like Care2x. For example, the Appointment module (see the screenshot in the Figure 13) allows health workers to schedule appointments for patients. In developed countries where one doctor gives a check up for a few patients a day, the appointment works well. But in developing countries, every doctor has often less than 5 minutes to give a check up for one patient. The time slot is too short for an appointment module to work. Furthermore, the nurse usually does not have time for receiving calls from patients for such kinds of appointments. The last thing is that it can take the clinician quite a lot of valuable time for filling and tracking the appointment, which could be used instead to examine another patient.

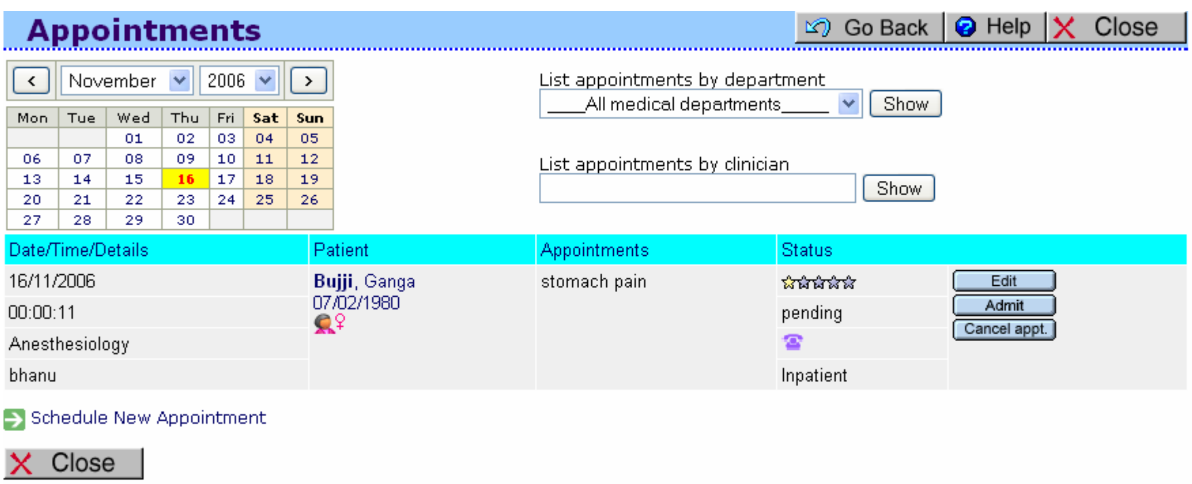


Figure 13: The appointment module in Care2x

I was also part of this Oslo group, and had some prior experience with Care2x in Vietnam. I thus took the lead in orienting the other team members to the application and initiating a discussion on its suitability for Ethiopia. After the presentation and subsequent discussion, the HISP members felt that the application was very complex, and even trying to understand how it worked was a mammoth task, leave aside having the capacity to customize it for Ethiopia. Furthermore, it was felt that all the functionalities provided by Care 2X were not required in Ethiopia, where the initial and most critical need was the patient registration system. The members were at a loss to understand how such a complex application could be stripped down to the needs of Ethiopia. Some more limitations were noted, such as Care2x being fixed for a number of forms and reports, customizing it would mean a direct intervention with the existing source code. Since the forms and reports in Ethiopia are very different from Western hospitals, coupled with the problem of capacity, it was acknowledged that it would be very difficult to effectively customize, given the various constraints.

One alternative approach suggested by the Oslo Professor was that we should focus on specific health programs such as HIV/AIDS or Tuberculosis or Malaria (all of which also require also patient specific system data) rather than try to work with a complete hospital system which was extremely complex. The main argument was to narrow the scope and conditions of the work, and do something which is more practically feasible. Furthermore, we decided that rather than working with Care2x as the starting point, we will build an application from scratch. Given the high need of an ART management system, we decided to focus on the ART management as a starting point. Even while realizing the limitations of a hard coded system, we decided to go that way given the expediency of the situation, where there was the need to quickly come up with a solution, as the Americans were threatening to introduce their own system.

7.1.2 The choice of technologies

There was not much debate surrounding technology choices, because our network was small and comprised only of students from Ethiopia, and myself. Again, for reasons of expediency, Oslo was not included in these discussions as it was felt the discussions would become more theoretical, and revolve around the technologies themselves while diluting the needs of the practical application. We got a signal of interest from the HIV AIDS program manager in Addis region, and he emphasized the need for speed.

We decided to choose PHP instead of Java (the choice for the DHIS 2 platform), as we felt that it was simpler and easier to learn and use. Therefore, it would potentially be more suitable in the context of developing countries like Ethiopia where Java capacity is limited. Since the Ethiopian students had no prior PHP (or Java) capacity, we thought PHP would be easier in terms of knowledge transfer (from me to the others) within the relatively short frame of the semester. An effective knowledge transfer could provide the basis for making the project sustainable locally, and allow it to be further developed even after I leave for my home country. Even in the architecture of the system, we selected a very simple structure comprising of many *.php files where we put all the user interface (html), server scripting (PHP) and client scripting (JavaScript) together. We were subsequently criticized by the experts for this "simple" architecture because it made it difficult to maintain, since there was no separation layer by layer (of the View from the Control etc). We defended our choice based on its appropriateness to the context, issues of capacity, time constraints, and most importantly the need for having the system under local control rather than creating a dependency on Oslo expertise. Apart from these, the most important reason was the user requirements, which only the local group had an understanding of. In the words of one of the local team members: "*Why PHP or Java is important? Most important is to have a system which works*".

We also chose Mysql as database server (using phpMyAdmin as a web based client), which we felt was simple but a very powerful database server. Mysql had been used very widely in both commercial and OSS applications. For the team members to learn this database did not prove to be difficult, as they already had formal education in databases and the underlying concepts.

Compared to the case of DHIS2, we used very little frameworks/tools to build the patient system. It is partly because of the technology we chose (PHP) but partly because we did not want to make it in a layered architecture (separation of view and control).

7.1.3 Project Management

As mentioned earlier, two pairs of Ethiopian groups were created, one working on the DHIS2 customization and the other on this patient based system, which we call the ART team. The two members of this team were interested both in supporting HIV AIDS management in their country, and also eager to learn new OSS technologies. Their level of motivation to engage in the development process was thus high. Furthermore, during development of the system, the team collaborated very closely with one medical doctor in Addis. He was taking a second degree in computer science and had enrolled for the HIS (Health Information Systems) course that was taught by the Oslo Professor. In interacting with him, the informatics students built up a minimal understanding of terms such as "regimen", "opportunistic infections," all of which were crucial for understanding the requirements, and the working of the paper based system.

Mailing list

As our network consisted of Vietnam, Ethiopia and to some extents India and South Africa, we did not set up a mailing list for communication but sent out the messages to all the members in the list. We felt that what we write in the email will be read and replied appropriately because it was to specific people.

Wiki project site

We did not have time to set up a wiki page for the project. And at a very late stage (in June or July 2006), a simple wiki website for ART was released under the domain of HISP India website as we did not have fund to hire a hosting. But the website did not attract so much use.

Version control

There was no repository to keep the source code in our case, it was stored in two laptops of the two members of ART group and the backup was done locally in the laptops and by USB disks. As the development was isolated to few members of the group we thought we should use all of our energy to do the actual work rather than the management work. And the other reason was that we did not have a good Internet connection in Ethiopia. Submitting and checking out source code frequently did not work. When I went home, I also got the updated source code weekly by email. Then I zipped the source code and put it temporarily on the HISP India website so that outsiders who were interested could download it.

7.2 The ART module in Ethiopia

The approach adopted to knowledge transfer can be called "learning by doing". Which we felt was appropriate given the various constraints. The first step was to expose the team to basic principles of web design; database design and simple client-server scenario such as how to add and delete records. The lack of prior experience caused difficulties to the members to start with web design, but this was required as a basic skill to convert the paper forms to web formats. Around database design, an initial learning focus was on how to create columns in the database so as to capture the required fields from the paper forms. Some further concepts concerning relational databases were also discussed amongst the members so as to understand how to better optimize the database schema.

The team always had a lot of questions, wanting to know about many things before starting to write real code. For example, the first PHP page the Ethiopian team tried to create was simply to get a name filled in the text field and show a message: "Hello xxx " with xxx is the input name. The code for this was like this:

```
<?php
echo $_GET['txtName'];
?>
```

Immediately, the Ethiopian team asked for an explaination of *echo*, *$_GET* etc. It could take 15 minutes to explain them all and maybe other 15 minutes to write an actual example including debugging. But we decided to cut off that 15 minutes of explanation, and replaced it by letting the team write that code and run it. It worked and then we gave them small exercises to gradually extend the coding expertise and skills. For example, we asked them to add one more combo box to select the gender of the person whose name was given. After it worked, the exercises for adding other types of controls such as list boxes, check boxes were assigned to them. As a result of this rapid learning by doing approach, in a very short time the team became very confident in coding, and was motivated by seeing what they were able to create.

But it was not easy to convince the team to follow this approach. Used to a hierarchical system of instructors in college, they preferred the academic approach which emphasized the understanding first. But we were keen to insist on abandoning some of this theoretical approach and adopting one that was more practice oriented. And when they could gain a lot of technical skills in a short time, they themselves became supporters and spokespersons of the "learning by doing" approach.

The team felt that it was very difficult to learn the technologies, more precisely to do real coding. Software development requires not only programming skills, but also analysis and design. The analysis skills were practiced by letting the team use the real business process, and try to find solutions. The field trips to the clinic every week to collect more and more requirements helped them to practically relate needs on the ground with their own skills and demands. Firstly, we selected important forms and reports to start the analysis part. Small pieces of analysis results were mapped into web form and the table structure in the database. While we can be criticized for our messy approach, this piece by piece approach was seen to be more appropriate with the limited capacity and time constraints,.

During this whole process, I consciously did not do any real coding, and only provided the Ethiopian students with technical direction and encouraged them to solve problems themselves. For example, the very common problem they faced during the development of the software was either the PHP syntax error or the un-expected response of the function. In PHP, the syntax error is easy to identify as PHP will describe the error and the line number. In stead of helping the team to fix the syntax error, we asked them to go to the line number in the source code and see if they had made any syntax mistakes. The second common bug (un-expected response) was logical in nature and was more complex to show than the syntax bug. We asked them to write one more statement (echo statement) in the page to show the values of some variables and based on that, they would be able to easily find out the cause of bug. At first, the team was not very pleased with this approach. They also thought that I did not give them enough care and help, and wanted me to directly fix the bugs so that they could quickly continue the development since bug fixing would often take hours which they felt was a waste of time. Even though I could have fixed the bugs easily I tried to convince them of its importance by telling: "*Imagine that you will be working alone since next month, then how would you solve the problem yourselves?*" Slowly, they realized the importance of fixing the bugs themselves and with time became more adept at it, taking lesser and lesser time to address the routine bugs.

In addition to this working in the lab, the team members spent a lot of time socially in dinners and other settings. These interactions helped to develop a strong social glue, which contributed to an easy flow of questions and answers between us, and thus helped in the development of capacity.

