UNIVERSITY OF OSLO
Department of informatics

Exploring the use and effect of Intermediary Objects in a global Open source organisation

Master thesis
60 credits

Stian Aleksander Strandli

2. may 2008

Acknowledgement

I would like to thank both my supervisors Gianluca and Knut, for their valuable guidance during these two last years. I would also like to thank the whole organisation for the opportunity to let me travel to India for over three months. My most sincerely appreciation goes out to the whole India team who made the trip to India a really enjoyable experience. A thank you also goes out to everyone who has given me feedback on my thesis. Last but not least I would like to thank all my family and friends for the support they have given during the last two years.

Table of contents

1 Content

1.2 Figures

1.3 Tables

1.4 Code examples

1.5 E-mails

1.6 Field Notes

2 Introduction

In this chapter I will present an introduction to what the topic of my thesis is. It will include a short description of what I planned to do, what I did and why it was done. The chapter also contains my motivation for doing this master thesis and a high level description of the content and order in the thesis.

2.1 Background

In a lot of developing countries it is difficult to deliver good health care to the entire population, especially in the rural areas. In many countries, including India, they have established local clinics where a nurse or another trained health-person give some care to the local population. Many countries have also established different programmes to better the health of their people; this can include increasing the vaccination rate, decreasing the number of miscarriages or detecting rapid spread of diseases. For reaching these goals they collect information at the local clinics and use this data, to be able to increase the health of the population. These programmes are often vertically organised and share very little information with other programmes. I have been involved in an organisation trying to create an information system to support this mission.

This organisation is always trying to expand their endeavours around the world or inside the countries they are already present. This organisation is based on the idea of utilising the data on every level: local, district, state, and national. The organisation is not tied to any of the vertical programmes. They try to develop and support a free, flexible, and user-friendly information system for collecting, analysing and presenting the data. One of their strategies for expansion is working together with or joining other programmes or organisations in their effort for better public health care.

In India there are many different vertical programmes, many of them with their own organisations, software and procedures. The organisation I was involved in has been given the chance to implement different pilot projects in different states in India where they supply the software. But with sudden changes in local government and unstable conditions, it is difficult to get approved for a state wide implementation.

My involvement in this organisation was mostly to develop integration between my organisation’s software and other Indian software. The most important reasons for this integration are to decrease the number of software used, providing the government with a flexible and free tool, and trying to help different health programmes to be able to cooperate more. The integration of the software proved to be a somewhat difficult task for various reasons, which will be presented later in this thesis.

My research was conducted with qualitative methods and an action research approach where I stayed in India for almost four months participating in the development, training, and expansion of the information system. I will try to see the development process through the different objects used in development of software and see what effect they had on the development and the final product.

The theory I have used to analyse my data with is the concept of Intermediary Objects. These objects can be used to read the design process. Artefacts that may be considered objects are design documents, requirements, prototypes etc.

Authors Motivation

I had decided that I wanted to work as a system developer after my education, and I were determined to apply this into my master thesis in some way. I was presented the possibility of doing a masters degree for this organisation in an introduction class in research methods, which was held by one of the founders of the organisation. The project was focused around Java programming and gave the possibility of travelling, and doing fieldwork in another part of the world. Working in an open source project would give valuable experience in both development skills and working in a team. It was also a chance for me to help develop a piece of software that is used in the real world. I contacted some of the Norwegian coordinators and I decided to join their efforts in improving the health of the world.

Structure of the thesis

The thesis consists of different parts and it begins with this introduction of the thesis. Following the introduction is the chapter where I present my research questions which include questions around Intermediary Objects and software development.

The next chapter is a background chapter describing health information systems, the organisation I have been involved in, their software, and the software which I was assigned to integrate. In the next chapter I present the theories I find relevant and will use in my discussion later. This includes theory about some aspects of software development, background information about Open source projects and how they are managed, and the theory around the concept of Intermediary Objects.

The chapter describing my methods for undertaking this thesis is the next part. I will describe qualitative research and action research and how I conducted my research and analysis of my research data. This is followed by my presentation of my empirical findings which includes the work on the integration between two very different systems, creation of a new module for my organisation’s system and the training I did with Indian developers.

After I have presented my empirical findings, I will try to discuss my findings with the theory presented in this thesis and present a conclusion of what I have found. The last part of the thesis consists of my used abbreviations, references and appendices.

In this part I have given a small introduction to the thesis. I will next present my research questions.

3 Research question

In this chapter I will present my research questions and an explanation of what they intend to answer.

3.1 Intermediary Objects

- *Which Intermediary Objects were utilised and what effect did they have on my development?*

In my project I have used many different objects for many different purposes. Some of them are more important than other. A part of the objects had major effect while other did not affect the development process at all. I want to explore what caused my project to end like it did and what objects that did not have any effect. The concept of Intermediary Objects has not been widely researched, and I believe it can be an interesting way of analysing and seeing a development process through it.

- *How does my Intermediary Objects relate to the Intermediary Object concepts created by Papadimitriou and Pellegrin?*

There has not been much research on the concept of Intermediary Objects yet. Papadimitriou and Pellegrin are only one who have defined a list of different types of Intermediary objects, which they created based on their research. I want to see how my objects differ and match their findings. The goal is to see if I find the same type of objects and if they are found in the same period of the project as they found them.

- *What measures does Health Information Systems Programme take to support distributed development of their software?*

There is a lot of theory about how to conduct distributed development which is what the organisation in my case is performing. Based on my knowledge of travelling and participating in the project I will discuss if they follow the guidelines for doing development across the globe.

Here in this chapter I have presented and explained my research questions. The next part will give some background information that will introduce the reader to the field of health information systems.

4 Background

In this chapter I will present the history of health information systems which is the type of tool I have been developing. I will also present the project I have been involved in with some history, vision and structure. Their software called the Rural Health Information System (RHIS) will be presented with some history. I will present both the old software, which is still used in many countries and the new version which I have helped develop. I will also present the Disease Surveillance Integration Project (DSIP) which is the health programme and software I tried to integrate into our software. All this information is presented to help you, the reader, understand my case and thesis in a better way.

4.1 Health information systems

In my introduction I mentioned the need for collecting data about the health of the community; this is done with a Health Information System (HIS). Sæbø & Titlestad (2004, p. 1) defines a HIS as, "*a set of tools and procedures that a health program uses to collect, process, transmit, and use data for monitoring, evaluation, and control*" (taken from the Equity Project 2000).

A HIS is a tool that captures and uses data at a local level. It can greatly help make a suitable healthcare policy at all levels within the type of decentralized system suggested by the WHO. They (ibid) explain how health services should be conducted: "*The basic tenets are that health services should be offered and managed from small demographic and geographic areas to best achieve good communication with higher and lower levels, be close enough to communities to understand and act upon their problems, and be able to handle decentralization of resources and decision making*".

According to Edwards & Lippeveld (2004, p.1) many routine HIS in third world countries are not capable of providing useful information for planning and management based on the data collected. A problem is that most of the systems are centrally planned and it collects a multitude of information, but it is not being analysed and used properly. There is also a problem with lack of involvement from the people who actually collect data with the ones who manages the programmes. The transmission of data upwards in the system is to complicated and slow and by the time the report have went from bottom to the top and back, the information is old (ibid).

According to Braa et al (2004) there has been a growing recognition by international agencies (notably from the World Health Organisation (WHO)), government authorities and researchers from different domains including information systems, development theory, and public health, that improved HISs can significantly contribute to help address health service delivery problems. They give an example to describe how this can help by counting the number of infants in one area and the number of infants who have been inoculated to get the immunization coverage. Then they health authorities can compare different regions and do an effort in helping the areas with poorest coverage. Programme of Health Information Systems (PHIS) is one organisation which develops such a system and a vision for improving the primary health care. By making the data available and usable at the bottom levels they can see development over time, compare data with nearby clinics and use the data to give better care to their patients.

So now that I have given a brief introduction to what the field of HIS is about, I will describe the organisation I have been involved in.

4.2 Programme of Health Information Systems

The organisation I have been involved in is called Programme of Health Information Systems (PHIS). To get a better understanding of my case I will present some history and background information about the organisation.

History

PHIS is a non-profit, international organisation that is mostly lead by people working in the public health sector and academia. Only a small part of the leaders are employed fulltime. The rest of the PHIS team contains a good mixture of government institutions, researchers and students from different institutions from many different countries. A selection of these countries are; Norway, India, South-Africa, Mozambique and Tanzania (Braa et al, 2004).

PHIS's vision is to "*to support the development of an excellent and sustainable health information system that enables all health care workers to use their own information to improve the coverage and quality of health services within our communities*" (Organisation homepage, 2007-04-27). The principles they strive to work by is among other; Encourage empowerment, democratization & transformation in its activities, contribute to an action-led district health information system and shift power from IT managers to health managers and workers. To be able to shift power to the health workers they have to create an information culture at the bottom (ibid). By knowing where the biggest difficulties and needs are, they can administrate their resources, schedules and workforce better.

One problem that arises is that high quality technical solutions are created, but they do not match the local user's needs or interests. This is why it is important in PHIS have some of the power shifted from the developers to the organization leaders and the health worker who have more domain knowledge about what the health workers need, and they have more focus on what is needed, instead of what is possible from a technical point of view. Because of this most of the Norwegian core developers usually have at least one trip to one of the southern countries that utilises PHIS's software (Braa et al, 2004).

Another important aspect of creating a sustainable health information system is to develop and carry out relevant and extensive training sessions. In PHIS's case they have to train workers in the health sector and their managers. These principles lead to an action research approach, which means that researchers and the workers have to cooperate. One of the purposes is for the researcher to improve the conditions for the worker. Another goal is to clarify the workers needs and thoughts by a participatory approach to create a more correct overview of the situation. When researchers participate they can obtain insights that cannot be understood by studying it "*from a distance*". (Starring, 2006) This often demands collaboration between different sectors of the public health administration in a country and is one of the problems with this approach. In countries like South-Africa, India and other former colonies they often have a strongly centralized hierarchical structure which has lead to a strong vertical system. This does not correspond with PHIS's principles and line of action. This bottom-up approach can lead to a more decentralized political structure and change the balance of power in these types of countries. This is often met with resistance by people that are afraid to lose their power (Braa et al, 2004).

Structure

In this section I will discuss PHIS's structure and how it unfolds in the countries it is involved in. PHIS can be seen as a relatively loose network structure forming at the first horizontal level in all the different countries they are involved in. In a vertical perspective the second level consists of all the institutions in one country. Braa et al (2004) describes three different structural instruments in PHIS that are important in the spread of knowledge in the network. The first is the software developed by the South African development team at Western Cape University (UWC) which is spread throughout the network (RHIS version 1.X). Instrument number two is the Ph. D programme run at the University of Oslo (UiO). The last one is the two integrated master courses in public health and informatics at University Eduardo Mondlane (UEM) in Mozambique. These programs are held in collaboration with the UiO and are funded by Norwegian Development Agency (NORAD). This master course is open for students from all the nations included in PHIS (ibid).

Implementation funding in PHIS usually comes from donor organizations and local health authorities. Some of these organizations include; NUFU, USAID and Dutch Aid. Funding for research activities in different countries, scholarships for doctoral and master's students and funding for inter country linkage building comes from different sources, although it is primarily supported by the Norwegian authorities (ibid).

In addition to the structural layers mentioned above there is also the political structure in each country. A change in government will often lead to changes in political decisions. The government support often involves economical support and this is lost when the contract is cancelled. PHIS has experienced this both in Cuba and India. In Cuba in mid 2002 there was a political change and the health minister that supported PHIS lost his position (ibid). This meant no more support for PHIS and the existence in Cuba stopped. The same also happened with PHIS in India in one of their first implementations there.

This chapter have given a brief introduction to PHIS network and I will now present the software they develop.

4.3 Rural Health Information System

PHIS is currently developing to different HISs in parallel which supports the collection and usage of health data. They are two different versions of the Rural Health Information System (RHIS) where version 2.0 of this software is the one I have been involved in development of. I will in this chapter give a description of the history of both these systems.

History

RHIS originates from South Africa. In accordance to South Africa's Reconstruction and Development Program from 1994 they created teams to develop a plan for reconstructing the health sector in all provinces. PHIS was then initiated through this initiative by a committee in the province of Western Cape (Braa et al, 2004).

Their main focus was to develop a standardised set of Essential DataSets (EDS) and to develop a health information system to support the collecting of this data. The EDS was going to represent a minimum set of data elements which would be used to calculate indicators. Indicators in this sense are a value that represents i.e. the immunization rate of on disease or the rate of successful tuberculoses treatments. They are an important part of PHIS ideology and are much more valuable than just pure data. Indicators can show how medicines and personnel can be distributed more efficiently among different districts or clinics (ibid).

The first EDS were implemented in a city in Western Cape, and later in the whole province. This was at the time they started making a RHIS prototype. This prototype improved the success of the EDS. The standard that was created in Western Cape was later adopted by the Eastern Cape where they were having trouble creating their own database application. After this, PHIS was an important part of the national standardization process (ibid).

In the year 2000 the first national EDS standard was created. It contained 45 data elements where 40 % was different than the set created in Western Cape. The solution they created made it possible for every province to include more elements as long as it contained all the elements from the standard set. Most provinces had between 600 and 800 elements they collected. In the same period as this standardization process was going on they also started developing the first stable version of the RHIS application. A professional development team was put together to work on the RHIS. After a few more prototypes the first version was implemented in Western Cape. When the different health personnel started utilising the software the developers got continuous suggestions for improvement. The development team worked on site together with the users and gave out new versions weekly or even daily in some cases. Eventually as the use of the software spread they had to release bigger and more systematically versions where stability was the main focus. The new versions were tested by advanced users and in some chosen districts before they were released. Some courses have been created at the University of Western Cape where they learn to use the RHIS software and how to utilize the data they collect. These courses are part of the master study in public health programmes (ibid).

Up till now this version of the application has been introduced in multiple different African and Asiatic countries; Mozambique, Tanzania, Ethiopia and India. The RHIS software has always been given free of charge and the source code has been freely available to everyone. This has lead to the fact that some countries have adapted the software to fit their needs. So far the application has been based on Microsoft's Access platform and the latest version is 1.4. Seeing that multiple nations have joined the development and use of the software there has emerged a wish that the software would be platform and database independent. This was not possible with the tools used in RHIS 1.4 so it was decided to create a new version called RHIS 2 (ibid). This was supposed to be based on the same functionality that 1.4 had. Some other reasons for a change includes; users where having performance issues with the Access database and big amounts of data, the software was the result of a prototype approach and this created a very complicated data mode, and it was developed with technology and a language that was getting outdated (Nordal, 2006).

This was a short introduction of the first version of their software. I will next present the second version.

Rural Health Information System version 2.0

This section includes the development of RHIS 2.0 from the start in Oslo. This is the software I have helped develop during my time involved with PHIS. The background for the software and a description of it will be presented in this chapter.

When they decided to make a new version they also wanted to make the application have a web interface. It was also important that it could be run where they have no Internet connection. This version was supposed to be much better split into layers and have a modular architecture. The reason for this, is that it will make it easier to maintain a global development environment, where tasks are developed in different parts of the world. With a very modular architecture it is easy to also develop a desktop module with much of the same functionality without having to develop everything from scratch. This was their primary goal. Another important point was that the system should run on both open source and proprietary platforms (Nordal, 2006). It was the Oslo node that got the responsibility of developing this version (Starring, 2006).

The first time was spent searching for different platforms, standards and frameworks to be used. These had to match the demand for web interface and make it easy to develop continuously in different countries. One platform that has gained momentum the years before the decision was made and still has is the LAMP stack. This involves four different free and open source applications. This includes Linux for the operative system, Apache as the web server, MySQL as the database management system and PHP (, PERL or Python) as the programming language. This platform has a pretty gentle learning curve for developers, but it was perceived as "*toys*" from the Norwegian master students and was not well received by Indian developers either (Starring, 2006). One decision they had to take when choosing which platform and frameworks to use was what maturity the frameworks should have. It was a big decision to develop RHIS 2.0 and they wanted do it right from the start and make it a sustainable effort. On the one side they could choose mature and stable technologies that are not longer under development, this could be seen as safe choice for users and also developers. Then there was also the research perspective, which will not find it interesting with frameworks that are finished (Nordal, 2006).