After 2 weeks of intensive development, a prototype of the ART system was released, including the implementation of the first form, which was to keep track of the ART clients who were receiving treatment. This form allowed addition, deletion and editing of ART clients. This was a rather complex form with more than 50 different data fields. This form is presented in the Figure 14 below:

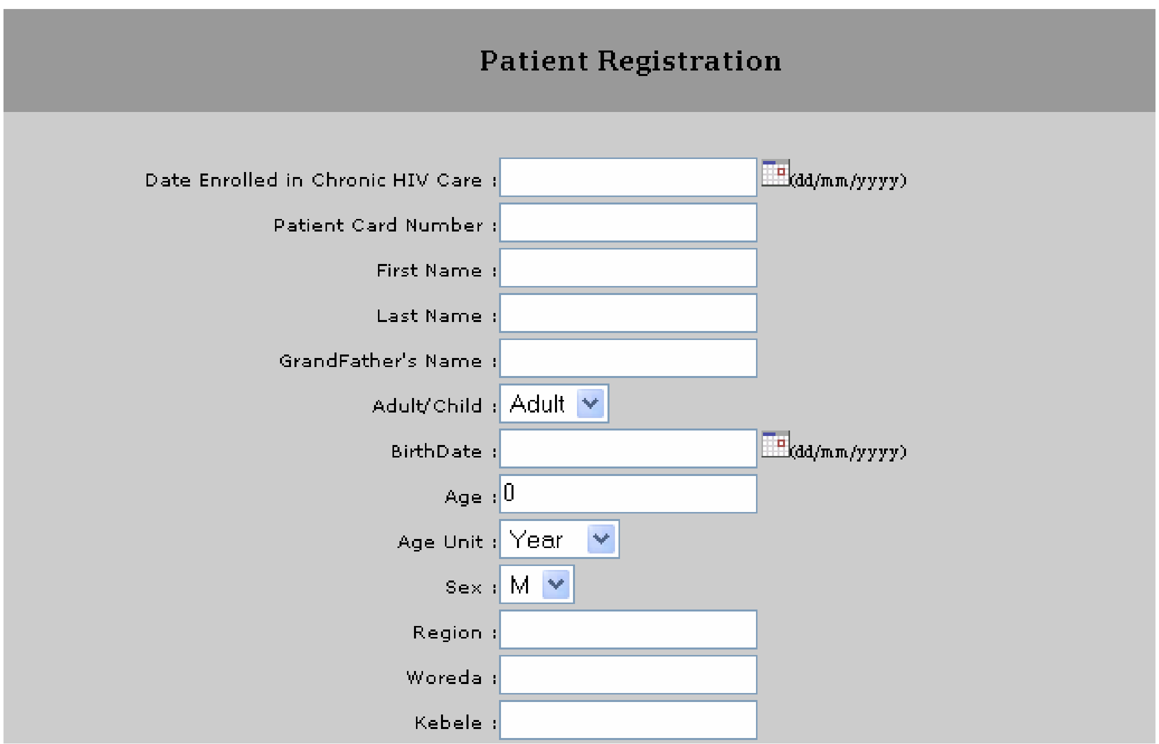


Figure 14: The ART registration form in the first working prototype of ART module

The form in the Figure 14 shows controls in the vertical layout. It is very simple compared to the current version of ART. For example, all elements of the web-based form are put vertically and not grouped into a section which is easy for design but not user friendly. Second, the validation rules were still limited, only allowing the validation field by field not relations between fields.

The successful implementation of this form gave the team a big boost and encouragement, and a sense of confidence that they could do the work. After that, the team started to work with reports, and also the forms for pre-ART and follow up. After 2 more intensive weeks, we had implemented three forms and one report, which formed the basis of the first version which was presented to the program manager who was supporting us. On his suggestion, we tried to get the first demo quickly installed in a clinic, so as to to get user feedback which could be incorporated into making improvements on the system. The positive feedback received from the manager gave further confidence to the team, who started working on multiple tasks such as improving the: user interface, incorporating validation rules, and adding menus to facilitate data entry…

7.3 The off-face continuity of development

After two months of intensive work, I returned to my home country, and the Ethiopian team started to work alone. The support mechanism now needed to necessarily shift from a face-to-face to remote mode, including through the use of email and instant messenger services. Requests from the Ethiopian team to me were very few, which either could be interpreted as a good sign that they were becoming independent, or alternatively they did not prefer electronic means of communication. Also, some of the questions were difficult to formulate and express over the electronic medium. For example, concerning user management function, session is a difficult concept in web application. Several times the team said that the users required the user management function so that only authoritized people can use the system. But we could not solve this before the second chance of co-located work.

The first version of ART was completed before I came back to my home country. And due to the Internet constraints in Ethiopia, it was not easy for me to upload it and make it online so that other developers of the HISP network from Oslo could also get involved. We had to ask the Indian team to help us upload it. When I came back home, the Ethiopian team sent the new version to me by email and then i deployed it in Vietnam. I downloaded the package which included the script to create the database from email, unzip it and using SSH to connect to a server rented by the HISP India team to place the package. The database was also deployed from Vietnam by me. We can realize from this that the actual development took place in Ethiopia but the deployment of it has never been done by the Ethiopian team. Therefore, there was always a mismatch between the version being worked on in Ethiopia and the one available in Vietnam. While formal mechanisms of version control would have helped, those tools were difficult to use because of Internet constraints and also learning a version control tool was in itself a complex task.

One month later, the next release of ART called ART 1.l was issued, which contained lots of improvements in terms of functionality and user interface. The Ethiopian team also informed that the system had been installed in two ART clinics in Addis Ababa and the process of entering patient data had started.

Simultaneously, while in Vietnam, I had also completed the first version of a flexible system which tried to address the limitations of the hard coded nature of the ART system, and also to enable a rapid expansion of functionality. Since the flexible system allowed end-users define the form as per their needs, it would enable the rapid generation of forms required for the other modules for HIV AIDS management such as Pregnant Mother to Child Transmission and Opportunistic Infections. This would help us realize our longer term vision of developing an Integrated HIV/AIDS management system including all the HIV/AIDS related programs. This flexible system was developed by only me, and i planned to transfer this system to Ethiopia to help develop the integrated system.

7.4 2nd chance of co-located work

In the end of May, the Indian coordinator and I went back to Addis to attend another conference on Africa e-learning. This provided another chance to further the face to face collaboration, and resolve issues which were not possible through internet, especially the integration of the ART and flexible systems. The first health program we tried to adapt was the VCT program (Voluntary Counseling and Testing). While the trial integration worked fine with respect to form generation, problems were experienced in generating reports because the database structure of the flexible system was more abstract then the fixed one. For example, the flexible system used only one table called datavalue to store all data of different forms. Values of fields in one form were stored as rows. In the ART system if the form had 5 elements such as patient id, patient name, sex, birth date, address, the data of this patient would be stored as 5 different rows in the datavalue table (see the Figure 15). These very different structures created serious difficulties in integration.

←T→	ID	FormInstance	FormID	ElementID	Value	PatientID
	292	1	11	45	28/05/2006	479
	293	1	1	1	Nguyen Ngoc Thanh	479
	294	1	1	14	37/5B Trung My Tay, Ho Chi Minh city	479

Figure 15: Database structure to store data value of one record. The one-step-more abstract layer made the flexible system more complex to generate reports and integrate with ART module

After a very short time of one week, we all left, and the Ethiopians were supposed to continue to try and make the integration work. Later, we were informed that they could not do that because of various technical problems, and also because the field implementation of the ART was every day throwing up new requirements, and they had to focus their energies into fulfilling those. As the system was put in to real use, the process of entering the data was started. In the first clinic, they hired a data clerk to enter all the patients' data available in the clinic, which was more than 10,000 records. Therefore, the team had the responsibility of supporting this process. This was a daily task and the development work would never end. The actual use of the software was also ongoing. We received live feedbacks from the users which was very necessary to complete the system. But the commitment to support use of the software made the team very busy with bug fixing. Therefore, there was very little time for functionality expansion.

After this second stretch of intensive work, I came back to Vietnam. During this time, I was requested to build a patient based system for a clinic in Zanzibar. This clinic had used an Access based system to capture information of the patients and their visits. This was a simple system with approximately 15000 patient records captured. I started porting this Access based system to my flexible system and migrating the legacy to the new format. The entry form was easy to define as the flexible system already supported these functionalities. But for the data migration, I had to write a new tool to read data from Access and insert to the flexible system's database. This was really a difficult task because the data structure in the Access based system was different from the one in the flexible system. This limitation in the design of data structure in the flexible system made me decide to give up the first prototype and start building a new one in the later phase. This is discussed in next section.

The Figure 16 shows the data entry form for patient admission in the early version of the flexible system which was a simple vertical layout:

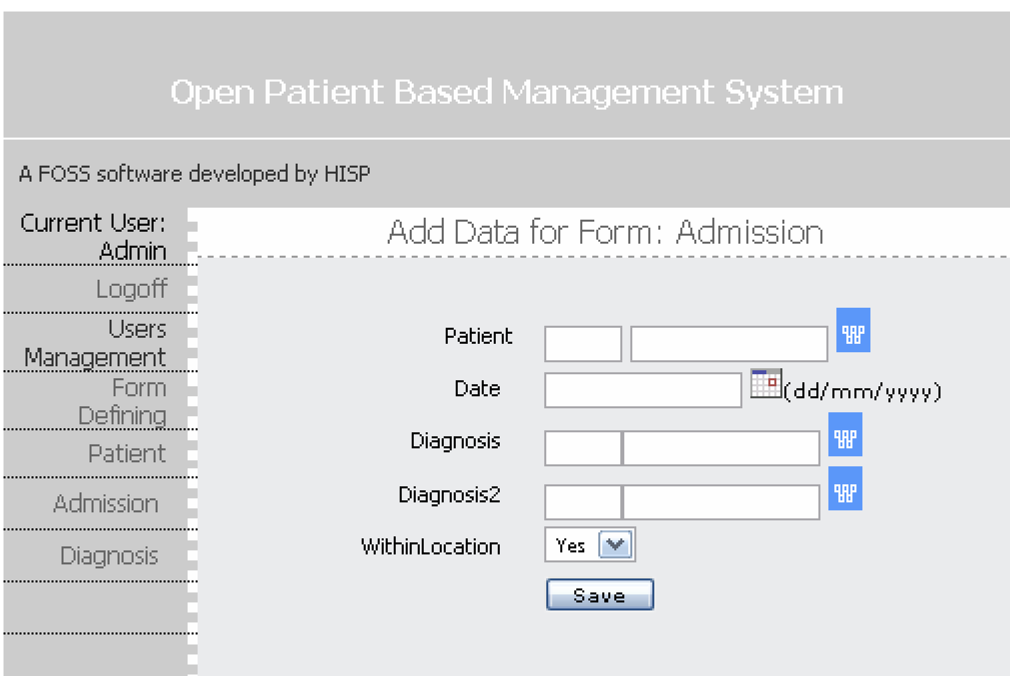


Figure 16: The data entry form

And the database structure to store the data in that entry form is shown in Figure 17:

ID	FormInstance	FormID	ElementID	Value	PatientID
2463	1232	1	1	2004-09-15 00:00:00	747
2464	1232	1	2	380	747
2465	1233	1	1	2004-09-15 00:00:00	999
2466	1233	1	2	694	999
2467	1234	1	1	2004-09-16 00:00:00	123
2468	1234	1	2	641	123
2469	1235	1	1	2004-09-16 00:00:00	310
2470	1235	1	2	641	310
2471	1236	1	1	2004-09-16 00:00:00	604
2472	1236	1	2	16	604

Figure 17: The database structure for storing data.

7.5 3rd chance of co-located work

In October 2006, there was another chance for developers in Ethiopia and me to work together during the DHIS developer workshop and International DHIS conference in South Africa. By this time, the status of the ART module had become relatively mature and it was being used in a number of clinics in Ethiopia. Figure 18 shows the Intake form of the latest version of ART module with a lot of improvement in terms of functionality and user-interface:

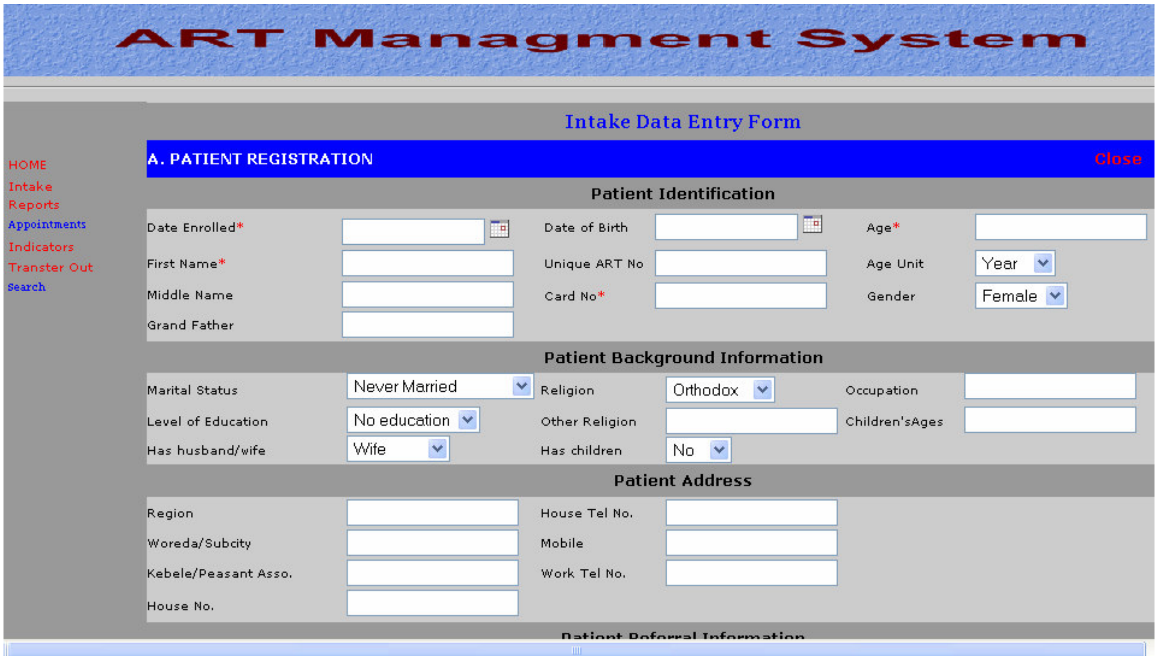


Figure 18: The Intake form of the latest version of ART module

One of the Ethiopian team members made a demonstration of the ART system from Ethiopia which really impressed the audience, who expressed surprise that the system had been developed in only 4-5 months, and that it was practically working well in the real settings of the Ethiopian clinics. We also heard several comments from the audience about how similar system development efforts of patient specific systems had failed, and had led to a huge wastage of resources. A key reason for these failures was that the development process had been very abstract and could not come to grips with understanding the complexities surrounding the ART work practices.

However, there was a flip side to the rapid development process adopted in Ethiopia. We realized the limits of this when we started to explore with a South African group who were interested in adapting the same system for a hospital in Pretoria. When our code was examined by a computer scientist in Ethiopia, who was both an expert in PHP and OSS technologies in general, he gave valuable feedback on the limits of the existing architecture, which he felt was like a "ball of mud" which was difficult to maintain and almost impossible to share. His key criticisms are summarized in the table 18 below:

No	Criticism	Reason
1	Version control: There is no version control during the development of the software.	Therefore, no back up of history of versions. All source code can disappear if the hard drives of the personal computers get damaged. It is not possible for multiple developers to work at the same time.
2	Coding style: No systematic coding style. Name of variables and structure of the pages were ad-hoc.	Very hard for new developer to follow up
3	Security: Data was not encrypted before storing in the database.	Confidentiality is a critical issue in HIV/AIDS treatment. He showed us how the un-encrypted database files can be read. He demonstrated the scenario in which one hospital in Pretoria was robbed and would express all data.
4	User interface/business logic separation. The length of files in ART system can be up to thousand lines of code and .then too be long	It is very hard to read and maintain the system. It needs to be separated by layers. He suggested the MVC (Model-View-Controller) architecture and several frameworks supporting this model such as Symphony.
5	Database abstraction. Another statistic made by the Pretoria FOSS expert says that there are more than 700 calls to database using my_query() in the whole system.	Hard to change database management system

Table 18: The list of improvement suggested by one FOSS expert from South Africa

His conclusion can be summarized in this excerpt from an email sent by him: "*it (the ART module) may will not (as a software artifact) be suitable for use in SA or perhaps anywhere else*". As a result of these criticisms, which became clear during a workshop in Pretoria, it was agreed that the database would be restructured, and the entire application would be rapidly recreated on Symphony framework which would help to make the system more shareable.

After 3 days of the workshop, I left South Africa, but the Ethiopian team stayed in Pretoria for learning the framework and doing some actual working. In the same time, there was a group of students in University of Oslo working on porting the current ART system to Java frameworks.

One month after the Pretoria workshop, a new version of the system with improvements in accumulative tables was released. This new feature allowed using the ART system for generating the monthly report right away without requiring all patient records to be entered. This was still based on other previous versions, and not on the new Symphony framework.

7.6 Post Pretoria workshop – 3 directions to pursue the same goal

However, at this time, there were significant changes in the development of the patient based system. There were three groups in the HISP network who pursued three different ways:

- The Oslo team with Masters students re-develop a Java-based for the ART module
- The Ethiopian team with the support from South Africa co-develop a flexible system based on Symfony and OpenMRS data model
- I myself continued improving the flexible system and customized it for several places. I will discuss this process in the next section

7.7 The development of the flexible system (openEPR) in Vietnam

After coming back to Vietnam, I started to improve the flexible system based on the feedback and lessons learned during the workshops in South Africa. First, I gave up the design of the database schema as it created difficulties in making reports. I decided to use another approach which was to automatically generate a new table in the database every time a form was created. The database schema of applications would then become similar to other usual systems such as the ART module etc.

The real form layout was also added to replace the vertical layout of form elements, making the system become more attractive to users. The Figure 19 and 20 show the data entry form of child health records and the respective database schema for the form:

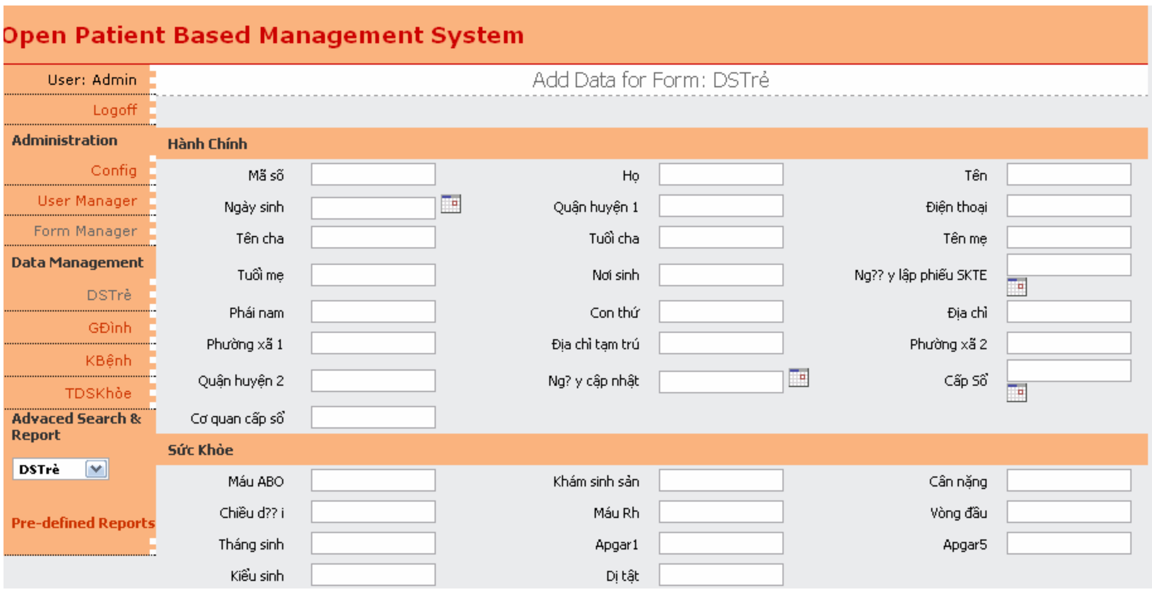


Figure 19: The data entry form for child health record

Field	Type	Collation	Attributes	Null	Default	Extra	Action
id	int(11)			No		auto_increment	
Maso	varchar(10)	utf8_unicode_ci		Yes	NULL		
NgayCapNhat	date			Yes	NULL		
NgayLapPhieuSKTE	datetime			Yes	NULL		
Ho	varchar(50)	utf8_unicode_ci		Yes	NULL		
Ten	varchar(255)	latin1_swedish_ci		Yes	NULL		
Ngaysinh	date			Yes	NULL		
Noisinh	varchar(50)	utf8_unicode_ci		Yes	NULL		
PNam	varchar(10)	utf8_unicode_ci		Yes	NULL		
Conthu	tinyint(4)			Yes	NULL		
Diachi	varchar(50)	latin1_swedish_ci		Yes	NULL		
PX1	varchar(50)	utf8_unicode_ci		Yes	NULL		
QH1	varchar(30)	utf8_unicode_ci		Yes	NULL		
Diachitamtru	varchar(50)	utf8_unicode_ci		Yes	NULL		
PX2	varchar(50)	utf8_unicode_ci		Yes	NULL		
QH2	varchar(30)	utf8_unicode_ci		Yes	NULL		
DienThoai	varchar(30)	utf8_unicode_ci		Yes	NULL		
Tencha	varchar(50)	utf8_unicode_ci		Yes	NULL		
Tuoicha	smallint(6)			Yes	NULL		

Figure 20: The table structure for child health records

Other improvements were also added for the flexible system. For example, it had the data element group and sorting order. Also, validation rules were introduced to increase the data quality, as showed in the Figure 21:

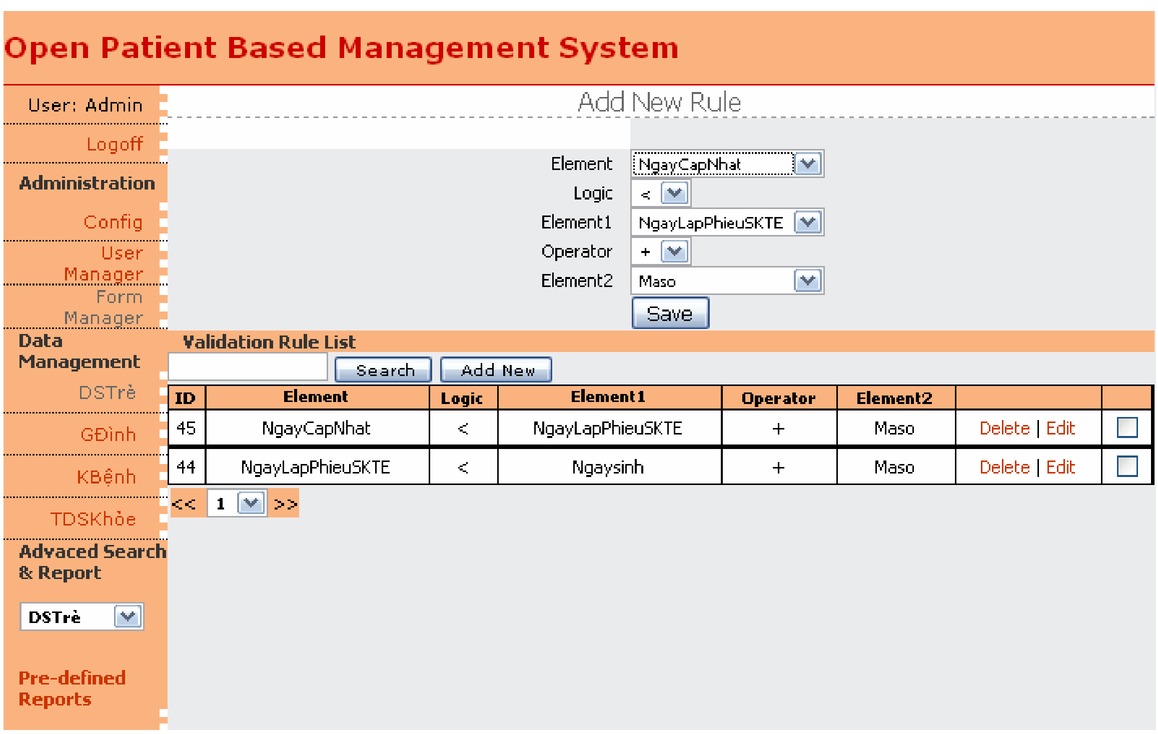


Figure 21: The GUI for defining validation rules

Realizing the great potential of the flexible system, I had a discussion with the Deputy Director of Mother and Child health center in Ho Chi Minh City to put the system into use.

Photo 6: The statistical staff in Mother and Child Health Center showing the forms and reports related to mother and child health records.

After the discussion, I started customizing the system for patient management in the center. The work was finished shortly after that. I made a presentation about the system for the Deputy Director and IT staff. They were interested in the system and eager to pilot it in the reception department of the center.

The real implementation of the system however needs more resources for user support and define more reports. And also I wanted to build the capacity for the team to deal with this system for which I suggested getting one member of the DHIS2 team to work with the patient based system. However, the team preferred to work with Java and no one really wanted to work with a system which was developed in PHP. On this issue, a team member emailed me:

"We need to discuss more about this task. I had a look into the code of your patients system around 1 month ago. And I'm really sorry to say that I don't believe it. If we decided to use it, we have to develop much much more."
(*DHIS2 mailing list*, 2006)

And since that time was the end of the year the Deputy Director was too busy with his work in the center to push the implementation. Lacking both the political and technical support, the system had not been put into implementation.

CHAPTER 8: ANALYSIS

This chapter focuses on analysing the findings presented in the earlier chapters of the case study. The case study discussion shows that there have been very different outcomes of applying the same technology (DHIS2 and patient based) in the health care sector of the two countries. For example, in the case of DHIS2, the HISP Vietnam team was quite successful with the adaptation and customization process and the system has been rolled out into the whole city, while the HISP Ethiopia team is still struggling with making the same system work. The theoretical perspective presented in the literature review chapter will be drawn upon to analyse what and how the social conditions can be seen to shape these outcomes. As discussed in the methods chapter, a 2x2 matrix is created to demonstrate the framework of the analysis, showed in the Table 19:

<table>
<tr><th></th><th colspan="2">DHIS2</th><th>Patient based system</th></tr>
<tr><td>Ethiopia</td><td rowspan="2">1 Comparison of DHIS2 between Ethiopia and Vietnam</td><td>3 Comparison between two systems in Ethiopia</td><td rowspan="2">2 Comparison of Patient based system between Ethiopia and Vietnam</td></tr>
<tr><td>Vietnam</td><td>4 Comparison between two systems in Vietnam</td></tr>
</table>

Table 19: Cross system and cross country comparisons

The analysis process is made along the following dimensions:

- (1) Comparison of DHIS2 between Ethiopia and Vietnam,
- (2) The Patient based system between Ethiopia and Vietnam.
- (3) DHIS2 and the Patient based system in Ethiopia, and
- (4) DHIS2 and the Patient based system in Vietnam.

These four sets of analysis are now presented

8.1 DHIS2: cross country analysis

I first start with discussing the differences in outcomes of applying DHIS2 in two countries summarized in the Table 20, followed by a discussion on the analytical relation between social context and these outcomes.

DHIS2	Ethiopia	Vietnam
Development	**Technical capacity** - Only one developer is capable of dealing with the core technology. **Local customization** - A local report module was developed based on JSP - The Morbidity and Mortality module was first developed in JSP, later ported to Webwork. - Amharic language was added	**Technical capacity** A team of 4 developers is capable of dealing with DHIS2. One Masters student has high level expenditure with the core technologies **Contributed to the core DHIS2 development** - Extended the User and Security module, Datamart and Dataview module. - Developed the internationalization module to support multi-languages based on i18n[31]. **Local customization** - Developed the installation package for scaling up the implementation including installation file, user manual, video training etc. - Developed the local Report module based on Jasper

[31] The W3C Internationalization - http://www.w3.org/International/

Use	- Two MOUs (Memorandum of Understanding) were signed with two regions but the implementation could not take place because the system was not ready. - DHIS2 database was prepared for Tigray region by migrating the data from DHIS1.4	- DHIS2 was customized for Mother and Child health programs in Ho Chi Minh City. - DHIS2 has been introduced in one city (24 districts and 6 hospitals) with Mother and Child Health Programs. Data entry and report generation are going on routinely.

Table 20: The summary of outcomes of DHIS2 in the two countries

Team structure and organization

In case of DHIS2, in Vietnam, the team comprised of 3 paid developers who had a one year experience doing internships in this project. In comparison, Ethiopia had only two Masters students without strong experience in Java and related frameworks comprising DHIS2. Also, the Vietnamese team was supported by a facilitator who was dedicated to field support, which helped the developers to focus exclusively on the development. The Ethiopian team comprised of Masters students who had to study and at the same time do the development, and hence could not provide the full time support that was necessary.

Technical capacity

In Ethiopia, the team had no prior experience in Java and the various frameworks that were involved. In Vietnam, three paid developers in the DHIS2 team graduated from a university which focused on OSS technologies such as Java, Hibernate etc. They spent a long time getting familiar with DHIS2 technologies during their internships. They also received extensive training and support from the Norwegian Masters students who came to Vietnam. All these conditions strengthened their technical capacity as compared to the Ethiopian team. This difference contributed to the varying outcomes of DHIS2 in the two countries.

Nature of the health sector

The differences in the nature of health care sector in the two countries also played an important role in shaping the outcomes of DHIS2. In Vietnam, the introduction of 15-forms with minimum data set initiated by the Ministry of Health had already contributed to the standardization and reforms of the reporting system in the other health programs. Also, since the Ministry of Health appointed Medisoft to be the official software to be used in all hospitals in Vietnam, it reflected the strong support of the government for the computerization of health information systems. These conditions helped the implementation of DHIS2 in Vietnam to be more successful than in Ethiopia.

The political support for OSS in Vietnam also created the advantage for applying DHIS2, an OSS application. The decision on fostering OSS was signed by the Prime Minister which helped to create the base for OSS. Therefore, it was easier to convince health managers to try DHIS2 than other proprietary based solutions.

In Ethiopia, the health sector was still in the reform process including the standardizing of data elements and defining of core indicators. The work was in progress, which created difficulties for the DHIS2 as the formal government decision was not yet in place. In respect to OSS, although ICTs were recognized as important for development, Ethiopian government did not have any stated policy to promote OSS. Therefore, the OSS nature of DHIS2 did not serve as an explicit advantage over proprietary software.

Technology introduction process

The process of technology introduction in the cases of DHIS2 in Ethiopia and Vietnam was also different. In Vietnam, it took place since early 2004, nearly at the same time as in Oslo. In contrast, in Ethiopia the process only started in February 2006, one year after Vietnam. DHIS2 in Vietnam started with the collaboration of local companies in Ho Chi Minh City and Hue City, and several teams were trained in DHIS2 technologies. The team of three developers who worked for the project after their studies was also trained in these technologies when they attended seminars given by the Norwegian Masters students. The long process of technology introduction in the case of DHIS2 helped to build capacity in Vietnam, a process largely absent in Ethiopia.

The two teams also approached the introduction process quite differently. In Vietnam, the team started to customize DHIS2 and adapt it for only one health program at first. This approach may be criticized as contributing to the fragmentation of the health information system, but in the context of Ho Chi Minh city it appeared an appropriate approach. Ho Chi Minh City is the biggest and most developed city in Vietnam and thus requires more detailed data compared to other provinces. Every health program is owned by a different office in the city and also at the district level, different offices are responsible for collecting and reporting data related to their respective programs. Therefore, if the implementation process had included all health programs at the same time, it would increase the complexity as more reports would need to be designed and more training and field support would be requested. Given the limitations of time and resources, the approach of focusing on only one health program, made it work smoothly. A gradual extension into other health programs helps to reduce the complexity of the initial implementation and allows for a gradual scaling process. This approach was also expected to create more credibility with health managers of other programs, as they were first showed the success of the project in the complex Mother and Child program in one of the biggest cities of the country.

In Ethiopia, the focus on adapting the DHIS2 for all the reporting systems created difficulties for the team. For example, to prepare the database for Tigray, more than 1000 data elements needed to be included and 40 report templates created. With limited resources and time, magnified by the problem of poor capacity, the team could not deal with the complexity of the software and the reporting systems.

In addition to the above conditions that have been identified to be relevant by prior SCOT research, other aspects were also found to be important. These are now discussed.

Technological infrastructure

"*Downloading and building DHIS2 is really a "nightmare*", was the description of a HISP coordinator in India, even though India had much better Internet facilities compared to Ethiopia. I use this quote to describe how severely the infrastructure affects the development of OSS, a situation magnified in developing countries. The HISP project in Ethiopia was lucky to have a base in Addis Ababa University, a national university. However, the connectivity was also very poor. For example, downloading speed was very slow, and it was broken many times. Also, several pages were inaccessible from Ethiopia such as Gmail etc.

The way in which DHIS2 was designed, makes it depend very much on Internet. For example, DHIS2 uses Subversion for concurrent version control. This software could not work through the proxy of the university network. Sometimes it could work but it was not stable. We had to ask the Oslo team to checkout the source code and send it to us by email. After downloading the source code we faced other problems. DHIS2 uses Maven to manage supporting libraries and to build project file. All packages needed for the project would be downloaded from the central repository located on the Internet (http://maven.org) automatically by Maven. However, the proxy setting prevented us from doing that. So we had to ask the Oslo team to zip the libraries and send it to us by email. This task was repeated every time there was a change in the source code. The complication of the process of "download-zip-send" discouraged the team in Ethiopia from receiving the latest version to work with. This created coordination problems between them and other nodes like Vietnam and India who would typically be working on more advanced versions than Ethiopia.