In the end they decided to go with Java, which has proved itself as an established enterprise solution backed by big companies. The platform chosen were also found to be growing in popularity in Norwegian consultancies which made it more desirable for students to get involved in the PHIS project (Starring, 2006).

When RHIS 2.0 was developed, the Oslo node tried to collaborate with both India and Vietnam in the development process. In Vietnam there was first erected a collaboration with an outsourcing company in Saigon. Here six Vietnamese students worked with RHIS 2.0 in nine months. Students from Norway assisted them and held lectures about the frameworks for the entire company. Most of the students were not able to create anything useful. This collaboration was terminated in July 2005 and none of the students continued their relationship with PHIS (Nordal, 2006). A new collaboration was started with a local university where the three best students were hired to develop RHIS 2.0 fulltime with assistance from Norwegian master students. Here the Norwegian students also held seminars and lectures for the university (Starring, 2006).

In India there have been three different phases in regard to development. The two first was based on persons who had earlier been working with, but not developed, on the older 1.3 version of RHIS, was going to develop on version 2. All of these employees had a degree in computer science but little programming experience. No usable code came from any of the two first tries in India. The third attempt happened late in 2005 where two developers re-implemented most of the basic functions from RHIS 1.3. They developed this using only Java Server Pages (JSP) files and utilizing none of the frameworks endorsed by Oslo and there were no coordination between India and Oslo or Vietnam. It was still considered a small success and it lead to the hiring of these two developers (Starring, 2006).

Also in 2005, a Norwegian core developer and a coordinator were in India for a month to train some Indian developers. They already had one and they hired one more and they had two Indian master students. The two master students picked up on the technology but the new developer did not and left after a while so when the students were done after 3 months there was only the original developer left. The attempt to create a sustainable development team in India failed this time (Nordal, 2006).

The attempt to make a RHIS light failed in 2005 and PHIS India needed a system they could deploy in Kerala. An Indian working in a Norwegian company developed a quick solution in PERL in 10 days which had the very basic functionality of RHIS 1.3. First one Indian developer was hired to port this functionality to JSP instead. He did not produce anything in a month and they hired two new developers instead. This development was initiated in November 2005 (Nordal, 2006).

PHIS India needed a working solution by the end of January 2006. The Indian team wanted to continue develop on their JSP solution, but the rest of the PHIS team wanted them to wait for RHIS 2 which hopefully would be done in February. The Indian team agreed, but they still continued to develop their own solution on the side. This was seen as understandable because the future of PHIS in India depended on what they could deliver in the start of 2006. The developers felt they could not contribute anything on the RHIS 2 project because of their competence. RHIS was ready in February 2006 and was released to the pilot districts in Kerala (ibid).

Concepts from Rural Health Information System

In RHIS there are a number of concepts and definitions which may not be clearly understood without some knowledge of what they mean. Here I will try to explain some of the most important concepts that are used in RHIS 2.0. Table 4.1 is loosely based on Absolute Basics (2006) and my own knowledge of RHIS 2.0.

Concept name	**Description**
DataElement	DataElement represent one variable that can be collected and entered into the system. This may be the number of children vaccinated for measles or the number of people belonging to one geographical area.
Period	A Period represents a period in time from one date to another. The length of the Period is defined by its type.
PeriodType	PeriodType describes what range a Period has. Examples are: weekly, monthly, and yearly
DataSet	DataSet is a collection of DataElements that are being entered in the same

	form. Usually based on a paper form.
OrganisationUnit	This is a representation of one Public Health centre (PHC), Community Health Centre (CHC), Sub centre etc. that reports data into the system.
DataValue	One DataValue represents a value for one DataElement, reported by one OrganisationUnit for one Period.
Indicator	Indicator is based on an expression where you find a relation between two or more DataElements and other numbers. An example is the number of vaccinated children less than 5 years divided by the number of children under five years, to find the vaccination coverage for children.

Table 4.1 – description of concepts in RHIS 2

Normally a health programme have a defined a set of data they want to collect. Every individual piece of data they have defined is then mapped to a DataElement. All the DataElement who is reported at the same time is then mapped to one DataSet. The DataSet is usually formed based on a paper report used earlier. One example of this from India is Form 7 which is a big DataSet containing information about Antenatal care, which means care for women that are pregnant. A Period defines one period of time from which the data is collected. When putting a value together with DataElement, Period and Organisation unit you get one DataValue. Indicators are used to get valuable information from the raw DataValues. In essence every DataElement should be used in an Indicator to be worth collecting, but this is not clearly followed.

Rural Health Information System 2.0 releases

Since the release of the first milestone in 2006 there have been many milestone releases, but not (yet) any final 2.0 release. I was involved in the project from Milestone 6 and until they release the first beta version in February 2008. In some of the milestones it was published a “service release” because of a significant problem in the software. In Table 4.2 you can see the different releases and when they were released.

Version	Month and year
2.0-M 1	February 2006
2.0-M 2	March 2006
2.0-M 3	April 2006
2.0-M 4	May 2006
2.0-M 5	June 2006
2.0-M 6	December 2006
2.0-M 7	April 2007
2.0-M 7.1	April 2007
2.0-M 8	October 2007
2.0-M 9	December 2007
2.0-M 9.1	December 2007
2.0 Beta	6 February 2008

Table 4.2 – RHIS 2 releases

Technologies involved in RHIS 2.0

Java

Java is an object oriented programming language created by Sun Microsystems. It was released in 1995 as a part of Sun's Java Platform. The language inherits much of its syntax from the programming languages C and C++, but Java has a simpler object model and less low-level facilities. Java code is normally compiled into byte code that can be executed by a Java virtual machine (JVM) on any computer architecture. Java can be run on Microsoft (MS) Windows, Linux, Solaris and many more with the help of third-party vendors. Java has been criticized about its performance, how the graphical user interface of Swing looks, not being a real object oriented language because of the primitive types and more (Java, 2008).

When the Java source code files are compiled into class files they can be run by a JVM. It is customary to put files that belong to one program into one archive. This can be either a Java Archive (JAR) for running applications or packaging a set of functionality together or a Web Archive (WAR) that contains Java files for running on a web server.

Java frameworks

RHIS 2.0 utilises a great number of frameworks to make development on the system easier. In this chapter I will present the most used Java frameworks. Most of the frameworks were chosen because they minimize the coding needed for a specific purpose and are popular in the Norwegian IT industry. For more information on the rationale behind the choice of these frameworks see Nordal (2006).

Spring is an open source layered Java application framework. Its main purposes are to make Java EE development easier and help the developer use interfaces and develop in an object oriented fashion. The framework offers a complete lightweight container which provides centralised, automated configuration and wiring of Java objects. It also contains transaction management, integration with Object Relational Mapping (ORM) frameworks, Aspect Oriented Programming (AOP) functionality and a Model View Controller (MVC) web application framework. In RHIS 2.0 Spring is mostly used as a container helping with dependency injection and inversion of control (Spring, 2008).

WebWork is a Java framework created to support the building of web applications. It is developed with code simplicity and developer productivity in mind. The framework helps to abstract some of the coding necessary when using Servlets and Java Server Pages. It also lets the developer separate out functionality into Interceptors that can run before and after the page is run. According to WebWork (2008) it is providing "*robust support for building reusable UI templates, such as form controls, UI themes, internationalization, dynamic form parameter mapping to JavaBeans, robust client and server side validation, and much more*". WebWork is also an open source framework and it uses a modified version of the Apache License. The framework allows for different types of templates to be used: Freemarker, Velocity and JSP among others. RHIS 2.0 uses the Velocity framework for its user interface (WebWork, 2008).

Hibernate is an object/relational persistence and query service. The framework allows the programmer to develop persistent classes following object oriented idiom, which includes association, inheritance, polymorphism, composition and collections. These objects are used to store and fetch data from the database instead of writing vendor specific SQL. When using Hibernate you can create queries based on its own query language HQL, native SQL, or with an object-oriented

Criteria and Example API. (Hibernate1, 2008) Hibernate lets you write Java code that is usable on most SQL database servers. The framework is also open source and uses the Lesser GNU Public License (LGPL) (Hibernate2, 2008).

Java software

For the development of RHIS 2.0 there is a need for different software to support the development. Some of these are written in Java. I will present a short description of these programs.

Maven started as an attempt to simplify the build process of a system Apache was building. They wanted a standard way of building multiple projects, having a clear definition of the project, an easy way to publish the project information and a way to share JAR files across multiple projects. This resulted in Maven, which is a tool that can be used to build and manage Java projects. The project have these goals: making the building process easy, providing a uniform build system, provide quality project information and provide guidelines for best practice development (Maven, 2008). In the RHIS 2.0 project Maven is used to manage all the different modules in an easy way, it provides a user-friendly way of running unit tests with reports and to support the whole building process.

The preferred Integrated Development Environment (IDE) in the RHIS 2.0 project is Eclipse. All the guidelines and documentation is based on the usage of this open source IDE. The focus of the Eclipse project is building an open development platform comprised of extensible frameworks, tools and runtimes for building, deploying and managing software across the lifecycle (Eclipse1, 2008). The project is open source and its Eclipse Public License is approved by the Open Source Initiative (OSI) as an open source standard license (Eclipse2, 2008).

Jetty is Java implemented standard based web server. It was first created in 1995 and has benefited from a huge user community and a stable core of lead developers. It is licensed with the Apache 2.0 license and is open source. One of the main principles of Jetty is simplicity. It has an easy and user friendly XML based configuration and is created to work out-of-the-box (Jetty, 2008). In the RHIS Jetty is both used as a plug-in for Maven to run instances of RHIS 2.0 while developing and it is also used in production environment around the world as the web server for RHIS 2.0 along with Apache Tomcat.

Now that I have introduced the organisation in general and the software they produce, I will give some background information about what PHIS has done in India.

4.4 Programme of Health Information Systems in India

The Programme of Health Information Systems has been involved in India for many years. Since my involvement in PHIS was related to India and I also travelled there I think it is necessary to present a brief background chapter about their history in India.

History

In this section I will discuss PHIS's activities in India from the first time the organisation entered Indian until present time. India is the world's second most populated and the seventh biggest country in geography with its 1.1 billion inhabitants and 3.2 square kilometres (Wikipedia India). Each PHC caters to around 40.000 to 60.000 people and every district is inhabited of around 2 to 4 million people. The state's goal is to use IT to reduce the poverty in the country. In India as like many other countries the political system is extremely centralised, hierarchical and bureaucratic (Braa et al, 2004).

PHIS first entered India in 1999, when they first made contact with the government in Andhra Pradesh which is one of 28 states (Wikipedia India) in India. In 2000 they conducted a situation analysis and held a presentation for the officials in Andhra Pradesh. In 2001 they started developing the data sets needed for Andhra Pradesh and in 2002 they were chosen by the government. At this point they were using version 1.3 of the RHIS application (Indian Coordinator). They started RHIS prototyping and hired people for training. By 2003 they had started to integrate the system with FHMIS, which is a patient record system. Funding has been provided by the authorities in Andhra Pradesh (Braa et al, 2004).

After Andhra Pradesh and an implementation in the state of Kerala they got a deal for a pilot project in the state of Gujarat. They were provided some districts for a pilot test of the software. They had a big operation and had training in Ahmedabad, Ghandinagar and Surendranagar. From my own experience the doctors in the training seemed to be intrigued by the idea of PHIS. The main purpose of these training sessions was to focus on the basics of the RHIS system. Another important aspect was to focus on data quality. The slogan used was "*garbage in, garbage out*". If you enter fake data or not accurate data you will not get good information out of the system. One way to get better data was to integrate validations in to the system. Some of the time in the training sessions was used to let the participants define validation rules based on what they see in the field. Not everyone fully understood the concept and suggested what their (state i.e.) target was. There were also big differences on what the value should be. One of the reasons for this is the big difference of patients and problems in a big city clinic and one in a rural area.

In the time PHIS has been involved in Gujarat there have been collected over 1.7 million data values from over 50 different clinics (data from the Gujarat database). In March of 2007 there was a bid out for the possibility of providing the HIS for the ministry of Gujarat. PHIS India and one other company made propositions. PHIS's proposal was much higher than the other companies, but the other proposal was just for the software itself. PHIS gives the software for free and just charges for training and support. There was a committee appointed by one person in the health ministry and they concluded that the other proposal was best and PHIS lost their contract with Gujarat (India coordinator 1 & 2). This means that most of the people in PHIS Gujarat were fired and a lot of training, developing and work were to some degree wasted.

In 2007 they have been given two new places to show what they can do. These places are Bhopal where they have only 3 CHCs using the system and Chattisgarh where they were implementing during my trip to India. In Chattisgarh they had not started to use the system yet. They were still in the process of creating the DataElements and DataSets when I was in India.

National Rural Health Mission (NRHM)

In India as in many developing countries they have a lot of different vertical programmes. They are often funded from many different sources and have no incentive to cooperate in between them. The goal of the NRHM is to improve the availability of and access to quality health care by people, especially for those residing in rural areas, the poor, women and children (NRHM, 2007). The mission was started in late 2005 and aims to provide effective health services to the rural population of India. They have chosen to focus on 18 of the states with the poorest infrastructure (ibid). One big problem with the data collection of many of the programmes is that they are not using the data at a local level. They are just sent upwards through the system, and many of the people that collect them don't see the need for high quality data.

An important strategy for the NRHM is to integrate vertical Health and Family Welfare programmes at national, state, block, and district levels (ibid). This is in compliance with the way PHIS visions the health service (Braa et al). PHIS has gotten accepted by governments as a part of NRHM (Organisation Newsletter, 2007).

Even though the NRHM isn't even three years old yet, there have been some significant gains in the states involved. 2230 PHCs have been upgraded to be open 24 hours per day and 7 days per week. These PHCs have also got two more nurses. Reports show that there is significant increase in institutional deliveries also. A programme called Janani Suraksha Yojana (JSY) has been launched as part of the NRHM. This provides women with some cash and institutional safety when they deliver (JSY, 2007). The reports say that the total number of JSY cases one year was 60.000 and the next year they had reached 21.000 before the end of the year (NRHM progress, 2007).

The next and last part of this chapter, presents the programme and software that I was assigned to integrate into RHIS 2.0.

4.5 Disease Surveillance Integration Project

Another vertical programme in India besides the JSY, is the Disease Surveillance Integration Project (DSIP). They have developed their own software for monitoring health data and reporting the data all the way to the national level. This piece of software is the one I was assigned to integrate with PHIS's own software. That is why I have included a chapter describing the need for this particular software and how it came to be.

Background

DSIP is according to NICD (2004, p. 18) "*intended to be the backbone of public health delivery system in the country. It is expected to provide essential data to monitor progress of ongoing disease control programs and help in optimizing the allocation of resources. It will be able to detect early warning signals of impending outbreaks and help initiate an effective response in a timely manner. DSIP will also facilitate the study of disease patterns in the country and identify new emerging diseases. It will play a crucial role in obtaining political and public support for the health programs in the country*".

NICD (2004, p. 18) states that the reason for initiating DSIP was India is currently passing through an epidemiological transition and although many states in India deliver good healthcare others are lagging far behind. The outbreaks of plague (1994), malaria (1995), and dengue haemorrhagic fever (1996) in different parts of the country highlighted the weaknesses in the surveillance system (NICD, 2004, p. 176). Some states have problems with communicable diseases while other are facing more troubles with non-communicable diseases and this program will have to cater to "*the wide geo-*

political and socio-economic differences" (ibid) in India. Equity in health delivery is one of the highest concerns for the Indian government and through this disease surveillance program they can improve the situation for tribal and underdeveloped parts of India.

Surveillance is important for the early detection of emerging or re-emerging infectious diseases. In the non-existence of surveillance, disease may spread unrecognized by those in charge of health care or public health agencies, because the sick people would be seen in small numbers by many different health care workers. By the time the outbreak is recognized, it may be too late for intervention measures. Constant monitoring is essential for detecting the "*early signals*" of outbreak of any epidemic of a new or resurgent disease. For disease surveillance to prevent emerging epidemics, the time taken for effective action will have to be short (ibid).