In Vietnam, I experienced the same thing when an Oslo Masters student came to Vietnam to train and co-develop with the interns in ComA, during the period from January 2005 to June 2005. He also had problems with the Internet in committing and checking out the source code using Subversion. According to the network's policy in the company, the computers which wanted to send and receive source code needed to have a static IP address. This was only obtained by getting the approval from the manager. The formality usually took a long time, creating difficulties for locally based people to join the global development of DHIS2 and communicate with the Oslo team.

In the fall semester 2005, when HISP started collaboration with Nong Lam university, another Masters student sent from Oslo could not do any work because the network there was so poor. One "breakthrough" decision was made which was to use the house of one of the students as the working office and subscribe to an ADSL. After having this Internet connection, the involvement of the Vietnam team with the global network was significantly increased. From this point of time, more commitments to the central repository and more email exchanges with the list were made by the Vietnamese team than before. The improvement in infrastructure both in terms of capacity and access arrangement really made a big change here.

Improving Internet connectivity was relatively easy in Vietnam because it has a very fast growth rate of ICT due to the economic boom. Wide bandwidth Internet has become more common and cheaper. While Vietnam represents a special case of ICT growth, the similar situation was impossible in Ethiopia. The banning of SMS (Short Message Service) in Ethiopia for political reason also impeded communication process, a means which was used commonly in Vietnam for short message exchange.

The same technology, DHIS2, produced very different results due to the differences in infrastructure. The team in Vietnam with a better Internet connectivity could easily join the development process and get full support from the Oslo team. Therefore, it was possible to customize and maintain the local implementation, which was in contrast to the situation in Ethiopia.

The differences in infrastructure were not the only reason contributing to the differences. Another important reason was the state of integration between development and use. This is now discussed.

Integration between development and use

The empirical findings in the case studies point out to the key role of the state of integration between development and use. In Vietnam, the efforts to put DHIS2 into use have produced relatively positive results. In May 2006, the first customized version of DHIS2 was installed in several pilot districts. Rolling out to the whole city in August 2006 was another important step to speed up the customization process. To achieve that, the internationalization and the installation module were developed. The successful scaling up to the whole city for the Mother Health Program also helped to gain more political support from other health programs such as child health program. Impressed by the result of the implementation of the Mother Health Program, the Child Health managers allowed the team to customize and scale up the implementation of their program to the whole city in December 2006. And later, in February 2007, the immunization program was also involved in a similar manner.

The implementation and use dynamics played an important role to "motivate" the development process. In Vietnam, several development tasks were done due to the pressure of the implementation, for example, the installation package, the improvements of report module, security and use module, GIS etc. In Ethiopia, the stage of use of DHIS2 had not taken place although two MOUs with 2 regions had been signed. However, without actual use, and minus the pressure from the implementation context, the development process could not gain adequate momentum.

The support from the network

The support from the network in the two countries was different. DHIS2 in Vietnam earned success, due to significant investments from Oslo. The continuous support from Oslo by sending Masters students to Vietnam who worked closely with the team for long periods of time, set up the foundation for capacity building and supported the learning process. This focused investment from Oslo helped to create a pool of local resources with capacity to deal with DHIS2 customization and also contribute to core development. Compared to Vietnam, the team in Ethiopia received very little support from Oslo. There was only one visit of a developer from Oslo in August and the time was very short (10 days). In contrast, Vietnam has had at least 7 developers spend an average of 3 months each. The longer stay also helped the visiting developers to gain more understanding of the local context, which then helped them to subsequently support the team when they returned to Oslo.

In this section, I have discussed how different aspects of social context affected the outcomes around DHIS2. In the next section, the cross country analysis of the patient based system is provided, again using the theoretical framework as the point of departure.

8.2 The patient based system: cross country analysis

The outcomes in this case are relatively opposed to the case of DHIS2, as Ethiopia has shown more positive outcomes than Vietnam. In the Table 21, I first summarize the outcomes and then discuss the social context.

DHIS2	Ethiopia	Vietnam
Development	- A working module for ART management has been created locally by the two developers - Efforts are ongoing to port it to new framework (Symfony) with support from South Africa - A team of the developers has been created, who are capable to deal with the various technologies and the business processes in ART domain	- A flexible system has been developed by only me
Use	- An MOU was signed between	- The system was adapted on a

	HISP and Addis Ababa regions to implement the system in 67 ART clinics in Ethiopia - UNAIDS now approved a project to link ART system with DHIS2	pilot basis for a clinic in Zanzibar, and by the child health and mother health records in Ho Chi Minh City. - The system has not been used because of lack of political and technical support.

Table 21: Summary of outcomes in case of the Patient based system between two countries.

The summary table indicates the great differences in outcomes between the two countries. I analyze the reason based on the social construction perspective.

Team structure and organization

In the case of patient based system, the team in Ethiopia was organized better than Vietnam which comprised only the author of the thesis. The team in Ethiopia comprised of 2 Masters students and a medical doctor who was taking his Masters in Informatics. This health domain expert played an important role to provide domain knowledge needed for building the patient based system. This team later received another Masters student in public health, who supported the training, implementation and scaling process. The combination of IT and health was crucial in developing this domain specific software. While in Vietnam, the team consisted of only me, responsible for developing the software, making other logistic arrangements and also working on my Masters thesis at the same time. The structure of the team and the skill mix was more appropriate in Ethiopia.

In Ethiopia, there was an effective and integrated approach to the development and implementation. For example, the learning by doing approach used to build technical skills concretely helped to link the development and use. The prototyping approach in close collaboration with end users created energy for continuously improving the system, and at the same time developed the buy-in of the use.

In Vietnam, although the system has been customized for the Mother and Child center, the "real installation" did not take place to catalyze the development. The "not-having-real-use" status placed the system in to the state of "hibernate", and new functionalities were added very slowly. For example, the first version of the flexible system was released in May 2006 but it was only in November 2006, that grouping and sorting functionalities were added. In both cases, the link between ART and DHIS2 has not been carried due to the lack of demand from the users for such integration.

Technical capacity

The team working with the Patient based system in Ethiopia started with a nearly null technical capacity. But through the "learning by doing" process, the team actively learnt the new technologies needed for their work. In the beginning, I played the role of a technical enabler who helped the team get acquainted with the technologies and software development process. While it could normally be assumed that the better technical capacity would lead to a better outcome, as analyzed in the DHIS2 case. This was not the case in the Patient based system despite Vietnamese having higher technical capacity. Technical competence was thus not necessary to determine the different outcomes.

The low technical capacity could be improved by adopting an appropriate "learning process", as done in Ethiopia, and supported through a prototyping process. Creating a link with South Africa in the later phase of the process helped to expand the network and further support both, for the technical capacity building and the scope of use. The support from this link helped the team to improve their system significantly by redesigning the database and porting it to a new framework (Symfony). This link was very necessary because the team expanded its network of support and reduced reliance only on the Vietnamese node. This has helped both the sustainability and scalability of the system.

Nature of the health sector – the use context

In the case of the Patient based system, although there was no political support in fostering OSS, generally the team had strong support by the Addis Ababa HIV/AIDS Program Manager. Scaling up ART is a national priority given the large scale effects of HIV/AIDS on the population. HIV/AIDS currently affects approximately 10% of the population (http://www.eldis.org/static/DOC14620.htm). The ART module developed by the HISP team came at the right time. The health managers in Addis Ababa region had been waiting for a long time for an ART management system to come, and so when it was introduced, it gained strong political support.

In Vietnam, first, the prior approval of the Ministry of Health to use Medisoft as the official software in hospitals created difficulties for introducing the flexible patient based system. Second, the lack of political support from health managers in Ho Chi Minh City impeded the chance for the system to be tested and continuously improved. Several rounds of meetings and presentations were made but still no MOU has been signed. Implementation of the software in health care needs as much of political brokering as technical development.

Technology introduction process

In Ethiopia, the system right after development was installed in the clinics for users to try it out and give feedback. This approach created a strong mechanism to get the system up and to support its roll-out. Also, its simple technological choice and design helped it to start and quickly get some momentum. A cultivation approach was employed to build the system, started by making one form, then slowly growing it to two and more forms, and ended with the full functional system. If the technology introduction process had started with a complicated design, the outcomes may have been different.

In contrast, in Vietnam, the flexible system only existed in discussions with the health managers and in various demonstrations, and not installed in use situation. This prevented the system to become embedded in actual context where it could be developed further based on feedback and pressure from users.

Technological infrastructure

The technical infrastructure in Ethiopia although was worse than in Vietnam but it did not reverse the results in relation to this system as the choice and design of the system had reduced the dependence on Internet. For example, the programming language used to build the Patient based system was PHP, which was easier to learn than Java. All necessary libraries were provided together with PHP, thus there is no need for Internet to download. In the case of the Patient based system, the technical infrastructure did not affect the development and the outcomes.

Integration between development and use

In Ethiopia, at the early stage of development of the Patient based system, the team had focused on building participation of the end-users. The first prototype after two weeks of development was brought to the clinic to get user feedback, and the first version was introduced after two months even though it was still in its infant stage with limited functionalities.

To make the implementation become more useful, in Ethiopia, legacy data was entered for more than 10 thousand patient records, which also placed real pressure on the two developers. They worked hard and went to the clinic any time the users needed support. Because of the pressure of use, every bug had to be fixed as soon as possible to remove any delays in using the system. After having all patient data in the system, the reports were generated from the system instead manual generation. For example, a statistician said that to prepare a monthly report, it took them two to three days and to prepare a complicated report such as a Cohort, it could take months. Now these reports were generated in seconds. The concrete benefits of the system in helping health workers and health managers encouraged the development team to continuously improve the system, and simultaneously get it increasingly embedded in the use context.

The success in Addis Ababa created a good credibility for the system for the other regions such as Oromia. They requested implementation of the system in clinics in this region. Due to the shortage of resources, an installation package was made to speed up the process of scaling. And later, an accumulation function was added to help in generating the reports without entering the whole data. The use of software played the role of an engine driving the development.

In contrast, in Vietnam, although the system was designed to be flexible, capable to fit different requirements, it has not produced any concrete results in terms of use. Moreover, this weakness contributed to the slow development process as compared to Ethiopia. The first version of the flexible system was completed in June 2006 but it was only in October, that important functionalities such as grid layout, grouping, and validation were added.

The integration between development and use plays an important role in leading to different outcomes between Ethiopia and Vietnam. Wherever this link is stronger, more favorable outcomes are created.

The support from the network

I differentiate the support at two levels of network – global and local

Global

The support from the network was also important in shaping the outcomes. The team in Ethiopia has built an effective network through which the learning process was maintained and sustained. First, they received the technical training from the Vietnamese student. To deal with other challenges of scale up, the team created a link with South Africa to receive more technical and financial support. The team traveled to South Africa twice to get support in porting the system to Symphony framework.

In Vietnam, the development of the flexible system was isolated within a team of only one developer. Failure in getting involved in network affected greatly the development of the system.