They analysed the current state of surveillance activities in India and came up with a list of factors that was making the process inefficient (ibid, p. 29):

- There are a number of parallel systems existing under various programs which are not integrated.
- The existing programs do not cover non-communicable diseases.
- There is need to bring the medical colleges and large tertiary hospitals in the private sector into the reporting system as well as the utilization of laboratory facilities.
- The laboratory network needs to be improved and there is a need to prescribe clear cut thresholds for response at each level.
- Surveillance is necessary not only for detection of epidemics, but for rapid response to arrest spread of disease and to generate essential data for decision making on a regular basis. Presently, surveillance is sometimes reduced to routine data gathering with sporadic response systems.
- There is a need for increased use of information technology (IT) in order to ensure that information is gathered rapidly and responses made immediately. IT can also be used to analyze and sort data so as to predict epidemics based on trends of the reported of disease so that preventive action can be taken.

Based on the above mentioned report, the Ministry of Health and Family Welfare made a list of recommendations for how DSIP should be executed. Here is a list of the most important points (ibid, p. 29-32):

- Health providers at all levels ought to understand the scope and benefits of disease surveillance and be able to delineate surveillance related activities from other health services.
- Surveillance formats should be simple, requiring minimum necessary information to be obtained and readily available as printed forms. The health workers must complete these during their contact with patients at home, field or health facilities.
- Surveillance should be restricted to 10-15 health conditions that are of public health significance in the area.
- Interval between first information and confirmation of outbreak needs to be minimized to avoid delays in response.
- An institutional mechanism is to be put in place where field staff makes proactive efforts to develop personal rapport with the community and solicit their support and cooperation in various health programs including disease surveillance.

- Social mobilization campaigns must address the prevalent socio-cultural beliefs and issues relating to gender disparities.

History

The DSIP project was initiated in 1998 when the Government of India (GOI) sent a request to the World Bank to support preparation of a disease surveillance project. After discussions they agreed to collaborate on preparing such a project (ibid, p. 32). In September 2000 they started the preparations with a meeting in Hyderabad, India. Four states, GOI, World Health Organisation (WHO), Centre for Disease Control (CDC), and other Indian organisations like National Institute of Communicable Diseases (NICD) and Indian Council of Medical Research (ICMR) were present at this meeting. After the meeting Ministry of Health and Family Welfare (MOHFW) set up a steering committee to oversee the projects progress (ibid, p. 32).

According to NICD (2004, p. 33) health is administered by the different states in India and is only guided by the MOHFW. The states have initiated multiple health programs to increase the health of the people. The vertical diseases control programs ran by the government are also mediated through the states. These programs have only been partially successful and NICD (2004, p. 33) blames this on the wide variation in socio-cultural differences and health infrastructure in India. Therefore it was important to have a grass root level approach to the development of DSIP.

Three state level workshops were held with the objective of understanding the current surveillance activities in the state. There were two main objectives with these workshops: understand how they can improve surveillance activities in the state and establish which diseases that would require surveillance. A bottoms-up approach was conducted during these meetings and they included grass root level workers from PHCs, Block and District level administrative officers among others. A national level project implementation plan (PIP) was created based on the workshops and PIPs from other states (ibid). In June 2001, a national level workshop was conducted where they decided to initiate the development of a surveillance system (ibid, p. 34).

The project is set to last from 2004 until 2009 (ibid, p. 178). The initiation will be divided in to three parts from 2004-2007 where 8 territories will start in 2004, 13 territories will start in 2005 and 13 more territories will start implementing in 2006 (ibid, p. 15). The total cost of the project is 4,083.60 million rupees (ibid, p. 16).

Technologies

Their plan was according to NICD (2004, p. 87) to have a web based solution, but it was up to the vendor and the actual requirements of the software. Their database solution was planned to use a lightweight database like MS Access at the client computers while using Oracle, DB2 or similar at the higher levels. It was important that it was widely supported and used RDBMS. Computer viruses are a problem in India and they had plans to secure their computers by not placing any CD-ROM or floppy drives to decrease the chance of virus attack. Should one computer be infected it would be immediately replaced by a new one and the old will be cleaned up.

The use of DSIP

DSIP will, depending on the level of expertise of the health staff, conduct surveillance under three categories. Syndromic diagnosis is made based on the basis of history and clinical pattern by paramedical personnel. The cases identified through the presenting symptoms are classified as "*suspect cases*" of a certain disease condition. An example is a case of fever with rash will be classified under the syndrome "*fever with rash*" and not as measles. Presumptive diagnosis is decided on typical history and clinical examination by medical officers, while confirmed diagnosis is based on a laboratory result (Manual, p. 6).

The reporting unit is what generates data and feeds it to the surveillance system. The primary reporting unit is the health workers at the sub-centres. Reporting units from the rural areas include Sub-centres, PHCs, CHCs and district hospitals while in the urban areas it might be urban hospitals or medical college hospitals (ibid, p. 7).

The health worker is the most important personnel for syndromic surveillance and they report from the sub-centres of the PHCs and urban health centres. On their routine visits to the village and urban wards, the health worker will note in their register the syndromes which are under surveillance. Every Monday the health worker will fill out Form S based on the data he/she collected for the last week and it will be given to the Medical officer in charge at the PHC. The report, will then immediately forwarded to the District Surveillance Officer (ibid). The transmission of forms is either done manually or by phone between the sub-centre and PHC (ibid, p. 8).

Sadly I didn't get any meetings with the person working with DSIP when I were in Bhopal, India. The plan was that PHIS India was going to get a meeting with this person which would allow me to get information about the usage of the software. It is not given that they use the software as presented in their documentation. The documentation says that they report weekly and this is what the system allows for, but it also says that they can report weekly in a case of widely spreading diseases. This seems hard to accomplish in the software I have seen.

More information about the analysis I have done of the software will follow in the empirical part of the thesis. This will explain how the reports function, the formats used and how I would be able to replicate it.

Why integrate DSIP into the Rural Health Information System?

The RHIS 2.0 software is technically superior to DSIP in many ways. In this part I will try to explain why both PHIS and I believe that the data should be entered into their software instead of into the software provided by DSIP.

The RHIS 2.0 is platform independent, which means it can run on any operating system that has a working Java implementation and a Java web server. This means that the use of RHIS will not be locked to any vendor of operating system in any way. It also means that you can run free software to decrease to need for costly license fees. Another important factor is that RHIS is open source and there are no license costs included in the acquisition of the software. The users of RHIS also have full control over the source code and can easily change functionality to match their own needs. This means that they don't have to pay big fees to match changes in their requirements.

DSIP is (in my impression) locked to use the MS Access database as its backend. When using RHIS 2.0 you are free to choose between free alternatives to big scalable proprietary solutions. DSIP can only be used as a desktop application, while RHIS 2.0 can be used both over the Internet and as a desktop application.

While DSIP only support one vertical programme in India, RHIS 2.0 can support multiple programmes at once with the flexibility of defining multiple DataElements and DataSet. In RHIS 2.0 you also have the functionality to add Indicators and validations for each DataElement. If they want to change only one element in DSIP they will have to change the whole software, and redistribute it once more while RHIS 2.0 gets all its definitions from its database.

Now that I have presented all the important background information it is time to present my literature needed for the thesis.

5 Literature

In this chapter I will present theories, models and research that I find relevant to present and discuss my case and my research objectives. This chapter is split into three different parts with its own sub-parts; Software development; Open Source Software; and Intermediary Objects.

5.1 Software development

In this chapter I will present information about different theories in software development. Distributed development is presented because development in my project is spread around the world. I will also present a definition of what proprietary software means.

Distributed development

In a world that has a rapid growth of globalisation it becomes more and more common to outsource or offshore work to other countries or parts of the world. Therefore distributed development is an important part of today's work flow. In my case there is also a great need for distributed development since members of the organisation are spread across countries and time zones.

Lings et al (2007, p. 1) writes that distributed development of Software Systems is an important subject for organisations today, especially given the current trend towards globalisation. The main difficulties of distributed development of software systems Lings et al (ibid) reports is in the complexity of maintaining good communication, coordination and control when teams are dispersed in time over multiple time zones and space, as well as socio-culturally. These three activities are all used to facilitate efficient use of numerous developers.

Lings et al (2007, p. 2) quote Carmel & Tija when they define communication as "*the exchange of complete and unambiguous information – that is, the sender and receiver can reach a common understanding*". The process includes the tools used to facilitate the interaction and the transfer of information.

In distributed development there can be multiple types of distribution. The can be temporal, geographical and socio-cultural (ibid) distances between the actors. Temporal distance is defined by Lings et al (ibid) as "*directional measure of the dislocation in time experienced by two actors wishing to interact*". The temporal distance can therefore be caused by working in different time zones or different work patterns. Usually low temporal distance means better opportunities for synchronous communication, but may decrease different management options. They (ibid) define geographical distance as "*a directional measure of the effort required for one actor to visit another at the latter's home site*". This is best measured in ease of relocation rather than in the distance itself. When the geographical distance is low, it is easier to co-locate and undertake team working. The social-cultural distance is classified as "*a directional measure of an actor's understanding of another actor's values and normative practices*" (ibid). This is complex dimension involving organisational culture, national culture and language, politics, and individual motivations and work ethics. Another complex factor of the social-cultural distance is that developer A might be closer to developer B than B is to A. Improved communication and lower risks are some of the possible benefits of low cultural distance (ibid).

Lings et al (2007, p. 4-8) have defined a set of guidelines for how to undertake successful distributed development. The reference model is an ideal model and it will be impossible for an organisation to fulfil every point, but it should be used to compare against today's practices and be used as what it is, a reference model. Here is the list of things that should be thought through when embarking on a distributed development project:

- Have a clear distribution rationale.
- Clarify all understandings, clarify and agree on goals before the project starts.
- Leverage modularity to reduce the need for intensive collaboration.
- Use cultural mediation, a person who can be a link between the teams.
- Facilitate human communication by providing rich technologies and improving the efficacy of standard technologies.
- Manage processes by having a team leader which regularly has contact with the different teams.
- Develop a sense of "*teamness*" by having a common platform (i.e. a website) with updated information about progress, the teams and goals.
- Encourage temporary collocation where understandings could be reached and strategic thinking can be done.
- Address heterogeneity in methods, tools and terminology, but plan everything in advance.
- Develop an effective tool base which can be replicated in each site.

Proprietary software

Proprietary and open software are two different ways of controlling the use and commercialising of the software. GNU (2008) defines Proprietary software (Non-free) as software that is not Free or Semi-free. The definition follows with "*Its use, redistribution or modification is prohibited, or requires you to ask for permission, or is restricted so much that you effectively can't do it freely*" (GNU, 2008). Proprietary Software (2008) defines Proprietary Software to be software that is imposed a set of restrictions; "*on use or private modification, or with restrictions judged to be excessive on copying or publishing of modified or unmodified versions*". The restrictions are enforced by the owner of the property and can be enforces either by law or by technical means. This is normally enforced by not releasing the source code to the public and only releasing compiled software that is understood by the machine (ibid).

Proprietary software can be both shareware and freeware, while it also can be commercial software. Shareware and freeware can both be sold for zero cost, but enforces some restrictions or is only free for a trial-period. Some proprietary software also comes with source code, but it will be restricted by a license or a non-disclosure agreement from redistributing modifications or sharing the software (ibid).

When you want to figure out how proprietary software works you can reverse engineer it. Reverse engineering (RE) can be defined as "*is the process of discovering the technological principles of a device, object or system through analysis of its structure, function and operation. It often involves taking something (e.g. a mechanical device, electronic component, or software program) apart and analyzing its workings in detail, usually to try to make a new device or program that does the same thing without copying anything from the original*" (Reverse Engineering, 2008). Reasons for doing RE can be interoperability, lost documentation, product analysis, security auditing and more. According to (ibid) RE can be split into two different types. One were the source code for the software is already accessible, but where the documentation is missing or in poor quality. The other type which is the most common understanding of the term is where the source code is not available and RE is viewed as the process of creating one possible source code (ibid).

Now that I have presented the relevant theories within general software development I will present the background for how the project that I have been involved in is developed.

5.2 Open source software

In this chapter I will present some background information about Open Source Software (OSS) and some theory on how an OSS project is administered and managed. I believe information about what an OSS project is, how they communicate and develop is important to understand my case and work. Even though the development of RHIS 2 does not follow the de facto standard of OSS projects, if there is such a thing, I will describe what is needed to undertake an OSS project.

Background of Free and Open Source Software

Free and Open Source Software (FOSS) began in the beginning of the 1980s when the computer manufacturers started licensing their software. Richard Stallman who was working at the artificial intelligence (AI) lab at the Massachusetts Institute of Technology (MIT) at the time did not like this direction of software development. He did not like the fact that you could not share your software with your neighbour freely. He decided to resign from MIT and start the Free Software Foundation and the GNU project. He also created a new copyright license that guaranteed that the source code would be free; the GNU General Public License (GPL). The GPL says that code may be copied and modified without restrictions, and that both copies and derived works must be distributed with the same license as the original with no additions to it. This protects the software from being put into proprietary software from the enemy (Fogel, 2005, chapter 1).

With help from likeminded hackers (as in someone who enjoys to program and are skilled) the GNU project was able to release free components for many important parts of an operating system (OS). They released a text editor called Emacs, and a C compiler called GCC which because they were quality software and ran on many computer systems were successful and gained a large fan base. At around 1990 the GNU project had created most of the required parts of an OS, but they were missing the kernel which boots the system and manages the interaction with the hardware. The GNU project's solution proved to be hard to implement. Luckily they got help from Linus Torvalds who had created his own free kernel, called Linux. Now they were ready to release one of the first free operating systems (ibid).

Problems arose when the corporate world started taking interest in FOSS. The word free caused corporations to think of it as zero cost software, but there is a difference between *gratis*/*libre* software and software with an open source and OSS license. The normal way to explain it is to think of it as free as in *freedom*, and *free speech*, not as in *free beer* (ibid). In the late 1990's the term Open Source was created as an alternative to *free software* and they created the Open Source Initiative

(OSI). This was not only to clear up the free issue, but to make it more suited for the corporate world (ibid).

Problems in OSS projects

Monteiro et al (2004, p. 1) reports that an important challenge in OSS projects is to "*cultivate and nurture a motivated community of developers*". It is often the case that OSS projects have a constant lack of manpower. This is a big threat to the project itself and causes fragility. To keep the network of developers together Monteiro et al (2004, p. 3) suggest ongoing effort of "*recognizing the constitutive element of repetitions and various social and symbolic forms of reassurances*". The organizational, ritual and technological mechanisms of the project are what keep the project together (ibid).

Advantages of OSS

One of the advantages of open source software is stated in Raymond (2001, p. 30) "*Given enough eyeballs, all bugs are shallow*". This means that if the project has a large community that understands the system and the source code, it will be able to discover a large number of bugs.
To have a successful OS project you will have to implement technologies that help manage information and this will have a big impact on how well your project do (Fogel, 2005, p. 28). I will now present some of these technologies.

Development in OSS

An OSS project should be developed in a modular fashion so that it is possible for distributed developers to work on it simultaneously. Parallel development is an important characteristic of OSS development and it is a result of a modular structure of many OSS projects (Câmara & Fonseca, 2007, p. 130). This can lead to substantial performance gains with a large number of developers collaborating efficiently (Weerawarana & Weeratunga, 2004, p. 19). It is also important to have a well designed architecture that is understood by the developers to enable collaborative work (Câmara & Fonseca, 2007, p. 130). To support distributed development and a modular structure it is imperative that a project implements a version control system.

Fogel (2005, p. 38) defines a version control system as "*a combination of technologies and practices for tracking and controlling changes to a project's files, in particular to source code, documentation, and web pages*". Today it is expected that every serious OSS project utilises version control on at least their source code and that it is used properly. Fogel (ibid) continue to say that version control helps with many different aspects of running an OSS project: "*inter-developer communications, release management, bug management, code stability and experimental development efforts, and attribution and authorization of changes by particular developers*".

According to Fogel (2005, chapter 3) everything that have the possibility to change, should be kept under version control. Another user friendly feature is the ability to browse the repository on the web without installing any version control software. By using a system that distributes e-mail whenever someone commits new code to the repository is a good way of keeping the developers up-to-date on what is happening. Some sort of authorisation will have to be decided so that for example only registered people can commit code or commit to certain areas of the repository. Concurrent Version System (CVS) and Subversion (SVN) are the most used version control system today. CVS is older and many people are familiar with it, while SVN has more features (ibid).

An important function of an open source project is to have a list of all the problems, errors and missing functionality in the software. This is called a Bug tracker or Issue tracker. It doesn't have to only contain bugs, but can also store information new features, patches and one-time tasks. It is important to have an easy interface, but a registered issue should contain as much information about the issue as possible. The bug tracker should also be connected to a mailing list so that the developers will get notified when a new bug enters the system. Once an issue is verified it will be assigned to a release and a developer. When the issue is resolved the tracker should be updated with its status (ibid).

Communication in OSS

Fogel (2005, p. 30) start the chapter about mailing lists with "*Mailing lists are the bread and butter of project communications*". The mailing list does not only have to be set up, but managed continuously. The mailing list should have an easy way to subscribe and unsubscribe, and the administrator must be able to monitor and filter the mails sent to the list. One common approach is to only let people who subscribe to post to the list, which will prevent spam being sent to the list. A mailing list should also be able to create a searchable archive for all the posts sent to it. It is important that you acknowledge the need for a mailing list from the beginning of the project (Fogel, 2005, chapter 3).

Another popular form of communication in OSS projects is the possibility of real-time chat. The most common is the Internet Relay Chat (IRC). This works as a forum where you can ask a question and (hopefully) get an instant response. IRC have different channels where everyone can join and talk freely in a channel, where everyone can see what others write and jump into the conversation. It is possible to set this up yourself or use a free service so you don't have to maintain it yourself. Some of the service can also be automated with the use of bots which can help you maintain the channel (ibid).

Documentation in OSS

The website for an Open Source project is important. This is were the project presents a clear overview of the project and bind together all the different tools like bug tracker, version control and mailing lists. There is an important choice of picking how to host the site, many OSS projects use canned hosting sites where you don't have to administer everything yourself. There are both pros and cons when choosing a canned hosting site; they have chosen the different software, they take care of backup, they have great bandwidth and storage, they will probably have advertisements and many other reasons both for and against (ibid).

To document the project and its ever changing functionality and content it is convenient to have an easy way to add, change and delete documentation. A wiki is a suitable approach to this problem. Such a site allows people to easily add and edit information on the web. A wiki will require some administration to make it work properly; here is a list of problems that will likely occur while using a wiki page: it will have a lack of navigational principles, it will have duplicate information on different pages, and it will have inconsistent target audiences (ibid). The users of this documentation will also be spread across borders, continents and cultures, so it is important to have strict guidelines for its use.

OSS in developing countries

Information and communication technology (ICT) has become an important part of the recent transformations in our society. According to Camara & Fonseca (2007, p. 121) most of these changes are taking place in industrialised countries some developing countries are starting to see the advantage of using ICTs. They continue to say that it is a consensus in that the use of ICTs can be helpful for developing countries to reach their development goals.

One of the most interesting reasons for adopting OSS is the reduction of licensing costs (Camara & Fonseca, 2007, p. 121). They also report that avoiding being hostage to proprietary software, advancing knowledge more quickly, and helping to set up an information economy are all good reasons for adopting OSS in developing countries (ibid). Weerawarana & Weeratunga (2004, p. 28) also reports that independence, autonomy and security are all reasons that especially governments in developing countries choose OSS.

For an ICT to be sustainable in a developing country it must absorb the knowledge embedded in the technology itself (Camara & Fonseca, 2007, p. 122). Another important factor for choosing OSS for developing countries is the fact that it will have an active part of integrating the ICT and the software should be both an instrument for social change and a way of developing software development skills inside the country (ibid). For the implementation to work properly it is important for the country to have strong and wise policies. It should be a combination of "*institutional vision, qualified personnel, and strong links to the user community*" (ibid, p. 130). For the software to be viable it should also be government funded (ibid). Another important point from Camara & Fonseca (2007, p. 121) is a quote from The Success of Open Source, by Steven Weber (p. 56);" *the essence of open source is not the software. It is the process by which software is created*". According to Weerawarana & Weeratunga (2004, p. 8) the generation of capacity and local skills is done through establishing localisation centres, sharing localisation skills with other developing and less developed countries, establishing OSS solution and Research & Development centres.

Software used by PHIS to support OSS development

I wrote earlier about the need for a number of programs to maintain an open source project. In this chapter I will briefly explain the software utilised by PHIS to support the development of RHIS 2.0 in an open source approach.

The Subversion project was created to build a version control system that is a "*compelling replacement for CVS in the open source community"* (SVN, 2008). The software is open source and the license is based on the BSD license. The software is created for use in Linux, but is possible to use in Windows via Cygwin or with the use of TortoiseSVN, which is a graphical version of the subversion client software. In the RHIS 2.0 project Subversion is used to contain, manage and distribute all the important source code for the project. It contains all the older versions (*tags* folder), changes to existing functions (*branches* folder), new functionality (*incubator* folder) and the current version (*trunk* folder).

Trac is an enhanced wiki and issue tracking system for software development projects. The software is based on the idea of minimalism and tries to stay out of the way from the user (Trac, 2008). The Trac implementation for RHIS 2.0 was set up in May 2007 and so far (30.04.2008) 293 issues have been reported into the system.

Confluence is the wiki solution PHIS uses for its web site and documentation needs. It is an enterprise wiki that it makes it easier for a team to share information and collaborate. The software has a user-friendly "*what you see is what you get*" interface, export to PDF and good security solutions (Confluence, 2008). Confluence is used to present their homepage with its documentation for among other things RHIS 2.0.

Mailman is software for managing e-mail discussions and electronic discussions. It has an integrated web interface for management of the lists and users. Mailman supports built-in archiving, automatic bounce processing, content filtering, digest delivery, spam filters, and more. The software is open source and is licensed under the GPL (Mailman, 2008). PHIS uses Mailman to support a multitude of lists concerning PHIS, RHIS 1.X, and RHIS 2.0 on different levels.

The project also has an IRC channel which gives users and developers a chance to communicate real-time with each other. They have decided to not host this service by them selves and used an already existing network called Freenode.

I have with this chapter described Open Source development in different ways to help support the understanding of my case. The next chapter will contain the main theoretical literature and research, which I will use in my discussions.

5.3 Intermediary Objects

In this chapter I will present the theory I base most of my discussion on. I have decided to use this concept as a way to explain my system development process and to analyse my use of the different objects. There is not much research with the use of this concept yet and think it is interesting to apply this to a software development process. Intermediary objects can be used to describe the different objects used while developing software, i.e. documentation, design documents, prototypes and much more. First I will present where the theory of Intermediary objects are drawn from, Boundary Objects.

Boundary Objects

A map may be drawn in different forms, but it has a set of standards which it should follow. One map may have a different meaning and function for different communities. A tourist may use it to find his way to an attraction, while a road worker might use it to tell where the road should be constructed. This is the basis for boundary object, a standardised object which could be used by multiple groups for different purposes.

Since Star and Griesemer (1989) initiated the concept of boundary objects, it has been used in a wide variety of research areas including research on collaborative information systems, organization science, and information science (Lee, 2007, p. 308).

Lee (2007, p. 308-309) writes that boundary objects are described as objects, that coordinate the perspectives of various communities of practice. Boundary objects rely heavily on the concept of standardization and examples of boundary objects are typically things with a standardised structure such as forms, maps, and grades, or things with a naturally predetermined structure. Boundary objects are created when groups from different worlds work together. Shared work creates objects which inhabit multiple worlds simultaneously.

Research employed the concept of boundary objects to show that a single object can be used for different purposes by different people (Larsson, 2003), to theorize information systems as boundary objects between communities of practice (Pawlowski et al., 2000), and to explore activities surrounding boundary objects within information or work flow (Mambrey and Robinson, 1997; Lutters and Ackerman, 2002) (references from Lee et al., 2007).

The concept of Intermediary Objects comes from the concept of Boundary Objects developed by Star in 1989, and IOs were created as a separate concept to analyse "*the nature and the content of the various objects produced during the design process*" (Papadimitriou & Pellegrin, 2007, p. 438). Papadimitriou & Pellegrin writes that Intermediary Objects of Design can act as boundary objects to enhance the communication among the different "*worlds*" of the design process, but are also composed with intermediate states which represent the future product (ibid). The next chapter defines more clearly what the definitions of Intermediary Objects are.

Definition of Intermediary objects

Vinck defines the intermediary objects (IO) to be any physical, graphic or textual entity being between several actors or as an outcome produced between several steps in a course of design. El-Kechaï & Choquet (2006, p. 2) writes that IOs are "*considered as a means through which a design process can be read*".

The definition of IO from Boujut & Laureillard (2002, p. 509) is that IO "*is a general category embracing all types of artefacts, whether physical (plans, mock-ups, sketches, etc.) or virtual (CAD models. Calculation results, etc.), produced by the participants during their work*". They (ibid) continue to say that IOs have two dimensions. The first is that they are related to the action itself, for example designing a product. The second dimension that they are a means for coordinating designers' activity. Papadimitriou & Pellegrin (2007, p. 437) has a similar definition of IOs which is "*all objects such as designs, prototypes, descriptive documents and pilot implementations produced by the project team and enclosing an intermediary representation of the final deliverable in some tangible form*".

According to Bojout & Blanco (2003, p. 211) the concept of Intermediary objects has already proved efficient in the analysis of a large number of case studies providing a good analysis tool of the interfaces between participants, departments, companies, etc. as well as in the development of co-operation within design processes. It provides a theoretical framework to analyse the role of objects (i.e. artefacts, mock-ups, CAD models, plans, sketches, etc.) in the design process. Intermediary objects are also intermediate states of the product if we consider the objects as mediators translating and representing the future product.

Papadimitriou & Pellegrin (2007, p. 441) gives the IO an action perspective. The object can be defined as Open or Closed. The object is defined as Open when the object is allowing interpretation and it is Closed when the object is considered definite and it is not subject to change. They found that the majority of IOs are Open, but some IOs tend to become closed towards the end of the project. The Closed IOs are mostly technical and codified (ibid).

Choosing Intermediary objects

The reason for not using Boundary Objects as my theoretical approach is the need for standardization mentioned in the earlier chapter. Lee (2007, p. 309) also says that "*the concept of standardization is important to the boundary objects itself*". The objects I have encountered and used in my case is not always standardised and complete, therefore does not perfectly fit the concept of Boundary Objects.

Because of this my choice was placed on the concept of Intermediary Objects which does not rely on standardisation like BOs, and is more appropriate to represent a design or product in an evolving state.

Intermediary Objects in software engineering

IOs are usually found in (industrial) design in the literature (Bojout & Blanco, Boujut & Laureillard). I believe IOs can also be used to explain the design and development process of computer software. In distributed development projects and in OSS projects like my case there will always be need for up-to-date design documents, requirements and more, that is shared between many different types of workers. Intermediary Objects in software engineering can also help communication between different worlds if they have defined some form of standardisation.

IOs can be used to view and analyse the documentation, design documents, UML diagrams, system requirements, prototypes, source code, source code management software and discussions about the software. Any state which represents the product as of now or the future product can be described as an IO.

Research on Intermediary objects

Jean-Francois Boujut and Pascal Laureillard have used IOs in an 18 month case study. The case study was conducted in a French truck manufacturer where they designed a front axle because they feel that co-operation is an important issue in design today. As a result they propose a conceptual framework for the development of design co-operation within design organisations (Boujut & Laureillard, 2002).

Hassina El-Kechaï and Christophe Choquet studied a collective design process in developing a training system for apprentices. The aim of their research was to analyse the structure of their design process in order to develop a model to support it, and based on that be able to develop a tool. Their research consisted of leading the designers to reflect on their design intensions, collecting the IOs and analysing them and formalising the results of the analysis in order to figure out the design process involved (Kechaï & Choquet, 2006).

K. Papadimitriou and Claude Pellegrin used IOs to understand the sense making in an organisational change in a Greek Bank. The projects purpose was to establish sales roles within branches, to centralise branch back-office activities, and to develop telephone banking. They analysed a multitude of IOs to understand the project life cycle and the different types of IOs used in different time frames of the project (Papadimitriou & Pellegrin, 2007).

Jean-Francois Boujut and Eric Blanco used a case study from a large truck company where the researcher was supposed to propose improvements in the organisation of information flow. The authors use Intermediary Objects as a conceptual framework for the involvement of objects in the design process and they demonstrate the advantage of this concept in the analysis and modelling of design situations (Bojout & Blanco, 2003).

Charlotte Lee conducted an ethnographic study of collaborative work. The project was a multidisciplinary collaboration in creating a museum exhibition. She uses this project to question the wide use of Boundary Objects and discuss the Intermediary Objects as a part of her research (Lee, 2007).

Papadimitriou and Pellegrin’s types of Intermediary Objects

In their (Papadimitriou & Pellegrin, 2007) research they statistically found three different types of IO based on their research of organisation changes. These three were found statistically while they found two other kinds of IOs through observation of the case material.

They found the *Oral ideal IO* often occurred in the beginning of the project and “*allow the diffusion of the project principles to the newly founded project team*” The second type they found was the *Written discourse* IO which is present in the whole project duration but mostly in the first half of the project. They may contain elements of ideals and techniques that make sure that there is continuity to previous or following objects. The third statistically proven IO is the *Codified technique*. They are the media for “*integrating principles and casual relations to action frames, routines, performance models etc*” (Papadimitriou & Pellegrin, 2007, p. 443-444).

The two other types of IO emerged during the observation of the data, because they had a high impact on the project. They say that they both share the same detail that they are both *open* and *closed* at the same time. The first type is *Quasi-technical* objects which combine written discourse and codified techniques that are difficult to detach from each other. They function as a support “*between ’business’ actors that provide operational requirements and ‘technical’ actors who implement them technically*”. *Pilot applications* are the second type of object. They usually occur in the time where the project is coming to an end and the implementation starts. From the project view they are closed, because now “*all alternatives have been explored, techniques, underlying discourses and ideals have been agreed and implemented, and the project team is now ready to deliver the final* *product*”. But a pilot application is also open in a sense that it can be modified through a real test of the product (ibid).

This chapter have presented all the relevant literature, software development, open source and the concept of Intermediary Objects. The next part of the thesis will present my methodologies for undertaking the research presented later. It will also present my research approach and analysis.

6 Methods

This chapter will contain the research methods used to undertake this thesis. Qualitative research will be presented together with Action research. It will also include my approach for the research, explaining how I gathered my data and did my research. For analysing my data, I have used a qualitative research software which will also be explained in this chapter.

6.1 Qualitative Research

In this chapter I will present how a qualitative research project should be performed and some information about my approach to using this methodology.

What is qualitative research?

Qualitative researchers typically rely on four methods for gathering information: participation in the setting, direct observation, in depth interviews, and analysis of documents and materials (Marshall et al, 1998). According to Silverman (2005, p. 55 the) validity of qualitative analysis depends more on the quality of the analysis more than the size of the sample (ibid).

Silverman (2005, p. 110) writes that in quantitative research it is expected that you begin by establishing a set of variables and methods. Miles and Huberman (1984) suggests that qualitative researchers have a range of options in how far they use what Miles and Huberman call "*prior instrumentation*" (i.e. predefined methods and measures). They present three different options in which this can be defined: No prior instrumentation, Considerable prior instrumentation and an open question. Miles and Huberman (ibid) also write that it is worth trying to structure your work early and "*no prior instrumentation*" should not be considered the default option for qualitative studies. My research started with no prior instrumentation because my research objective was not clearly defined when I started doing the research for this thesis.

Before I travelled to India my thoughts on the title or topic for the thesis was something like "*How can PHIS gain sustainability through integration?*", which is in my opinion very vague. This was when I thought that more of my work was going to be related to integrating other Indian HISs. This changed along the way as there was less and less focus on integration from both the organisation and me.

Diary keeping is an important part of qualitative research. According to Silverman (2005, p. 251) a diary should contain your research activities with dates, your reading, details of data collected, directions of data analysis including "*special achievements, dead-ends and surprises*" (ibid), your own personal reactions, and your supervisor's reactions and suggestions.

Since my initial plan was to focus on the integration I tried to write as much as possible about this topic in my diary. Since this was my main focus in data gathering, it is also used as my main empirical findings in this thesis.