Local

Beside the global, the role of the local network in supporting the development and use of OSS was also important. In Ethiopia, the team built a learning network through a mixed team with both IT and health experts. This team later was supplemented by one Masters student majored in public health, creating a stronger link between Informatics and Health. This local network played an important role in making the learning process become effective because the availability of such local expertise is common and the ability of context understandings is higher. In Vietnam, the existence of both global and local network was limited and thus support was poor.

I now move to a comparative system analysis, in examining the outcomes of the two systems respectively in the two countries.

8.3 Ethiopia: comparative system analysis

As discussed earlier, in Ethiopia, while the Patient based system was greatly successful, the DHIS2 was not so. I now analyze the reasons for these differences, as it is puzzling why the same context influenced two systems differently. Again, four conditions (identified in the literature) which are seen to be relevant to my analysis are now discussed:

- Team structure and organization

In Ethiopia, the difference in team structure and organization between the two software teams was significant. The Patient based system team was more balanced with the involvement of a medical doctor and a Masters student in public health, alongside the Informatics students. While in contrast, the DHIS2 team was comprised only of Informatics students.

- Technology introduction process

In Ethiopia, the technology introduction process started similarly for both cases through a "learning by doing" approach. In the first two months, there was no big difference between the two teams. But right after that, the Patient based system was brought and installed in one specific clinic, the DHIS2 system was only in the discussion phase and expectations of it being installed in Tigray and Benishangul-Gumuz regions. Later, the team sent the Patient based system to South Africa to receive constructive feedback to improve the system, setting the base for furthering the learning process. In the case of DHIS2 team, the learning process did not continue after I left Ethiopia, leaving the team to work alone. Long after that, the team received a short visit by the Oslo team;

however, it was too short and late to become effective, at least with respect to the Tigray and Benshangul MOUs.

- Integration between development and use

This aspect is greatly different between the two systems. The Patient based system team created a strong link between development and use right from the beginning.
The team members were very keen to go to the clinics to give training and support every week. Through the close communication with the end-users who actually use the system, they learnt how to make the system better. They could feel the difficulties of the users and therefore quickly give solutions.

In contrast, the DHIS2 team also had the same advantage to create this development and use link through the two MOUs signed between HISP and the two regions. However, this advantage was not implemented in reality.

- The support from the network

In the case of the Patient based system, the team received support from Vietnam and South Africa. In the case of DHIS2, the team received support from Vietnam and in a very limited manner from Oslo. For example, the Oslo team gave only one visit to Ethiopia for 10 days. The lack of support from the network made the DHIS2 team in Ethiopia become very constrained to engage with the complicated technologies used in DHIS2.

Four additional conditions also identified to be significant are now further discussed.

- The heaviness of the technology

Although there are many social aspects that were significant in shaping the outcomes of the systems, the heaviness of the technologies was also influential. While the Patient based system team had to learn only PHP and Mysql, and could produce the application themselves; the DHIS2 team had to learn many frameworks, to be able to start the development, such as Spring, Hibernate etc. Also, DHIS2 takes long time to build and run, and given the low computer capacity in Ethiopia, the development process was therefore more difficult than the Patient based system which was under local control and less Internet dependent. The aspect of local control is now further discussed.

- The control over the development process

From the empirical evidence, the control over the development process was identified as an important condition shaping the different outcomes. The Patient based system team had decided the design of the system and selected technologies they were capable of handling independently. They had the latest source code in their laptops and were always able to work with the latest version. Time was optimized to do the actual development work and provide field support instead of spending all their time on downloading and uploading the source code. Also, these time-consuming steps were impossible due to the Internet problems. Moreover, as the requirements were identified locally, no group discussions over Internet were needed thus helped to reduce delays and misunderstandings in communication. In other words, the Patient based system team had full control over the development process. In contrast, the DHIS2 were nearly completely dependent on the Internet and external control of the Oslo group.

The instability of DHIS2 also constituted to the problem. While it took the DHIS2 team one or two days to get the latest source code, the speed of change in the source code from the original was so fast, some times on a daily basis, that the local team did not have the infrastructure and capacity to deal with it.

- The geographical and cultural proximity between the development and use domain

The distance between the development team and the clinics in the two systems also constituted to the different outcomes. The Patient based system team could go to visit the clinics in half an hour while to the DHIS2 team a single visit could take days to organize (given the bureaucracy of the management), and was experienced. Students working only with scholarship could not afford to undertake this travel. And for the two regions of Tigray and Benishangul-Gumuz, it was a case of "out of sight and out of mind".

- The commitment of the development team

The Patient based system team was comprised of two very dedicated and committed developers who genuinely wanted to contribute to solve the HIV/AIDS problem in the country. In contrast, the DHIS2 team members lacked similar commitment and dedication, and during the course of the year, both the students dropped out from the studies. Their resolve to contribute was further dismissed.

The Manager of HIV/AIDS program in Addis Ababa was a champion for the Patient based system gaining the political support and encouraging the developers. No such champion existed for the DHIS2 team.

The comparative analysis between two systems in Ethiopia highlighted many different conditions across the two systems even though in the same country. A similar process is also employed for the case of Vietnam, which is discussed in next section.

8.4 Vietnam: comparative system analysis

Similar to the cross system analysis in Ethiopia, in this section, I provide the analysis between the two systems in Vietnam. As discussed earlier, DHIS2 was reasonably successful while the Patient based system not so. While the technological infrastructure and nature of the sector were relatively similar, other differences in the social context can be seen to contribute to the differences.

- Team structure and organization

While the DHIS2 system had a team of 3 developers and 1 facilitator, the Patient based system had only me. This shortage of resources significantly affected the development efforts.

- Technology introduction process

As presented in the empirical chapters, DHIS2 was introduced to Vietnam in 2005 while the Patient based system only in May 2006. The period was too short to create positive outcomes.

- Integration between development and use

While the use of DHIS2 in Vietnam was politically supported by a MOU signed between HISP and the Health Services, the Patient based system did not have that kind of support. Without political support, the link between development and use was not created, thus also impeding the development of the system.

- The support from the network

Global

DHIS2 received strong support from Oslo since the beginning as they are interested in making Vietnam as a strong node. The team members were trained by a Norwegian Masters student, who came during their last semester at the university. After that, every month, another Masters student would come to work with the team to replace the previous one. In the Patient based system, there was only myself with no Oslo involvement.

Local

Locally, the DHIS2 team had strong support from the local network. For example, the team had a facilitator and a health domain adviser, and also political support through the signed MOU. In contrast, the Patient based team did not receive any support from either the global nor local network.

- The complexity of the system

Compared to DHIS2, the Patient based is more complex in terms of business logics because it deals with the patient. Technically, the flexible patient based system was also more complicated than DHIS2 as it provided the functionalities for users to define forms and reports as per their needs etc. This functional and technological complexity also contributed to the difference in outcomes of the Patient based system compared to DHIS2.

So far, the four comparative cases have been analyzed. In the next section, I provide the discussion which helps to develop broader inferences from the study.

CHAPTER 9: DISCUSSIONS AND CONCLUSION

This chapter provides the discussion based on the analysis presented in the last chapter. At the outset, I summarize on a more consolidated and overall basis, three key categories under which the social conditions presented in the analysis could be collapsed to. I then outline how these conditions shape the development and use of OSS projects for health care, more broadly in developing countries. A summary table (Table 22) is presented and then discussed:

Categories	Description
Technological Infrastructure	- Internet dependence - Heaviness of technologies - Policies of access to infrastructure
Organization Arrangements	- Skill mix of the team - Technical capacity of the team - Dedication and commitment - Role of champion
Development Process	- Network support and local agency - Integration between development and use - Proximity of support

Table 22: The summary of social conditions

These 3 categories are now discussed.

- **Technological Infrastructure (TI)**

TI includes the hardware, software, Internet and bandwidth etc that are required to constitute the applications, the technologies used to build them, and the policies which enables development teams to have access or not to the infrastructure. Through my empirical work and analysis, 3 key facets of TI were identified as relevant in shaping processes of OSS development and use:

- ***Internet dependency:***

OSS technology, as expected, is heavily dependent on the Internet for downloading and uploading source codes, communicating with the development team to get support etc. However, in the context of developing countries within the health care sector, this availability can not taken for granted, and particular strategies need to be adopted for reducing this dependence by choosing "lighter" technologies, or those which are easy to learn and use. For example, the problems of DHIS2 in Ethiopia was avoided in the Patient based system because PHP could be learned relatively easily by the local team, and thus they did not need to depend on the global team for support, which reduced the Internet dependence. This local control through reduced Internet dependence also contributed to the speedy dealing of user requirements by the development team.

- ***"Heaviness" of technologies***

For OSS in developing countries, the choice of technologies plays an important role in affecting the outcomes. Many developing countries due to outdated university curriculum tend to have weak capacity and experience in dealing with the latest technologies that the OSS domain demands. The choice of technologies needs to be more balanced between the "superiority" of new technologies and the sensitivity to local needs. There is no "best technology" but it depends on how social groups interpret it from their own perspectives and apply it for their own needs (Bijker *et al.* 1987). So it is not obligatory for developing countries to select such heavy and complicated technologies because they are the "bars" to build OSS software if it is contrary to their local conditions and demands. For example, the Patient based system used technologies which could enable the local team without prior programming skills to start the development work within a month while similar efforts to master DHIS2 seems to take years and not months.

- ***Policies of access to infrastructure***

In the West, access to high speed Internet and other tools is taken for granted. However, in developing countries, this is not the case. In addition to the technological constraints discussed above there are also policy related issues. Support from the government can help to create awareness about OSS and encourage sectors like health care to take up OSS instead of proprietary applications. For example, DHIS2 in Vietnam was enabled through government policy that promotes OSS, while in Ethiopia that was not the case. Another example of the role of policy is the banning of SMS in Ethiopia, something which is taken for granted all over the world.

The notion of TI discussed above is quite different from earlier SCOT literature, where Internet has not played a prominent role. The specific maternal characteristic of the Internet such as networks and connectivity, create quite particular implications in the OSS domain. Also, the manner in which the visibility created by the Internet is seen as threatening by some governments, there may be policy implications.

I now move to the second category of the social conditions.

- **Organization Arrangements**

This term is needed in a broad sense to reflect various characteristics of the team including the mix, capacity and the nature of team member involved.

- ***Skill mix of the team***

In the health care sector, the balance between expertise in the public health and informatics domains is important to ensure the success of the project. Independent skills in either informatics or public health are not adequate for success. The involvement of domain experts (Medical Doctors, Public Health graduates etc) are required to provide domain knowledge to clarify requirements, rectify the design and engage in acceptance testing. The example of the Patient based system reinforces this argument that the appropriate skill mix contributes significantly to the development and implementation process. This was not the case in DHIS2 as arguably the skill mix was inappropriate. This finding reinforces the argument that application software like health within the OSS domain, need a very different level of skill mix as compared to traditional OSS projects where the developers tend to be users. Either additional domain experts need to be introduced in the development network, or mechanisms for crossover between the two groups need to be developed.