During my field work I had some periods where I didn't know how to focus my field notes, and my supervisors were stressing me to improve the way I wrote them. I was feeling a bit lost at the time.

Field note 6.1 is an excerpt from an e-mail I wrote to my supervisors around the middle of my trip which described how I felt at the time.

> *It is totally understandable that we have to start taking better notes*
> */ notes that fit our research goals. The problem is we feel like we*
> *have no where to start.*
>
> *This is how I feel right now, and this is my honest thoughts. I feel*
> *like it had been better for me to travel next semester. I would*
> *probably have matured more and gotten a better picture of have master*
> *paper should be, what to write about, have more knowledge about RHIS,*
> *had the opportunity to read some master papers and got done with my*
> *courses. I have to spend this fall taking three (maybe four if I fail*
> *Jens' class; haven't got the new results yet, from januray :P) classes*
> *this fall and will not have to much time focusing on my paper and RHIS*
> *development.*
>
> *Now that I are down here I have to do the best I can. I cannot say*
> *that I regret going in no way. I really enjoy traveling around India*
> *with my travel partner. But I feel I might had gotten more out of it had I traveled*
> *later. The reason I wanted to travel this semester was because my travel partner was*
> *going and I already knew him and we had worked on some thing the*
> *Indians might enjoy and use.*

Field note 6.1 - From My personal diary, 18.03.2007

Even if there were some times I was frustrated and lost I feel the data I had gathered was enough to write a good and complete story of what I had accomplished in India.

Why did I use qualitative research methods

I was presented with the opportunity of travelling to India and do whatever work was necessary for the growth of PHIS, as an organisation in India. The plan for our work was to mostly help with integration of other Indian software and train Indian developers. With this background I didn’t have a totally clear idea of what to study and I approached the trip with an open mind. Qualitative research and Action research (see below) is good for doing exploratory research.

Limitations of qualitative methods

All research is contaminated to some extent by the values of the researcher. Only through those values do certain problems get identified and studied in particular ways. Even the commitment to scientific (or rigorous) method is itself a value. Also the conclusions and implications to be drawn from a study are largely grounded in the moral and political beliefs of the researcher (Silverman, 2005, p. 257).

When doing qualitative research it is important to be careful in how you specify your claims based on your approach. It is important to note that you research is only one way of “slicing the cake” and other research using another approach may not be directly competitive (Silverman, 2005, p. 65).

Generalizability is a standard aim in quantitative research and is normally achieved by statistical sampling procedures. Such sampling has two functions; "*if the population characteristics are know, the degree of representativeness of a sample can be checked*". Second, such representativeness allows you to make broader inferences: "*the purpose of sampling is usually to study a representative subsection of a precisely defined population in order to make inferences about the whole population*". Such sampling procedures are, however usually unavailable in qualitative research. In such studies, our data are often derived from one or more cases and it is unlikely that these cases will have been selected on random basis. Very often a case will be chosen simply because it allows access, like in my case (Silverman, 2006, p. 303-304).

In qualitative research there is also the problem of *anecdotalism*. This is based on the problem of convincing themselves and their audience that their findings are indisputably based on a critical investigation of all their data, and do not only rely on a small set of well-chosen examples (Silverman, 2005, p. 211).

Silverman (2005, p. 211) quotes from Mehan's *Learning Lessons* from 1979 where he says that the researcher seldom provide the criteria or reasons for choosing certain data and excluding other. As a result of this it is difficult to determine the typicality of the findings of this data. Mehan also says that as the researcher abstracts data from its original form to summaries it loses its original form.

Now that I have presented what qualitative research is, it is time to present my used method, which is based on Action Research.

6.2 Action Research

Action research will be presented in this chapter. The reason for choosing an active approach as my method is the motivation of becoming a better system developer, which is not as easy to accomplish with only quantitative methods. The opportunity of travelling to India involved helping and supporting the Indians in development and it implied that I had an active participation in my research. I am also under the impression of studying the use and effect of Intermediary Objects is best done with a qualitative approach. This research approach is also widely used within the PHIS network of both master students and PhD researchers.

History of Action Research

According to Baskerville & Wood-Harper (1998, p. 92) the origins of the action research (AR) method was developed when the disasters of World War II was over and there were massive changes in the research field of social science. Kurt Lewin was credited for developing the method for his study of social psychology within the framework of field theory.

Because of the scientists troubles of understanding the complex causes of social illnesses caused by fighting on the battlefields and being in prisoner-of-war camps the idea of social action came to be. Now the researcher changed some aspect of the patient's being or surroundings in every experimental case. Since the scientist and therapist were one, the scientists were now participants in their own research, and the effects of the changes were recorded and studied (ibid).

Action research in all its various forms is characterised by: its multivariate social setting, its highly interpretive assumptions about observation, intervention by the researcher, participatory observation and the study of change in a social setting (ibid).

Action Research in Information Systems

From the 1970s originated as a distinct application area for action research. Mumford's work on ETHICS as an action oriented socio-technical information systems development and Wood-Harper's user of action research is some of the pioneering work (Baskerville & Wood-Harper, 1998, p. 93). During the 1990s the practical domains of IS action research are explored, and many of the IS action research journal publications are inventoried and classified. According to (ibid) IS action research appears to arise from these four distinct streams: the original "*canonical*" form of action research, management consulting, soft systems methodology, and organizational learning. The term action research is frequently mentioned in each stream, but in each, the term has a slightly different contextual meaning.

Learning process in Action Research

Riding et al (1995) write that all different forms of Action Research "*adopt a methodical, iterative approach embracing problem identification, action planning, implementation, evaluation, and reflection. The insights gained from the initial cycle feed into planning of the second cycle, for which the action plan is modified and the research process repeated*". This repeated cycle can be seen in Figure 6.1.

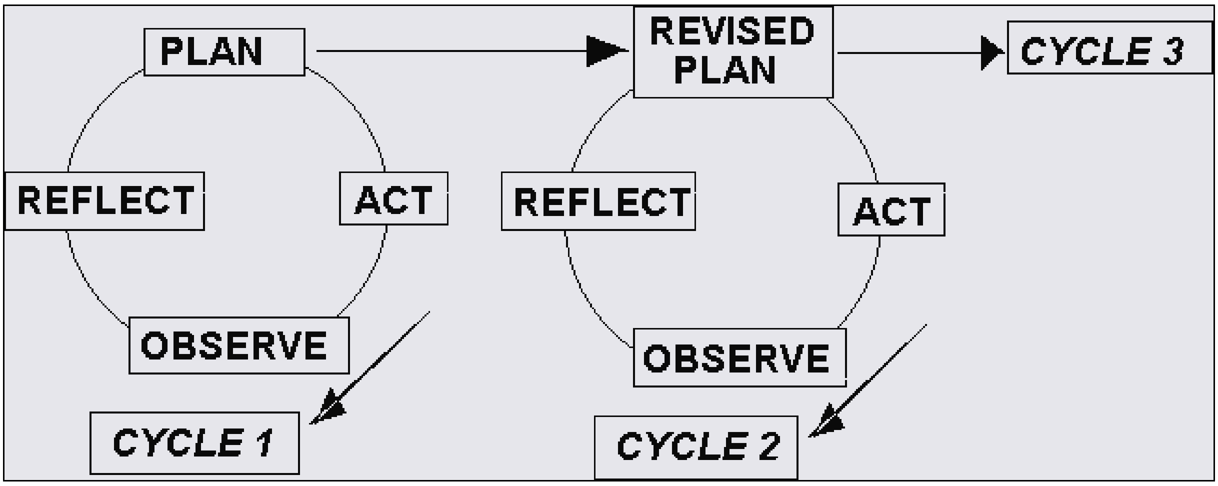


Figure 6.1 – Action research cycle from from Riding et al (1995)

In the beginning of the research project, the researcher plans what he would like to do and what he or she hopes to accomplish by the action planned. The next step is to perform the planned action. When the plan is accomplished the researcher has to observe in what way his actions affected the environment he or she intervened in. In the end of the cycle the researcher reflects on the action taken and the changes observed. Then something in the plan can be altered and implemented again.

The next chapter will describe in more detail my research approach.

6.3 My research approach

In this chapter I will present how I undertook my research project and how it relates to the theory presented here. The chapter will contain how my AR approach was and how I gathered my data from the various data sources. With the use of AR as methodology I follow in the line of other master students doing their master thesis for the PHIS project. Most of them have also travelled to countries and supported the work done there.

Baskerville & Wood-Harper presents (at least) four characteristics that separates different action research approaches. These types include process model, structure, typical researcher involvement, and primary goals (Baskerville & Wood-Harper, 1998, p. 94-95).

Much like the work of Hammer & Champy (1993) I was incorporated into an organisation to help them with a problem or a task they could not solve by themselves. Their approach was to aid them re-engineering their businesses, while I was there to help them integrate other Indian HISs. Their approach is considered to follow the Clinical field work form of action research (Baskerville & Wood-Harper, 1998, p. 102-103). In this type of action research there is a "*highly trained professional who get involved in a helping role with individuals, groups, communities or organisations*". The researcher's role is facilitative and the motivation for the researcher is organisational development and scientific knowledge.

The ETHICS (Baskerville & Wood-Harper, 1998, p. 102) methodology also fairly matches my research approach. ETHICS is a strong participatory methodology intended to produce effective socially compatible IS. One way this method does not match my project exactly, is with regard to the close involvement of end-users. I went to India to get closer to the people that needed the software, but I did not work that closely with the end-users with regard to the DSIP export module.

My *process model* (ibid) started as an iterative one. The plan was for us to integrate multiple different vertical Indian health programs as different cycles, but because of various problems I ended up only creating integration for one system. The whole project ended up as one linear where I would use only one integration project as the background for my research. My research approach had a structure (Baskerville & Wood-Harper, 1998, p. 103) that was *fluid*. All the activities were loosely defined and there were no strict plan from the beginning.

My involvement in the research can be viewed as both *collaborative* (ibid) and *facilitative* (ibid). I was incorporated into PHIS India as a developer for them and I acted as an equal, but I also acted as an expert among the people I worked with. It was a cooperative task where I relied on help from the members of the organisation where the burden of solving the task still relied on me.

The *primary goals* (ibid) of my trip to India and my work on the exporter(s), were *system design* and *training*. My hope was to modify the organisational systems of the Indian primary health sector by letting them use fewer information systems in their daily work and thereby ease their day-to-day work. Another *system design* goal was also to support PHIS India in becoming more involved in the development process. My other goal with the trip was *training*, which is an *educational* goal, where I set out to learn how to work with a global open source software project and see how they utilise the software in the field to improve my developing capacity.

My research started when I got the possibility of joining PHIS and doing development for them. I got the assignment of integrating the Indian software into PHIS's health information software. The first part of the research where done in Norway, where the work done was part of the INF 5750 course at UiO. The work done in this period was not connected to India in any way, as I got all the requirements and help from Norwegian coordinators and developers. In the period between the holidays and the trip to India, they had planned to get me more involved with Indian people, but it never happened.

The research material from India contains constant updates of what I was developing. I tried to include detailed information about the meetings I had with PHIS India personnel and all the discussions related to my work on the integration module. I tried to involve some of the people around me in my work to get a better impression of what is needed. Not having access to stable electricity or the possibility of writing notes when events happened, might have caused some sections where the data is vague and lacking in detail.

Interaction in the development involved mostly high level coordinators and some developers in the context of India. For most of the technical discussions and problems, contact with Norwegian personnel was important. The coordinators (both Norwegian and Indian) where the one defining to some degree how the development should continue, so these are the people who have most relevance to the research. I have also had important contact with PHIS employees from both Vietnam and South-Africa.

I have been participating in meetings with government officials and helped support and evaluate training-sessions conducted with the people who are actually using the software. This gave me good insights of the actual practice of using the software. I have also held a weeklong workshop with Indian developers at a request from the Indian coordinators; this was done to help establish India as base for development. It was very interesting to see the level of competence they possessed and see what they were struggling with. PHIS have been trying to move development to India, but the tools and the big number of frameworks needed to be learned, has caused this not be big a success yet.

Data gathering

I started gathering data when I first realised that my work on integrating Indian HISs into RHIS 2 would most probably be a part of my master thesis. This was in January 2007 before I travelled to India. I started writing a private diary of my daily work. Before that I rely on information from e-mails, SVN logs, deliverables, and documentation. In the diary I noted down what I did on a daily basis especially with regard to the work I did on the DSIP exporter. I also wrote about thoughts I had on the whole PHIS as an organisation and things happening that affected what I did. I wrote about meetings I attended and what happened on the e-mail lists. This period was from the beginning of January to the 12th of February.

My period in India from the 12th of February to the 25th of May I continued to write a daily diary. I wrote much more each day and one day might be up to 2 or 3 pages long. My field notes now contained a lot more personal information as I also wanted to be able to see what I did without regards to my work for PHIS India. At a weekly basis I sent my notes to both my supervisors and got feedback. So every week I had more experience and knowledge about writing field notes. Eventually I also started including important e-mails and pieces of code into the document where I found it appropriate. I did not make use of any framework or tools for writing my diaries. The detail in information varies between days. This is based on what we did, where we were, and how much time I was able to spend writing my field notes. I continued to write some diary through the summer whenever I did anything concerning my work with PHIS. There were no real analysis of the data during the trip, and I my analysis began in the winter of 2007.

My research data therefore consists of my own field notes, e-mails sent between me and coordinators and discussion on the list. It also contains real-life discussions, meetings and mine and other related people's thoughts. My interviews have been done in a semi-structured way and mostly done informally while working.

The actual coding done in the project has been done alone, but important feedback and discussion around how it should be done has played a big part of the outcome. Requirements have been updated and solutions changed based on my discussions with people from India, Norway and Vietnam.

Other data sources

Subversion log

At the beginning of my work on this project I did not take any notes on what I did and when I did it. So using the subversion client I could retrieve information about my progress on the exporter from the fall of 2006. The subversion log was also used in later parts where there were uncertainties in the field notes and I needed to see exactly what was done at which times.

Using mail archive as a log

My mail archive is a very good and reliable source of information. It has accurate information about what I did and when. A lot of the problems I encountered during this process is documented in detail, tried solved through and solved with help from e-mail. This is also a good source of information to complement the e-mails I have written into my field notes.

In this chapter I presented the way I did my research, and in the next I will explain how I analysed my research data.

6.4 Analysis

This chapter will contain how I analysed the data that I have collected. First I will present a computer system for analysing data and how I used this software.

Computer-assisted analysis of qualitative data

The use of computers for basic content analysis became popular in the humanities from the 1960s. In the beginning it was used for tasks like counting the number of occurrences of a word in a large text. Up until the early 1980s it was mostly used for statistical use, but then the qualitative researchers began to catch up. (Silverman, 2005, p. 188) The most important reason for the late entry of qualitative researchers is that "*qualitative research is far less formulaic, often requiring an approach to computing that gives quick feedback on the results of emergent questions, involving an interactive* *cycle of thinking and innovation only really made possible with the personal computer*" (Silverman, 2005, p. 189).

Advantages

According to Silverman (2005, p. 189) the advantages of Computer-assisted Qualitative Data Analysis Software (CAQDAS) fall into four main categories: Speed, rigor, team research and sampling. The most obvious advantage for researchers is speed. When there is a large amount of word-processed qualitative data which a CAQDAS can help sort data into categories or coded segments that can be filed and restored easily at a later time. Computer software can do this at a remarkable speed which saves time and effort that can rather be used by the data analyst to think about the meaning of the data (Silverman, 2005, p. 190).

The second reason Silverman presents is rigor. This is based on that CAQDAS can help researchers demonstrate that their conclusions are based on rigorous analysis (ibid). It will add trust in the researcher because it can demonstrate that the researcher has searched for negative instances by examining all of the data rather than selecting only anecdotes supporting the researches interpretation. This can be easily done with a CAQDAS (Silverman, 2005, p. 191).

When doing research in a collaboratory manner it is important that all the researchers agree on the meaning of codes. Silverman (2005, p. 192 from Durkin) writes that CAQDAS helped the team check whether it was interpreting segments in the same way.

CAQDAS can help with sampling decisions, be these in service of representativeness or theory development (Silverman, 2005, p. 189). One example of this presented on page 193 (Silverman, 2005) is that it can help the sampling process in having a reference of all the people that have already been interviewed in the matter.

Disadvantages

Silverman (2005, p.196) presents disadvantages and limitations of CAQDAS. The first point is that a good word processor can do most of the work a CAQDAS can, but most of the features are easier to use in a CAQDAS and are less time consuming (Silverman, 2005, p. 196-197). The second issue according to Silverman (2005, p. 197) is concerning the possible imposition of a narrowly exclusive approach to analysis. The third and last disadvantage raised by Silverman is the issue with small data extracts. It might not be a point in spending time entering data from a ten-question interview , done with ten people, in order to find out how many people answered "*yes*" or "*no*" (Silverman, 2005, p. 198).

ATLAS.ti

The software I have used as my CAQDAS is ATLAS.ti. According to Atlasti.com (2007) ATLAS.ti is a ".. *powerful workbench for the qualitative analysis of large bodies of textual, graphical, audio and video data. It offers a variety of tools for accomplishing the tasks associated with any systematic approach to 'soft' data*". Silverman (2005, p. 201) writes that ATLAS was explicitly developed to enable a grounded theory approach, resulting in a program of considerable sophistication.

ATLAS calls a project "hermeneutic unit"; this simply means a tool to support interpretation. You can add as many documents to this project. Codes are used to mark a piece of text for retrieval at a later time. All the codes can be tied to a network, to create a picture of how everything is related and be used in the analysis.

My analysis

When starting my analysis I compiled all my diaries into one document. I had information from when I first started working on the project from the course INF 5750 from the fall of 2006, my diary from working on the exporter before my trip to India, my diary from the trip in India and my diary from the summer of 2007. This document became 171 pages long and contained most of my information used in this thesis. I have some other sources of information also which are defined earlier. These other sources were used when the data was unclear or I had no accurate information about a particular event.

After advice from my supervisor I downloaded ATLAS.ti and got familiar with the system. It came to be a valuable tool for my analysis of my field notes. The system has many capabilities that I have not used, but it was of great help to sort out which information was useful and which was not. My approach was mostly "*code-and-retrieve*".

I started with a list of codes which proved to be incomplete. Some codes had to be added while reading through the document because I didn't think it through thorough enough. Table 6.1 is a complete list of all the codes used to analyse the diary first time. Most of the codes represent a different part of what I did while in India. While some others, like Feedback represent a piece of text that contains feedback on my field notes. While reading through the document I coded information along the way. This is done easily with marking a piece of text and clicking on code. Some paragraphs and lines where coded in multiple codes where this was appropriate.

Name	**Description**
Date (129)	Not really used. See the explanation later.
Export and dataprovider (20)	Used to document my work on general export and the dataprovider module in RHIS 2.
Feedback (5)	Used to code feedback on my diaries and other feedback from my supervisors.
PHIS Information (60)	Used to mark information I had gotten about the PHIS network.
DSIP code (29)	Used to show code snippets from the system.
India trip (166)	This became a pretty wide code and it documented most information about my travels around India. This information was not private or directly regarding integration.
Indian culture (15)	Used to code my impressions and thoughts about Indian culture.
Integration (206)	Used for everything that concerned integration, is a family of codes with more sub-codes (see later).
Open Source (7)	Used to code information about open source.
Private (207)	Used to code everything that did not directly concern this thesis.
Settings module (44)	Used to code information about my work on the Settings module for RHIS 2.
Teaching (82)	Used to code information about my work with teaching Indian developers and the workshop held in Noida.

Theory (1)	Used to code theory and information about articles I had read.
Training (17)	Used to code information about the training sessions I attended in Ahmedabad and Surendranagar.

Table 6.1 – List of top level codes

The number in the parenthesis behind the code name show how many paragraphs or lines I coded with the given code. After I had read through the whole document once I decided to go through all the information coded with Integration once more and coded it more thoroughly to find the story and find what problems really happened. This was because the first approach was much to abstract and it was difficult to find the right information in the codes. Many of the codes used where not used or found interesting for this thesis, but was coded in case of any change in research perspective.

I first wanted to make a document with the codes date, integration and idsp-code in correct chronological order to be able to code a document once more. This was so that I could easily know on which date the different happenings occurred. I did not manage to do this with ATLAS.ti as I had hoped. I was able to get all three codes as output, but they were not in the order they occurred in the document, but sorted by code instead. I realise in retrospect that I should have coded the dates with the integration code instead. So I decided that I would instead look up the date whenever needed from the original document, and this solution worked perfectly fine.

I had decided to re-analyse all the text that had been classified as integration. I tried to take some time to create codes as sub-codes of integration, but also this time I ended up having to add three codes after I had first started coding. I wanted to be able to find the story to tell I the empirical part and see the problems and where use and effect of IOs where happening. Based on this I came up with this list in Table 6.2 which is the set of codes used to analyse the new document.

Name	**Description**
Background information (26)	Used to code background information about the Indian software.
Big problem (8)	Used to code information about problems that were big.
Changes in criteria (10)	Used to code changes in the requirements for the software.
Changes in functionality (7)	Used to code changes I made to the functionality.
Decision (1)	Used to code when decisions were decided.
Help (4)	Used to code information about me seeking help from others.
India (0)	Supposed to be used to code information that "*originated*" in India.
Intermediary Objects (80)	Used to code creation, use and effects of IOs.
Manual/FAQ (13)	Used to code information concerning my work no the manual or the FAQ for the DSIP exporter.
Organisational problem (25)	Used to code problems regarding working in global/open source organisation.
Oslo (0)	Supposed to be used to code information that "*originated*" in Oslo.
Problem solved (21)	Used to code information about one of the problems being solved.

Small Problem (30)	Used to code information about small problems that occurred.
Technical information (8)	Used to code technical information.
Work log (234)	Used to code the work I actually did on the exporter.

Table 6.2 – List of Integration codes

I wanted to find everything related to the creation, used and effect of what I thought of as Intermediary objects. The criterion is that the object was something that described my project in any way, either present solution or a later stage. The object was also meant to be shared between me and other people who needed and wanted information about the software and state of it. Based on this I used the code IO to label matching information. This was extracted from the document using ATLAS.ti, into a new one and used to create a list of what I considered the use and the effect of my IOs. This table show the time of the event, type of object, in which context it occurred and what effect it had. The table of IOs can be found in Appendix A.

I also wanted to find all the problems, when they emerged, how they emerged and when, if and how they got solved. Based on the newly coded document I wanted to find out which problem that actually occurred during this process. So I extracted these three codes: Big problem, Small problem and Organisational Problem. I analysed this document and was able to create a list of problems from both India and Norway that happened in relation to my work on the DSIP export module. I separated the problems into three different categories: issue, organisational (problem) and problem. This table of problems can be found in Appendix B.

A graphical representation of my codes displayed in a network can be seen in Figure 6.2.

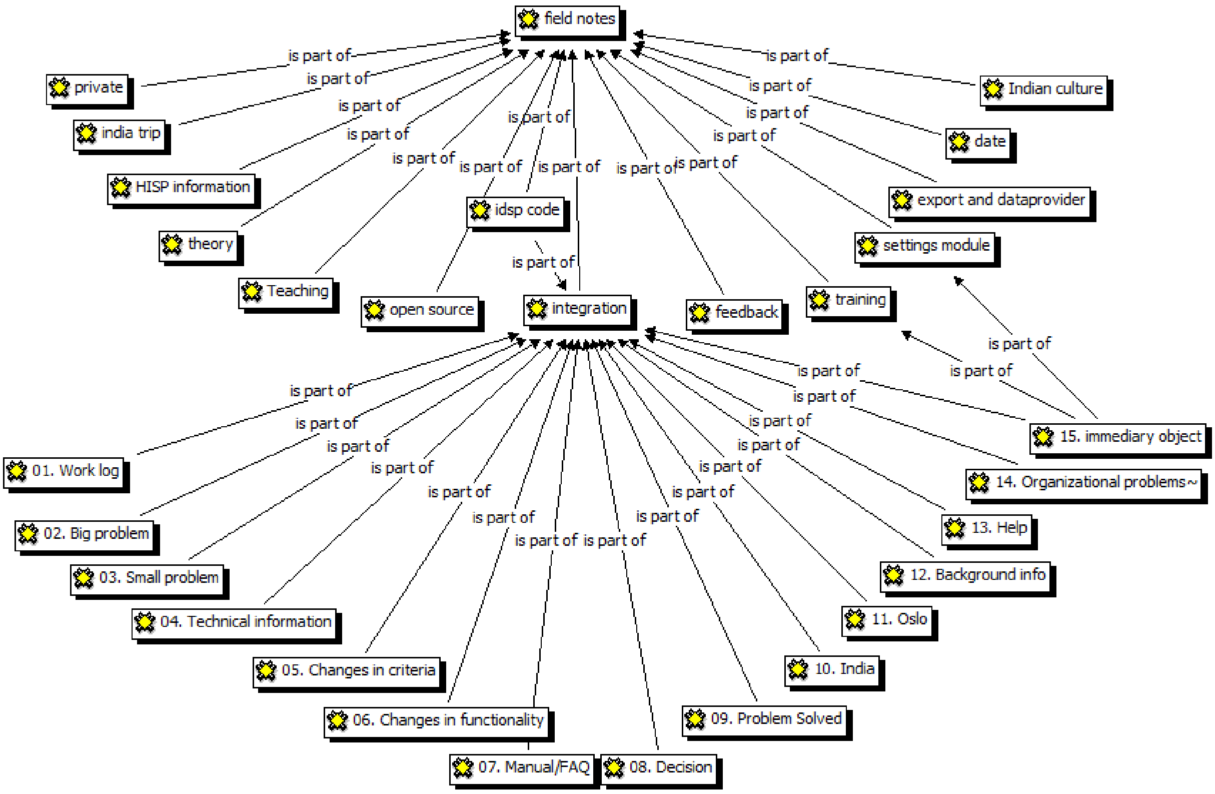


Figure 6.2 – Picture of my codes in a network

Figure 6.2 represents all my codes put in a network, with their relationships mapped. Everything is a part of my field notes, which is everything I have written down. Then there was a "high-level" coding of this whole diary which ended in 14 codes which are all a part of *field notes*. Based on these 14 codes I decided that four of them were interesting for my further work. Based on the information coded to these four types, I created a new set of codes which were all part of the first four codes. This is what is represented in Figure 6.2.

As the list present I encountered 12 problems, 16 issues and 7 organisational problems during the project. Other problems that occurred frequently in India were the sudden power failures and the lack of access to Internet the first part of the trip. The story behind these problems will be presented in the next section of the paper.

To find the pieces of the story I used ATLAS.ti to export all the quotes with the code Work log to get a document with all that I have done. I used this document together with ATLAS.ti and my two tables with information to write my empirical chapter about the DSIP exporter.

7 Empirical data

In this chapter I will present my empirical data. This includes the work done from the fall of 2006 until the summer of 2007. Most of the data will be from India where my travel partner and I were for 102 days. I will also present some of the artefacts used as the Intermediary Objects they are. Table 7.1 describes the timeline of our travelling in India. A short description of what the main focus of each city will be described below.

Place	Time	Main events
New Delhi	13.02.07 – 21.02.07	Vacation and some meetings
Ghandinagar	21.02.07 – 15.03-07	Training sessions and DSIP exporter
Bhopal	16.03.07 – 30.04.07	DSIP exporter, system settings and teaching developers
Noida	01.05.07 – 25.05.07	System settings, workshop and developing reports with Indian developer

Table 7.1 – Cities in India we visited

We first came to New Delhi during the night on the 13th of February 2007. We had the first days to get to know India and have a little bit of vacation. Our first encounter with PHIS India was when we had a small encounter with an Indian coordinator at a Reliance Internet café. He helped us set up an account at the Internet café and gave us an Indian database and the Graphical Analyser. He didn't stay long because he was just travelling through New Delhi. We also had a small meeting with another coordinator the next day. Before we left for Gujarat the first coordinator came by on another connecting flight and helped us get plane tickets to travel further.

The next place we visited was Ghandinagar which is the capitol of the state of Gujarat. We stayed here for a three week period. Here we met a big part of the PHIS India team as they were working hard to gain the support of the Gujarat government for implementation of RHIS 2.0. During the stay in Gujarat I continued work on the exporter and we participated in different training sessions conducted by PHIS India. From here we travelled by train to Bhopal, Madya Pradesh.

In Bhopal we mostly worked on RHIS 2.0 related projects (mostly DSIP export) and started educating some of the Indian developers on the RHIS 2.0 technologies. We also joined the PHIS members on some of their meetings with government officials. The stay in Bhopal lasted for one and a half month, before we travelled by train to our last location.

In Noida, Uttar Pradesh, PHIS India was setting up their main office and this is where we stayed our last part of the trip. Here we held a week long workshop where we trained a group of Indian developers in the technologies of RHIS 2.0 and helped them get started developing with the tools needed. I also continued to help their main developer integrate an Indian report module and their Graphical Analyser module into RHIS 2.0. After about three weeks in Noida we travelled back to Norway.

7.1 India

The Republic of India is the seventh largest country in the world by geographical area, but the second largest in population (India, 2008). The government type of India is federal republic. In July 2007 it was estimated that India had a population of 1.3 billion people. Hindi is the national language and is the primary tongue of 30% of the population, but there are also 21 other official languages in India. The country has problems with many diseases, among them: Hepatitis A and E, typhoid fever, dengue fever and malaria (CIA, 2008).

India's Human Development Index, which is a measurement of life expectancy, literacy, education, standard of living, and GDP per capita for countries worldwide, was 0.619 which means that India is ranked 128th in the world as of 2007 (India, 2008).

This was a very brief introduction about India, and I will now continue with the work I have done in integrating DSIP into RHIS 2.0.

7.2 DSIP integration

In this chapter I will go through my involvement in creating an exporter that was able to integrate RHIS 2.0 with DSIP. The reason for this was mentioned earlier and is mostly because PHIS would like to expand their efforts in India and this is a way to increase their chances of getting DSIP 2.0 approved for implementation.

The requirements were to create solution that would mimic the work done by DSIP. This required that we could enter the same data, and provide the same reports. Entering data is already very good implemented, it was for me to create the report mechanism. Since DSIP exports MS Excel sheets from the software and report those upwards, this was the requirement.

This part represents my main empirical findings and will describe my work from the fall of 2006 when I first started the project in the course INF 5750 until the summer of 2007 where my work on the project stopped. The project went through many different stages and there were multiple problems along the way.

Learning the technologies

This project started with me wanting to take part in the PHIS, and write my master thesis about global information systems. I got the task to implement this exporter by one of the Norwegian coordinators and I used it as my compulsory assignment the course INF 5750, which is held at UiO, in the fall of 2006. The course is an introduction course to Open Source ideology and development in the RHIS 2 project.

The course gave me a short introduction to Maven, Spring, Subversion and the RHIS 2 structure. It also taught me some of the concepts used in PHIS. The first part of the semester were used to get familiar with these tools and figuring out how I could export values from the system to a file that matched the exported file from DSIP. Many different Intermediary objects where used to learn the system; the slides from the course; the code itself; the webpage with information; and of course discussion with group teachers and other PHIS members. In the beginning of the course, the course leader and two out of three group teachers travelled to South Africa for period of time. This made learning the technologies and understanding the RHIS 2 structure much more difficult. When I was seeking help about how export worked in the system at the time, the developers said that the solution they had now was not a good one and that we should not do it the same way. So I was forced to wait with the implementation until they had decided on the Exporter interface.

In this course we all worked in group and I was in a group with other people working on different types of exporters. When multiple people try to learn new technologies, there will always be some trial-and-errors. At one time some of the people in the group decided that they had messed up their part of the project to the extent that they wanted to delete the branch and start over. At one time one of the students on my group sent out an angry mail complaining about other people in the branch while all along the problem was caused by him not using the tools properly.

After some small bumps in the beginning, getting to know the system and tools, it was time to start working on the exporter itself, starting with learning what the DSIP is and how it works.

Finding a MS Excel framework

To figure out how to mimic the export of DSIP I downloaded the software from PHIS India’s web pages. The software is a traditional Visual Basic application utilising MS Access as its backend. The DSIP system creates a report which is in the MS Excel format, which is what the integration software would have to mimic. Another master student on the group had the task of finding a free Java framework that could create MS Excel files from scratch and be able to edit them.