- ***Technical capacity***

Technical capacity is vital for dealing with technologies - OSS or others – and needs to be built through a proper learning process. Developing countries, due to historical dependence on proprietary software and poor Internet, tend to be lacking in appropriate human resources capable with OSS. Poor Internet can be also a barrier for learning. However, building technical capacity is not an impossible mission, as experienced in the Patient based case in Ethiopia. The learning was done by the process of doing actual work. The learning process was enabled through a cultivation approach, starting with some easy and simple technologies first, and then gradually increasing the levels of complication. Learning took place primarily in a co-located environment. This approach to technical capacity building is quite in contrast to the typical approaches in OSS projects where Internet is emphasized as the key medium. Developing country projects need to find different mechanisms for learning and capacity building.

- ***Dedication and commitment of the team***

Attracting and retaining OSS developers in developing countries is a challenge, as there is always competition from a more lucrative global private sector. While monetary motivation is indeed important for both the public and private sector, the public sector of health can not compete with the private in financial terms. So other forms of motivation are necessary to supplement monetary ones. As shown in the Ethiopia patient based case, the motivation for the 2 developers was intrinsic, an internal driving force. The two members of this team have been working very hard and dedicated nearly the last one year of their life completely for the software development and implementation. From their heart, they believed strongly that they do need to help humanity in the fight with HIV/AIDS. Their total dedication and commitment have even led to delays in their thesis submission. This dedication was totally reversed in the DHIS2 team, where both the members dropped out from the team, and one apparently left for America.

- ***"Role of the champion"***

The role of champion has been emphasized by various IS studies as critical for success. The champion is defined as the individual who has the vision, who keeps pushing when going gets tough, and who generates creative energy, and makes it all happens (Cook 1988). The case of the Patient based system in Ethiopia reinforces this argument. The champion in this case was the health manager of HIV/AIDS program in Addis Ababa, who gave the political support to pilot, continued to fight for the project existence, provided a strong source of feedback on the system and built enthusiasm. Without him, it can be emphatically said that there would have been no ART system in Ethiopia.

The discussion on organization arrangement is quite different from earlier literature on SCOT in the manner it emphasizes various facets. While Nhampossa and Sahay (2005) also drawing upon SCOT have identified conditions of technical skills etc, they have not looked upon issues of dedication and commitment - which we see as crucial.

- **Development process**

There facets of the development process identified relevant for OSS are network support and local agency, integration between development and use; and proximity of support. Each of them is now discussed.

- ***The network support and local agency***

OSS projects especially in developing countries need support from outside to build up capacity, develop and support the system. However, external support can be a double edged sword as it can both support and also inhibit local agency and capacity. Support from the network should enable the learning process, not just be orders and setting up of criteria to follow. For example, in respect to the patient based system, the Ethiopian team received support from South Africa to strengthen their capacity and the collaborative work was built upon what they had already done and were committed to. Local agency was respected by the support network. The support process was started by open communication to identify the needs for support, and mechanisms for it, not what the outsiders wanted to impose. In the DHIS2 case, all the decisions and choices were made in Oslo, and the other nodes were then expected to follow. The logic of local agency development was quite different in the two cases, contributing significantly, I argue, to the difference in outcomes.

- ***Integration between development and use***

The OSS literature typically discusses general domain software such as compliers, operating system etc in which the business logic is commonly understood by the developers who also are the users. In OSS for specific domains like health care, the functionalities of the system are not well defined, and the developers and users are two different groups. The involvement of the users will help to clarify the business logic and specify the requirements concretely. For example, patient based system is a term referring to various systems used in hospitals and clinics. In the context of Ethiopia, by closely integrating the development and use, the system was developed to fit the requirements of ART clinics. Later, the generic system (using Symfony) was also developed based on the defined processes of the ART system. The integration between development and use also contributed to the motivation of OSS developers as they felt their efforts were producing concrete benefits for humanity.

- ***Proximity of support***

While the link between development and use is important, proximity (geographical and cultural) becomes a precondition for strengthening that link. With geographical proximity, the development team can quickly go to the field, observe the problems and solve them. This proximity helps to reduce the cost for travel and time delays to give support. Cultural proximity such as similar language can help build trust and confidence amongst the local users towards the development team.

In summary, I have in this discussion presented more theoretically informed inferences that flow from my analysis. Based on this, I present my revised theoretical perspective.

My REVISED THEORETICAL PERSPECTIVE

As a starting point, based on the reading of the literature, I had in figure 2 presented as initial perspective. Based on my specific empirical analysis, I propose a revision of this perspective based on the specific characteristics of OSS, health care, and Vietnam/Ethiopia. This perspective is schematically summarized in the Figure 22 below:

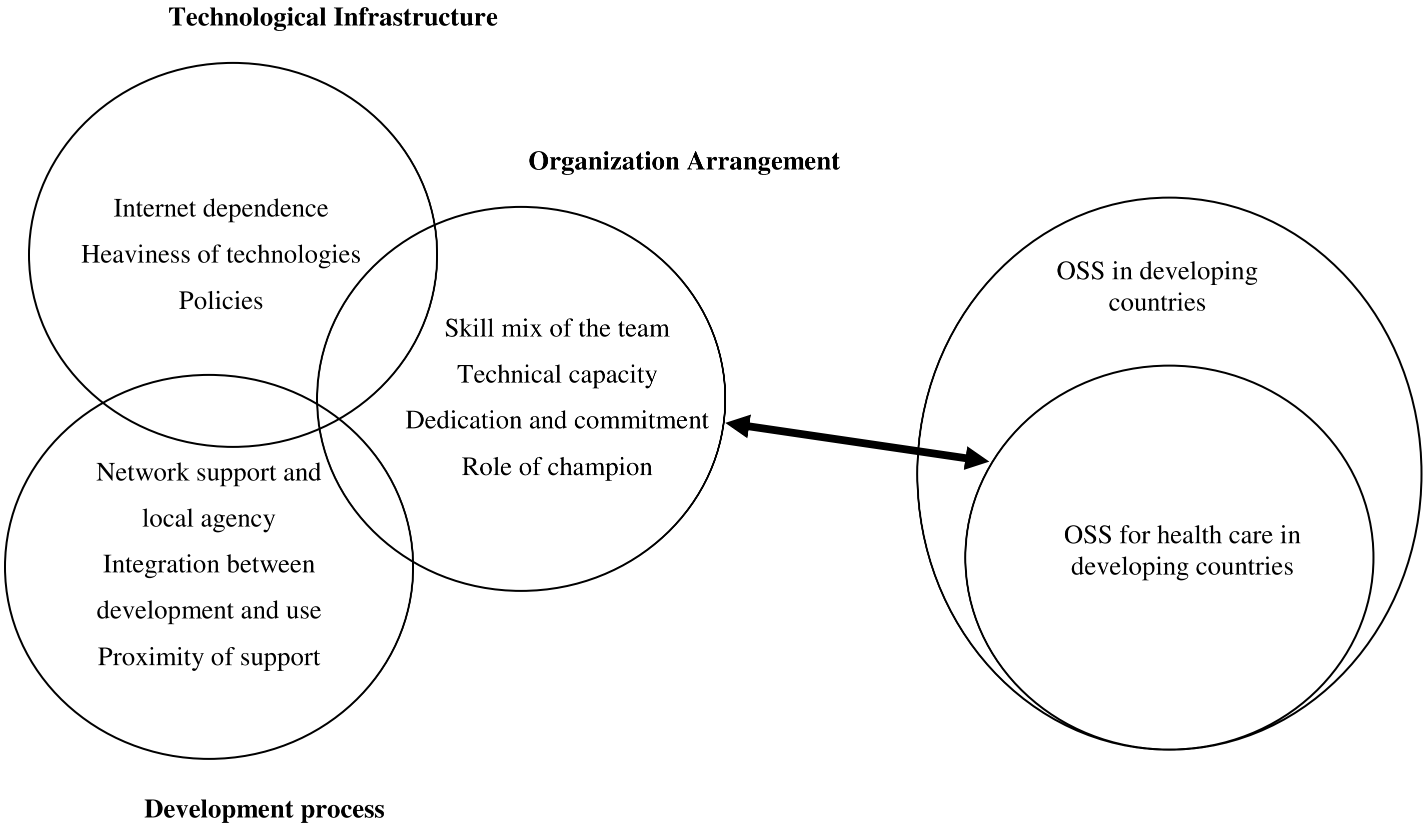


Figure 22: The framework of social conditions and OSS for health care in developing countries.

The Figure 22 summarizes the three sets of social-technical conditions that were seen to shape the trajectories of development and use of the two OSS projects for health care in the two developing countries studied. I argue that this conceptual framework can be useful to apply OSS projects more broadly and also developing countries more generally. While the specifics of how these conditions may mediate the relationships can vary, and the particularities of the countries will be different, the proposed framework can be considered as a starting point for future OSS in health care while developing country projects. Empirical analysis can further contribute to revise and expand this framework. .

- **Conclusions**

The thesis started with the twin aims of understanding the nature of the social context and how that shapes the outcomes of the development and use processes of OSS projects in developing countries, and secondly to develop implications of how global development of such projects can be managed effectively. In response to the first aim, my proposed revised framework (Figure 2) summarizes the social-technical influences that were identified to be relevant. The response to the second aim can be developed for from this framework.

For example, the question of internet dependence should be considered when planning for such projects, and projects originating from the North while making technological choices should not assume access to high speed internet is available like it is for them. Even if the technological infrastructure may be in place, the policy environment may not be conducive in establishing an enabling environment. Another important implication concerns the constitution of the team. While of course the appropriate skill make up needs to be considered carefully, we must also think about the issue of incentives for the developers to participate. Monetary incentives though important and necessary, might not be sufficient, We have to understand what kind of intrinsic motivations are possible, as it was seen that with this as a driving force, there is complete ownership and commitment, and learning barriers can be overcome. Network support is also seen as a crucial condition, and the need for cultivating both local and global networks are important. It is also emphasized that the North should play a more enabling and cultivating kind role in the development of the networks, and the South should be equal partners from day one- in the selection of the technologies to be used. Without this, there is the danger that their dependence on the North may be perpetual which are then subject to the usual constraints of internet, expertise, and communication problems.

I believe that my research makes useful contributions both theoretically and practically, and could be used as useful inputs to the broader HISP network as they engage with the challenging task of organizing and revising their global OSS development strategies. Like all research, mine too, suffers from limitations, such as the basis for comparison, at least for

DHIS2, may have been better conducted with India rather than the two countries selected, as they are in a more advanced stage of implementation. Also, given the different stages of maturity of the patient based systems, the comparison may be seen to be a little unbalanced.

Personally, this thesis has been a learning experience for me, and I believe that it has established a strong basis for me to conduct future research in this very interesting and challenging domain. I hope to further this learning process with future PhD studies. I will also try to take this learning experience into the practical management of the Vietnam HISP project, which currently needs urgent attention.

Ω Ω Ω

BIBLIOGRAPHY

Avison, D. and Myers M. (2002) "An Introduction to Qualitative Research in Information Systems: In Qualitative Research in Information Systems". D, Avison, and M.D. Myers (eds). London: 3-12

Avgerou, C. and Walsham, G. (eds) (2000) "Information Technology in Context", Ashgate, Aldershot, UK

AbouZahr C. and Boerma T. (2005), "Health information systems: the foundations of public health", WHO Bulletin (2005) Special theme on HIS, Vol 83, No. 8

Appadurai, A. 1997, "The Colonial Backdrop". Afterimage 24:4.