The framework found was a framework called POI-HSSF which belongs to The Jakarta Project that is a part of the Apache Software Foundation (jakarta.apache.org). This software was under the Apache license which meant that we could use it in our implementation for RHIS 2.0.The framework has a very easy-to-use interface with good documentation. This works in an object-oriented fashion where the workbooks, sheets and cells are created as Java objects and when you have created the whole workbook, it can be written to the file system.

My analysis of DSIP

While the other person was looking for the Excel framework I studied how the software worked and exported data. The software has three different data sets it enters data for. These are Symdromic Surveillance (Form S), Laboratory Surveillance (Form L) and Presumptive Surveillance (Form P).

The period for reporting in DSIP is one weekly, ranging from Monday to Sunday. The system only export from one week at the time. When an export is done, all three forms get exported automatically. If there is no values entered the sheet will be empty, but all the 6 sheets in the Excel file is always created.

The filename of the exported Excel file is different based on what level you export from. Table 7.2 explains how the filename is constructed by the software, depending on where in the hierarchy.

Level	Name
District	"Name of state""name of district"("Period first date").XLS
State	"Name of state"("Period first date").XLS

Table 7.2 – DSIP file name analysis

To get a grip on how the export worked I had to enter data for all the 520 different data elements in the software and document which value belonged to which data element. I entered all the data elements with numbers beginning from 1 and increasing with one each time and documented it with a screenshot. For screenshots of the actual software, see Appendix C. I was now able to see what the different information from the exported file meant, by using the unique numbers.

The MS Excel file contains six sheets; see Table 7.3 for an explanation of all the different sheets.

Sheet name	What is contains
tblfrml1	Contains the first part of Form L
tblfrml2	Contains the second part of Form L
tblfrmp1	Contains the first part of Form P
tblfrmp2	Contains the second part of Form P
tblfrms1	Contains the whole Form S
tblreportingunit	I have not found out what this does, it is possibly for containing information about the reporting units

Table 7.3 – DSIP Excel sheet explanation

The first cells in an exported sheet contained the same information. Table 7.4 explains the nature of the different elements.

Name	Value
StateId	Id number of the state
DistrictId	Id number of the district
BlockId	Id number of the block
Year1	What year the export belongs to
WorkerName	The name of the worker who entered the information
SupervisorName	The name of the workers supervisor
ReportingUnit	The name of the clinic that reports
Idno	The id number of the clinic that reports
FromDate	The first date of the period (Monday)
ToDate	The last date of the period (Sunday)

Table 7.4 – DSIP Excel sheet column analysis

The rest of the sheet contains the data elements that were exported. This is stored different in all the three forms. The first element of the second page is Idno, which is the same number that is last on the first page. The last three elements on the second page are; FromDate and ToDate which is the same as in the first page and append which is set to 0. The value 0 means that it is not yet exported. This way it is possible to export it once after it is imported from the Excel workbook. See Appendix D for an outtake of DSIP exported file and my analysis of it.

Now I had to figure out how the Excel file was organised by using the data I had entered and finding the position the exported value had in DSIP. I looked through all the sheets in the Excel file and compared values to the screenshots I had and wrote the down the corresponding coordinate. There was not much order in how they created the output, but luckily each row of data in the data entry screen was in the same order in the Excel file. The naming convention used in the exported file made no sense to me, with names like CMGF1, CMGF2 and CMLF1.

Based on the Excel files I created a file for mapping RHIS data elements into DSIP data elements. I used the Excel files to create this. Since I did not have access to any Indian database and was only going to test with dummy data elements for now I only mapped the same RHIS DataElement to all the different DSIP elements. Code example 7.1 is an excerpt from my mapping file.

```
MA00MLT=DataElement1
MA00MGT=DataElement1
MA00MLGTTotal=DataElement1
MA00FLT=DataElement1
```

Code example 7.1 - IdspDataValues.properties

This means that the value for MA00MLT in DSIP will be fetched from DataElement1 in RHIS 2.0. This was how the mapping file was at the time. Now that I was familiar with DSIP and I am beginning to understand how RHIS 2.0 works, it was time to start creating the exporter.

The exporter works!

I started coding in November of 2006 and my first commit to the repository contained all the files needed and some example code. A lot of different Intermediary objects where utilised in understanding how to code and commit my work. Examples are the code style standard defined on their webpage, information about unit testing and that every service class should have one, and the rules defined in the code repository. The system was now able to export values, but much of the functionality was still hard coded. So I added a JavaScript calendar so the user will be able to choose which period the system should export from, this was based on how the core developers told me it should look. Until now the system had only been exporting information to a text file for debugging purposes. So I decided to try to integrate the POI-HSSF framework to create an MS Excel file as the output. I used this framework to create an MS Excel workbook with all the necessary sheets and columns and inserted a hard coded value. An example of Java code from the exporter that creates a MS Excel workbook can be seen in Code example 7.2.

```
// Create the Excel workbook and sheets
HSSFWorkbook wb = new HSSFWorkbook();
HSSFSheet[] sheets = new HSSFSheet[6];
sheets[0] = wb.createSheet( "tblfrml1" );
sheets[1] = wb.createSheet( "tblfrml2" );
sheets[2] = wb.createSheet( "tblfrmp1" );
```

```
sheets[3] = wb.createSheet( "tblfrmp2" );
sheets[4] = wb.createSheet( "tblfrms1" );
sheets[5] = wb.createSheet( "tblreportingunit" );

// creates to rows for data - should be done later
for ( int i = 0; i < sheets.length; i++ )
{
      sheets[i].createRow( 0 );
      sheets[i].createRow( 1 );
}
```

Code example 7.2 - From IdspDataValueExporter.java, lines 64-76, revision 2378

Since the system still only selected data sets written in the code I had to make it able to read files. First I implemented functionality for reading the file which says which data sets the systems should export. When this was in place I created the part that read the mapping files so the system could find the values defined in this file. The Java code that read the files can be seen in Code example 7.3. The last part of the project was to add the property files to the project and to make sure that it worked.

```
// Reading idspexport.properties
FileReader leser = null;
try
{
     leser = new FileReader( "IdspDataValues.properties" );
}
catch ( FileNotFoundException e1 )
{
     // TODO Auto-generated catch block
     e1.printStackTrace();
}
BufferedReader leser2 = new BufferedReader( leser );
```

Code example 7.3 - From IdspDataValueExporter.java, lines 76-87, revision 2458

We presented the module for the class as a part of the exam in the course. This presentation was an important Intermediary Object, explaining what had to be fixed before it was ready. The object contained both technical information about the structure of the program and a high-level description of what we had accomplished. One of the slides in the presentation can be viewed in Field note 7.1. It describes what was missing at this point.

-Standard naming for datasets and elements
- Over 800 elements
- All of them need to be mapped
- Must mimic the work practise of IDPS in order to collaborate(Export data every Monday) Check the properties (mapping) files
 - Currently no validation of the mapping files exists
- Write unit tests
- File naming scheme and clean up

Field note 7.1 - From Presentation about DSIP exporter in INF 5750 page 6

So now the exporter was able to produce MS Excel files which had the same information as the DSIP files and it was able to get the values from the RHIS 2.0 database. I thought it was only a short time until this would be used in the field in India. I was wrong!

Preparing for India

According to one of the Norwegian coordinator an Indian delegation were supposed to come to Norway and work with us before we travelled to India. Instead this was postponed multiple times and we did not get to meet them before we travelled to India. Instead I worked more on tweaking the module so it could be presented when we reached India.

When I got back from the holidays there were a lot of changes in the core of the system since it was still very much under development at this time. Classes had moved between modules and some classes and classes were removed. So I had to figure out how everything was put together. Luckily I was able to work beside two of the main developers at the time and they were very helpful. This problem also showed itself when I created a new branch for the DSIP exporter only, because I did not know which other modules I would need.

Another problem that revealed itself in January was that the solution for reading the property files from within the packaged WAR file was not working anymore. This was a big problem because I had to store these file somewhere so they could be changed without compiling the code. I sent out a mail on this RHIS 2 developer mail list, but there was no reply. I also tried to go back revisions and see if you could make it work again, but every time I tried to compile a specific revision there was an error causing the system to crash when trying to build or start it.

So after some discussion with other PHIS developers I decided to put the files into the User Home catalogue where some other RHIS 2 related files are also stored. This will not be a good solution when the system is used as a server/client system, but all the places in India have so far only used it locally, because of lack of infrastructures. I also created a class that would take care of writing the property files to this location whenever they are not present. The code that changed from Code example 7.3 can be seen in Code example 7.4.

```
valueReader = new FileReader( System.getProperty( "user.home" ) + File.separator + "dhis" +
File.separator + "IdspDataValues.properties" );
```

Code example 7.4 - From IdspDataValueExporter.java, lines 56-57, revision 2781

Since the system is internationalised I had to add to put all the text into resource files and use the i18n functionality to get the text to display. All the text is stored in different files based on which language they represent. Every piece of text has a keyword which is used to retrieve it. All the text presented on the web is stored in these files. I had to put my keyword into the web module for export, which really should not be dependent on my module, but there is not much text for this module so I do not see this as a problem.

The last thing I had to do to make it work from a user interface point of view is to make the user be able to choose which Organisation unit to export data from. The RHIS already have a JavaScript gadget that takes contains the functionality for this. The user clicks on one of the organisation units and the system stores in a class which one is selected. I had to include some JavaScript files and include the organisation unit module. Now I was able to use the selection manager and fetch values for only this organisation unit.

Still I had no Indian database to test the exporter with and it only worked for the dummy set included in the in-memory database. This was now the only thing causing it not to be able to be tested by Indian developers and testers.

When I was working both at home and at UiO I encountered a problem when using Subversion. I once forgot to commit my changes from home and I was not able to get any work done at school because all my changes were still on my desktop computer at home. So this day I tried to read about India and do some reading on RHIS.

The journey was near, and at one of the last days, my travel partner and I had a meeting with one Norwegian and one Indian coordinator at UiO. Up to this point there had been relatively little interaction with other people involved in PHIS, and there had been no contact between me and PHIS India. The plan was for us to help integrate different Indian software into RHIS and to help the Indian developers get up to par on the tools and frameworks needed to be able to participate in developing software for RHIS. On the morning of February 12th we got on the plane to London and transferred from there to the British Airways flight to New Delhi.

Getting the exporter ready for presentation

During our stay in New Delhi I was not able to do much work. We had a very limited number of places where we could use our own laptop on the Internet and we were just trying to get used to living in India. We did not have any good place to work neither; our hotel room consisted of only a bed and one chair. We met a few of the PHIS members flying through the city and had a small chat, but that was it.

It was first when we came to Ghandinagar in the state of Gujarat that I got to do some work. Here we met around four people from PHIS India, which we regularly met. The situation was a bit better here; we each had a nice hotel room at around the same price as the one in New Delhi. We also had an okay Internet café 5 minutes drive away where we could use our own laptops and with a relatively stable Internet connection.

To create every export of this kind in the same way, I decided to utilise the Exporter plug-in system created for RHIS 2.0 after a tip by another developer. This follows the Intermediary object in RHIS 2.0 documentation of how to do it, and make the system easier to use and maintain. This will mean that every exporter will be selectable in a list and you use them all in the same way. When using this exporter system I no longer have to have a separate web module that needs maintenance. I copied all the information from the PHIS Confluence page when I were at an Internet cafe and tried to fix this at home. The guide had multiple errors which I fixed when I got back to the Internet cafe. I had some trouble getting it to work, because I had to actually select one or more DataElement in the actual user interface to make it work. The selected DataElements don't affect how the exporter works because I have already mapped all the needed data elements.

Normally when working in Norway I am used to having stable electricity and a broadband Internet connection wherever I am. In India this was not as common. When working offline on the RHIS project I encountered an annoying problem. When I was in my hotel room trying to build the project I got a dependency issue. My Maven repository did not have the Java framework iText 1.02b and therefore, RHIS 2.0 would not compile or build. That meant no more work done that day. The time I had access to Internet I tried to manually download the file and put it in my repository. When I got home and try to build the project, there was another module missing. Then I finally realised that I could just disable this one module as it was not necessary for my project to work. After disabling the module I did not have any more trouble of this kind.

To make the whole module more usable for other Indian export projects and easier to maintain I decided to separate the writing and reading of the property files to new Java interfaces with DSIP implementation. This was every exporter can use the same interface with their version of how to read the actual files. This made it much easier to make unit tests for each part of the DSIP exporter also.

While we were in Ghandinagar I finally got my hands on a database from Kerala that contained the DataElements created for entering DSIP data. They had 432 data elements which after some analysis I figured was all the elements for Form S and Form P. There had been upgrades in the database that was not fixed in this Kerala database because they were running an older milestone release of the system. So I could not get the system running from my hotel room. The next day when we got to the Internet cafe I sent a mail of the developers list and got a response with a link to the RHIS 2.0 documentation space. Apparently there was a page for each milestone release with the lines of MySQL code that have to be run to get the database to work.

```
alter table organisationunit rename to source;
alter table source add column sourcetype varchar(2) not null;
update source set sourcetype = 'OU';
```

Code example 7.5 - Example of MySQL code for updating database to Milestone 7

The Kerala database was exported the 5th of January, so I got the right script, but one of the lines of code didn't work. There was no "*sourcetype*" in source table, but I thought I would continue anyway. So I tried to start the system. I took a long time for the system to start, probably because it tries to alter the database to fit the new version. The first time it started I had no working organisation units, but after some tinkering in the database it seemed to work. I still got some error messages when the system booted but it was enough for me that that I could export data. I tried to find some data values I could export, but I did not find any. I asked the Indian national coordinator, and he told me that they had not started entering data yet.

So one later day at the office I was able to print a list of all the DataElements for DSIP. I used this list to create the mapping files between RHIS and DSIP. This work took several days, but went considerable quicker than I thought it would. This was mostly due to the fact that they had used the code property in RHIS 2.0 in a good way, and the fact that I did very thorough work in analysing DSIP last fall.

At the 1st of March, the Indian national coordinator called us and said he would be coming with another coordinator to the Info City Shopping area. After a while, a delegation from the PHIS team come and joined us at Dawat, a local restaurant where we used to eat. During the lunch we discussed the DSIP and RIMS exporters. RIMS stands for Routine Immunisation Monitoring System and is another Indian HIS which my travel partner attempted to integrate. I told them that I thought I could have a complete mapping table for RHIS-DSIP ready later that day. Some new requirements from India were also discussed during the lunch. This discussion is a good example of an IO that has been important for the project. It describes the current state of project and where it will go in the future. They wanted to have a separate license on the Indian specific modules because of competition from other NGOs and software companies. Ideally they wanted to store the files somewhere safer also, so the source code could not be stolen. They also eventually wanted a separate tab in the system with all this new functionality. This will be possible to do, but the exporter will have to have its own interface again instead of using the plug-in manager in advanced export. We agreed that we would meet the national coordinator later to give him the files for the DSIP exporter.

After the meeting at Dawat I continued for a few hours with mapping the elements from RHIS to DSIP. I entered values for the RHIS data set and did an export. Then I quickly did an eye-balling check to see that it worked like it should. I contacted the Indian national coordinator and told him to meet us at the hotel. He and four other PHIS Indian members came to our hotel and we tried to set up the software. As always when interacting with an Indian computer I had to be careful not to get any computer viruses. Most of the Indian computers lack service packs, security patches and also antivirus software for their OS which causes a good environment for computer viruses. The virus' travel through networks and USB sticks.

The problem at this point was that it would only work with the right version of RHIS and so far only the Kerala database I had, so everything was very fragile. There is still also the issue of getting the DSIP specific information into the exported file. We got almost everything set up, but the database import took too long and they had to leave for their flight. He told me he would set up the rest the next day. I also sent a mail to the coordinator later that day explaining what could go wrong and how he should present the exporter (Mail can be read in Appendix E). I never got any information on how the meeting went. I believe the meeting was cancelled, at least the exporter never tested at this time. This prototype was an important object displaying the Indian members the current state of the project and the current user interface. The object was created for displaying and communicating information by testing the application, sadly it was never used as it was supposed at the meeting.