Bezroukov, N. (1999). "Open source software development as a special type of academic research" (critique of vulgar raymondism). First Monday.
http://www.firstmonday.dk/issues/issue4_10/bezroukov

Awala Equar (2003) Morbidity and Mortality in Respiratory Infections Registered in Health Facilities in Ethiopia. Master thesis, Umeå University, Sweden

Barley, S. (1986), "Technology as an occasion for structuring: evidence from observations of CT scanners and the social order or radiology departments", In Administrative Science Quarterly 31(1) 78-108.

Braa, J., Monteiro, E., Sahay, S., Staring, K. and Titlestad, O. (2006), "From asymmetric to symmetric models of offshoring: the case of application development for public health in developing countries", manuscript, (Subm).

Braa, J., E. Monteiro, and S. Sahay (2004) "Networks of Action", In MIS Quarterly 28(3): 1-26.

Braa, J. and C. Hedberg (2002) "The Struggle for District-based Health Information Systems in South Africa", In Information Society 18(2): 113-127.

Bijker, W. E., Hughes, T. P., & Pinch, T. J. (Eds.) (1987), "The Social Construction Of Technological Systems: New Directions", In The Sociology And History Of Technology. Cambridge, Ma: MIT Press.

Bokhari S. H. and Rehman R (1999): "Linux and the Developing World". IEEE Software 16(1): (1999)

Baskerville, R. and A. Wood-Harper (2002) "A Critical Perspective on Action Research as a Method for Information Systems Research". D, Avison, and M.D. Myers (eds). London: 129-145.

Bensbasat, I. et. al. (1987). "The Cse Research Strategy in Studies of Information Systems". MIS Quarterly, September.

Creswell, J.W. (2003). "Research design. Qualitative, quantitative and mixed methods approaches". Thousand Oaks, CA: Sage.

Cornford, T and S Smithson (1996) "Project Research in Information Systems", Macmillan, London

Câmara, G., and Fonseca, F. (2007), "Information Policies and Open Source Software in Developing Countries", Journal of the American Society for Information Science and Technology, Volume 58, Issue 1, Pages 121 - 132.

Carnall D (2000), "Medical software's free future", Bmj 2000; 321(7267):976.

Carnall D (2000), "Open Source Software in Healthcare". In: Informatics review; 2000.

Chilundo, B. (2004), "Integrating Information Systems of Diseases-Specific Health Programmes in Low Income Countries: The case studies of Mozambique", PhD, thesis, Faculty of Medicine, University of Oslo, Oslo.

Cook P G (1988), "The role of champions in successful computer applications", ACM SIGOIS Bulletin Volume 9 , Issue 1 (Jan., 1988) Pages: 1 - 13 Year of Publication: 1988

DHIS2 mailing list (2004-2007), http://www.hisp.info/confluence/display/DOC/Mailing+lists

Fitzgerald B. (2006), "The Transformation of Open source software", Forthcoming in MIS Quarterly, Vol. 30, No.3, 2006

Imsland, V. and Sahay, S. (2005) Negotiating Knowledge: The Case of a Russian – Norwegian Software Outsourcing Project, Forthcoming, Scandinavian Journal of Information Systems, 17.

James J. (2002), "Free software and the digital divide: opportunities and constraints for developing countries", DOI: 10.1177/016555150302900103 2003; 29; 25 Journal of Information Science

Giddens, A. (1984). "The constitution of society: Outline of the theory of structuration". Berkeley/Los Angeles, CA: University of California Press.

Hanseth O., Jacucci E., Grison M. and Aanestad M (2006). "Reflexive standardization. Side-effects and complexity in standard-making". MISQ Vol. 30, Special Issue on Stadnardization Aug. 2006, pp. 563-581.

Hanseth, O. and E. Monteiro, and M. Hatling (1996) "Developing Information Infrastructure the Tension between Standardization and Flexibility", In Science, Technology and Human Values, 21(4) : 407-426.

Heeks, R.B. (2003), "Designeality Gap Assessment", IDPM, University of Manchester, UK,

http://www.egov4dev.org/causegap.htm [accessed 27 Mar 2007]

Kshetri, N. (2004) 'Economics of Linux Adoption in Developing Countries' IEEE Software (January/February): 74-81.

Krishna, S., Sahay, S. and Walsham, G. (2004), "Managing cross-cultural issues in global software outsourcing", Communications of the ACM, 47, 4, 62-66.

Lagebo and Mekonnen (2005), "Challenges and Approaches to Scaling and Standardizing Health Information Infrastructure in Developing Countries: Case Studies from Ethiopia", Master Thesis, Department of Informatics, University of Oslo, 2005

McDonald CJ, Schadow G, Barnes M, Dexter P, Overhage JM, Mamlin B, et al. Open Source software in medical informatics--why, how and what. Int J Med Inform 2003;69(2-3):175-84.

Mosse, E. and S. Sahay (2003) "Counter Networks, Communication and Health Information Systems: A Case Study from Mozambique". In The IFIP TC8 & TC9/WG8.2+9.4 Working Conference on Information Systems Perspectives and Challenges in the Context of Globalization. M. Korpela, R. Montealegre and A. Poulymenakou (Eds), Athens, Greece: 35-51.

Myers, M. D. "Qualitative Research in Information Systems," MIS Quarterly (21:2), June 1997, pp. 241-242. MISQ Discovery, archival version, June 1997

Meystre S. and Muller H. (2005), "Open source software in the Biomedical Domain": http://www.sim.hcuge.ch/medgift/publications/SMI2005_OS.pdf

McNAMARA, K.S. (2003) "Information and Communication Technologies, Poverty and Development: Learning from Experience" [online] Last accessed on

7/23/2006 at URL: http://www.infodev.org/files/833_file_Learning_From_Experience.pdf

Ministry of Health (2002) Health Sector Development Program II: EFY 1995 – 1997, the Federal Democratic Republic of Ethiopia. Addis Ababa, Ethiopia. (Unpublished document).

Mockus (2000). "A case study of open source software development: the Apache server". Proceedings of the 22nd international conference on on Software engineering, Limerick Ireland, ACM.

Marx, L., Smith, M.R. (1994), "Introduction", in Smith, M.R., Marx, L. (Eds),Does Technology Drive History? The Dilemma of Technological Determinism, The MIT Press, Cambridge, MA, pp.ix-xv.

MURPHY T. (2004), "U.N. Forum Delegates Cite Technology Gap", Associated Press Online, June 18, 2004, Trade Observatory, Retrieved from http://www.tradeobservatory.org/headlines.cfm?refID=31781.

Nhampossa, J.L. and Sahay, S. (2004). Social construction of software customization: the case of Health Information Systems from Mozambique and India, Submitted to the IFIP 9.4 Conference to be held in Abuja, Nigeria, 2005.

Nicholson, B. and Sahay, S. (2004). "Embedded knowledge and offshore software development, Information and Organization", 14, 4, 329-365.

Orikowski, W. (1992) "The Duality of technology: rethinking the concept of technology in organisations", In Organization Science 3(3): 398-427.

Robey S. and Robey D. (1996), "Organizational context, social interpretation, and the implementation and consequences of geographic information systems", Accting.Mgmt, & Info.Tech., Vol.6, No.4, pp 255-282, 1996

Robertson R, (1992). "Globalization: Social Theory and Global Culture". London: Sage.

Stallman R, (2002). "Free software, free society": Selected essays of Richard M. Stallman. Boston: Free Software Foundation.

Redmiles, D., van der Hoek, A., Al-Ani, B., Quirk, S., Sarma, A., Silva Filho, R., de Souza, C., and Trainer, E., "Continuous Coordination: A New Paradigm to Support Globally Distributed Software Development Projects", Withschaftsinformatik, 49:2007 (to appear). Retrieved from http://www.ics.uci.edu/~etrainer/publications/cc-globalsoftdev.pdf

Sampsa Hyysalo (2006), "The Role of Learning-by-Using in the Design of Health Care Technologies: A Case Study", The Information Society, 22: 89–99, 2006

Sahay S. and Nicholson B. (2004), "Embedded knowledge and offshore software development", Information and Organization (14) 2004

Sahay S. and G. Walsham (2005) "Scaling of Health Information Systems in India: Challenges and Approaches". In BADA, A and O. ADELAKUN, (eds.) Enhancing Human Resource Development through ICT, Nigeria, Abuja, IFIP: 41-52.

Sahay, S. (2003) "Global software alliances: the challenge of 'standardization'", In Scandinavian Journal of Information Systems 15: 3-21.

Sahay, S. and Sarker, S. (2003). "Implications of Space and Time for Distributed Work: An Interpretive Study of US-Norwegian Systems Development Teams", European Journal of Information Systems, 13, 1, 3-20.

Tushman, M. and Anderson, P. (1986) "Technological Discontinuities and Organizational Environments", Administrative Science Quarterly, 31(3), 439-465.

Tigray Health Profile (2002) HMIS division, Planning and Programming Department, Tigray Health Bureau. Report for EFY 1994.

Tigray Health Profile (2003) HMIS division, Planning and Programming Department, Tigray Health Bureau, Report for EFY 1995.

Walsham, G. (2002) "Cross cultural software production and use: A structurational analysis" MIS Quarterly Vol. 26, No.4, 2002

Walsham, G. and S. Sahay (1996) "GIS for district-level administration in India", In International Journal of Geographical Information Systems 10: 385-404.

Walsham, G. (1995) "Interpretive case studies in IS research, nature and method", In European Journal of Information Systems 4: 74-81.

Walsham, G. (2001), Making a World of Difference - IT in a Global Context, John Wiley & Sons Ltd, Chichester, U.K.

Walsham, G. (1993) "Interpreting Information Systems in Organizations", John Wiley and Sons, Chichester, UK.

Weber S. (2001), "Open Source Software in Developing Economies", www.ssrc.org/programs/itic/publications/ITST_materials/webernote2.pdf

Weber, S. (2004), "The Success of Open Source". Cambridge, MASS: Harvard University Press.

Weerawarana, S., & Weeratunga, J. (2004). Open Source in Developing Countries (No. SIDA3460en). Stockholm, Sweden: Sida 2004 - Department for Infrastructure and Economic Cooperation.

Wright G and Myurray P. (2002) "Open Source: Global Issues" In: MIST 2002; 2002.

Yackel, T.R. (2001), "How the open source development can improve medical software", Medinfo 2001, IOS Press, Amsterdam 2001.

WHO Bulletin (2005) "Special theme on HIS", Vol 83, No. 8

World Health Organization (1999) An Assessment of Reproductive Health Needs in Ethiopia. [Online] Available from http://www.who.int/reproductivehealth/publications/HRP_ITT_99_1/HRP_ITT_99_1.abstract.en.html. [Accessed 13 March 2004].

Winner L (1993), "Social Constructivism: Opening the Black Box and Finding It Empty," Science as Culture, Vol. 3, part 3, no. 16, pp. 427-452.

World Fact Book (2004), https://www.cia.gov/library/publications/the-world-factbook/print/et.html

World Fact Book (2007), https://www.cia.gov/library/publications/the-world-factbook/geos/vm.html

Yin, R. K. (2002) "Case Study Research, Design and Methods", 3rd ed. Newbury Park, Sage Publications, 2002.

Yin, R. K (1981). "The Case Study as a Serious Research Strategy," Knowledge (3), 1981, pp. 97-114.