Trouble with DSIP

A few days later we came to the Internet café in Info City Shopping only to find that their Internet connection did not work properly. Some web pages showed only half of the page and some would not load at all. The owner told us that there was a problem with the wireless connection to the building so there was no point in going to one of the other cafés in the building either. This was the only place we had found where we could use our own laptops online. So we went back to the hotel to try to do some work at home instead.

I still had some trouble with the organisation unit and source table in the database and sent a mail to the developer's mailing list. I got a response from a Norwegian developer later that day saying I had to copy everything from the source table to the organisation unit table. This gave me an error saying I could not add or update a child row because of a foreign key constraint. So I removed all the foreign keys on these tables and tried again; now it worked. So I started RHIS 2.0 again and the system failed to boot because of an exception. I thought I had seen this before and searched my mail archive next time I was online. The problem was fixed in revision 2995, so I downloaded the newest version of the code from the repository and built it. Building the project takes a long time, around 20 minutes on my laptop; this must be a problem for some Indian computers with far less memory and CPU power than mine. After the update the system booted properly and worked like it was supposed to.

To get a better confirmation that my exporter works like it should I decided to create a small Java program that checked if the exported values was what they were supposed to be. After tinkering with the application for a few hours I was ready to test the files. The first thing I did was test data from Form P which lead me to find three errors. Two of these errors were believed to be, after analysing the data value table, because of faults in my data entry. The last error was made by me in the mapping file, I was using the wrong code one of the elements which lead to wrong value. So I decided to check Form S also. This proved to be without any errors, at least caught with my program. The errors from Form P were still not fixed so I thought I would look more into my assumed typing errors. I realised I did not do anything wrong in data entry and after looking more closely through the data element table I found the Indians had made an error and switched two data elements that caused two errors in my export. I decided it would be easier for me to change the mapping than to get them to change their database so that it was I did. This has also been documented in the manual for the software.

Since the exporter was starting to come together I thought I would try to import the exported file into the DSIP software. The exported MS Excel file was still not complete so I had to change the date fields, clean up in the DSIP database, so there were no other rows with the same date. The import did not go well and the id number for my PHC was not entered into the DSIP database. So I decided to try to export the values I now imported. The software gave me a runtime error which was in Norwegian that said "*Ikke samsvar mellom datatyper I vilkårsutrykk*". I tried to Google the error and it means that one of the values does not match the type it is supposed to be.

So I decided to try a fresh install of DSIP and import the data again. I got the same error once again. Then I decided to look through the manual that is included with DSIP, but it contained no help on runtime errors only information about how to install the software. I also tried to look inside the database and see if I could find the scripts that handled the export functionality, but no luck there either. So I tried to play around in the MS Access database and see if I could make the export work. I realised that it created empty rows for the Form L which I do not use. So I deleted the extra L rows and entered the ID that was missing from the other tables. The export from DSIP was now working again because of the manual database cleanup.

I removed the L form completely from the mapping and I kept a backup in the mapping file for later use. It seemed to work better now, but there were still problems. If they decided to update use the L form all they need to do is update the properties file, no need to recompile or change the war file.

Next I have to do is to make sure numbers become formatted as numbers in the sheet. I also have to make the filename follow the DSIP standard.

While working on these problems I found another error in my mapping file. I had one element wrong in sum value. I never found why my application for checking for errors did not catch the error.

Realising my solution won't work

I was still optimistic about getting the export/import to work and I thought I would try to format the input into the MS Excel file as numbers instead of text. So I added a part that formatted all the numbers to doubles. I also made some static changes to the mapping file making the state, district and block ids correct. When I tried to run I got a test error from Maven. I had downloaded an update and apparently my travel buddy had made changes to our project. In the beginning I could not understand what was wrong since it was a test failure, but I eventually realised it was an error in the WebWork configuration file. I also found an error in my unit tests because I had removed Form L from my mapping. Now the system worked again, but I still had problems with the date format.

So I decided to start working on getting the periods in the right format for the DSIP export. I had to find out how I formatted the date to the right pattern needed in the Excel sheet. The format that worked was: "dd.mm.yyyy". Then I used SimpleDateFormat to format the date object in period to the correct form. I encountered a problem moving from my user interface to using the plug-in for the advanced export interface. It took a while to understand, but the advanced export passes on a list of periods instead of just one period like I did in my solution. This is reasonable because you are able to select more than one period in the advanced export interface. Now the test, date and export worked, but still not the import into DSIP.

I decided to do some more testing with imports into DSIP. It still made the rows in form L and it didn't even help to remove the creation of the sheets in the MS Excel file. The DSIP import still didn't save the id value from the MS Excel file. I decided to put the L Form back into the export since it really did not help to remove it. Every time an import fails, I have to go into the DSIP database and remove all the records that failed. Luckily the database for DSIP is not password protected, so it is easier to see the data structure and be able to do this type of cleanup. There was an Indian software called RIMS, where the database was locked and this made integration work much harder.

Since it still wasn't working I decided to try to make the file as like as the file exported from DSIP, so I decided to get the width of the columns exactly the same. I finally the last one gave the same result in Excel as the DSIP export, width of 10.57 and 79 pixels, but it still wouldn't import the data.

I thought the Excel file POS-HSSF makes had some other differences from the DSIP file, so I tried some more changes to the file. Then the property for author was set, to Your Name. I tried to change the ID number. I also tried to format some of the values in the Excel files differently. The spreadsheet was saved as an older version. I also tried more changes to the ID, but none of these efforts made any difference with importing the MS Excel file into DSIP.

The reason for the error might be because of the way MS Access creates the Excel file. Once I exported a real sheet from DSIP, I tried to change one value in the sheet and saved the file with my installed MS Excel. This made the import crash. So it seems it depends on an exact copy of sheet made from MS Access and its framework for creating reports.

On the 17th of March I sent out a mail to a group of Norwegian and Indian coordinators about the problem with DSIP export (see Appendix F). I explained what I had tried and what my plans were. Within two days the mail thread had a count of 36 and many more people had been included. A PHIS coordinator helped me examine the software and according to him the software is a Visual Basic program with an Access backend. He also said that it utilises Crystal Reports version 8 which according to him was at the time over 10 years old. Actually the software was at the time around seven years old (Crystal Enterprise). At this time my belief was that the DSIP system also was much older than it turned out to be.

By this time I had realised that this solution was not going to work. The next chapter will explain the new solution created.

Creating a new solution

From the e-mails discussing the problem, there were two solutions brought up; some other type of export from RHIS 2.0 to DSIP database and a direct connection to the DSIP database. Since I already had thought about the solution with export/import I decided to try that. I tinkered with import and export in MS Access and tried many solutions; CSV, DBF, XML and so on. I found the XML format to be the best solution for this problem. It provided a method for importing multiple tables and rows in one file, which is exactly what I needed. This would also minimise the work load on the people using the system. This choice was also supported by one of the Norwegian coordinators. Example of the new XML export can be found in Code example 7.6.

```
<?xml version=”1.0 encoding=”UTF-8”?>
<dataroot xmlns:od="urn:schemas-microsoft-com:officedata" generated="2007-03-17T20:13:11">
<tblfrmp1>
<StateId>01<StateId>
<DistrictId>03</DistrictId>
```

Code example 7.6 - From exported XML file

So the functionality was changed to export an XML file instead of MS Excel format. I had to use the same format that is used by MS Access in my exported XML file. Everything seemed to be okay, but the id was still not imported. So I tried to add a row to the reporting unit table with the same ID I used as my organisation unit. I was trying to do it with only entering and exporting data from DSIP, but the id was still not entered. It changes the column “*append*” from 0 to 1 when the data is exported and this makes the software not export the data again. Code example 7.7 is a snippet that shows new writing to the stream back to the user, this starts the XML document compatible with importing into MS Access.

```
PrintWriter writer = new PrintWriter( stream );
writer.println( "<?xml version=\"1.0\" encoding=\"UTF-8\"?>" );
```

```
writer .println( "<dataroot xmlns:od=\"urn:schemas-microsoft-com:officedata\" generated=\"2007-03-17T20:13:11\">" );
```

Code example 7.7 - From IdspDataValueExporter.java, no revision

Since there were problems with the ID in DSIP imports also, I installed the DSIP application three times (used the two old ones) and made one district, state and national. I entered data at the bottom level, exported this. I then imported this data at the state level. It didn't fail, but the id was left out. I then tried to export this data and this went okay. The export missed the id but it didn't fail. So I tried to import this file at the national level, it crashed. This software seems very sensitive to errors and it is strange if they actually use this in the field.

I thought it was time I should update the documentation page for this project. I backed up all the information that was there in case I would change something back in the future. I started to delete everything that was related to the INF 5750 course and things that were very outdated. I took a backup first of the code needed to produce the page. I also added a list of things I need to put on the page: All the errors with Excel, the new xml format, the new files created to separate functionality used by multiple exporters, how it should be used, link to the setup program, and link to the manual file.

It was proposed by multiple people that we should try to get the source code for the DSIP software, so we could see how the export/import happens. The Indian national coordinator said he would try to get a hold of it through some government people. We eventually decided that it would be nice to have the source code and probably find out how things fails, but we would probably not be able to fix the problem based on seeing the source code. If we could have changed the program in any way to make it work, it would not be possible for us to distribute the new version anyway. The problem probably lies in the way POI-HSSF creates the MS Excel files and the way DSIP/MS Access/Crystal Reports reads the file.

I decided to start working on the DSIP setup, which is a small Java program meant to help set up the mapping files for DSIP. I thought I would make it easy to add another setup class so that other exporters also will be easy to set up and can use the same software. Spring was used so that I could try to learn some of the Java frameworks better. But after many hours of trying and failing I decided to drop it because it wouldn't have made the software better in any way. The problem was getting Maven to package the Spring framework in to my executable JAR file.

I had to search the net to find out how I read input from the keyboard. It has been a few years since I learnt that. The class to use was Scanner. I spent a lot of time figuring out how the Manifest thing would work. It created at manifest but it didn't say which file was the main class. So I decided to Google for it. I first thought it was Spring that was going to make it work. But I quickly realized it was Maven that should take care of that. I found one page on the Maven site which explained it. And I got it to start, but then came a new problem. It didn't contain the Spring library.

So I had to see if I could find out what was wrong. I asked a Norwegian developer and he told me what I thought was wrong; that the Spring library file wasn't packaged and I had to include the JAR file. He pointed me to a page in the Maven documentation. The Assembly plug-in would help me. So I added the code to my Pom.xml file and tried to run it. That didn't work so I went back to the page and found that I had to run assembly:assembly to make it work. I also found code that made it happened in the normal build cycle, but a new problem arose. When I used the assembly plug-in the manifest file wasn't included. So I tried one time to copy the file between my JAR files, but then I got a Spring error. I tried to debug that for a while, it was an error regarding the beans file. I realized it was no point of using Spring, because it just made everything much more complicated than it needed to be. So I removed the Spring code and made the Java beans locally in the code instead.

Some of the functionality I could copy from the DSIP export module. I had already created functionality for creating and validating the mapping files there. I copied all the code and cleaned it up and removed some small bugs. I changed some of the methods to private so it would be easier to use and understand. I still used the interface/implementation method so that it would be easy to make implementations for other exporters in the same way.

I continued to work on the export setup tool and I made it safer for the user. I added checks on the IDs to make sure it was numbers and that the length was okay. This was in reality no problem, but as always small errors show up everywhere. This checks that the id number is the right length and that the value entered is numeric. It also checks if the user presses y(es) to save and saves the values, and validates the created file. I believe this tool works like it should now. I was thinking that someone should probably test it.

At this time this was what I thought was missing in completing the exporter:
- Some different people need to read the manual, to see that it works like it should.
- Missing one picture in the DSIP exporter manual
- Do more checks and tweaks on the setup tool
- Have some people test the setup tool
- Fix up the DSIP module with the latest changes; removing the creator and making a new test for the new XML format.

Now that I had the setup tool working I removed the file creator and threw an exception instead if the files were not present. I also added the validator to the class to the DSIP export module. I made new unit tests for the exporter that checked some lines in the exported file. It checked that the values exported were correct. Now I decided to send a mail to the developers list (See appendix G) describing what was done and if there were any comments.

Once again there were database changes and I had to fix the Kerala database up to date. I did a backup of my database and tried to use the script provided by the documentation for RHIS 2.0. On the second line of the script the fix failed. The first that was tested was to delete all the foreign keys which did not work, so all the indexes in the database were removed, which corrected the error. The script was started again and it ran for 10 minutes, it produced one more error, but I did not think it was an important error. So RHIS 2.0 was started again, it took a long time and produced multiple errors, but the web server actually started.

I edited the DSIP confluence page with some information on how to add the DSIP exporter to the war file. A sample of this documentation can be seen in Field note 7.2.

All you have to do is add a dependency to the module (see below) in you pom.xml in the dhis-web-portal folder. You have to have access to the net while you build the .war file and the needed archive will be downloaded and included. The exporter will automatically add itself to the advanced export page.

pom.xml code

```
<dependency>
<groupId>org.hisp.dhis</groupId>
<artifactId>dhis-service-export-idsp</artifactId>
<version>${version}</version>
</dependency>
```

Field note 7.2 - From the documentation of DSIP export, see Appendix K for link

Creating a better solution

I got a new mail from the Norwegian coordinator where he said the procedure was a bit too difficult as it was now, this was in response to mail in Appendix G. This started a discussion which was an important IO for the project. He would like to be able to not have to enter the DSIP variables into a DOS window and shorten the whole procedure. I wrote back with a list of possible solutions on the variable problem which is represented in E-mail 7.1.

- *Add elements to the dataset with this information (still have to be entered every time)*
- *Add a small dataset (with these 7 elements) where this information could be stored (entered once)*
- *Add all of the information from the DSIP database to a big dataset (lot of mapping and entering has to be done). They might not use the exact same name for the PHC in both RHIS and DSIP i.e.*

E-mail 7.1 - From me, to RHIS developer mailing list, subject: [Dhis-dev] IDSP export setup-tool and manual, date: 22.03.2007

Of these choices I was mostly leaning towards number two. I also suggested that I might be able to connect directly to the MS Access database and export the data via SQL queries. While I waited for a reply the electricity went out, so we went out to eat instead. When we got back the electricity was luckily working again. There was a mail response from a Norwegian developer with more information on connecting to the DSIP database via Java Database Connectivity (JDBC). He provided link to a website with some information. I had done this earlier and when we tried this solution we had to register the database as an ODBC source in Windows. I decided to try this approach, and wrote a mail back. I searched some on the Internet and it seemed as I had to create at ODBC source before it can be used. I started the ODBC setup, odbcad32.exe, which is located in the system32 folder. I created a source called MOH to the MS Access database file (MOH.mdb). I tried different formats on the location of the database. The different trials to get this to work are displayed in Code example 7.8.

```
// Not working
String location = “Jdbc:odbc://MOH”;
// Not working
String location = “Jdbc:odbc://MOH.mdb”;
// Not working
String location = “Jdbc:odbc://localhost/MOH”;
// Working
String location = “jdbc:odbc:MOH”;
```

Code example 7.8 - Example code

Now the system changed from creating an XML output to connecting to a database and inserting values into the actual DSIP database. Creating the query itself was not a problem, but the date format in Access was hard to understand. It was also hard to know what was actually wrong, because I got no error explaining what was wrong when the insert-query failed. Finally I found the right format and created the Java code to present the date in the correct format:

```
DateFormat df = new SimpleDateFormat( "MM/dd/yyyy" );
```

Code example 7.9 - From IdspDataValueExporter.java, no revision

While I was in the process of fixing the date format I got a mail from a Vietnamese developer giving me a tip for how to solve the ODBC problem. The description of how to solve this problem is an important IO for the way the project evolved. It is possible to connect directly to Access database file on the file system and use it as an ODBC source, which very much simplifies the procedure. He sent with some code that would make it. The code can be seen in Code example 7.10.

```
String filename = "D:\\studentlist.mdb"; // The absolute path of MS Access file.
Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
String database = "jdbc:odbc:Driver={Microsoft Access Driver (*.mdb)};DBQ=" + filename;
Connection con = DriverManager.getConnection(database, "", "");
```

Code example 7.10 - From IdspDataValueExporter.java, lines 76-87, no revision

I decided to implement this instead, which is much more user friendly than having to register the file as an ODBC source first. There was another small issue arising with this solution. The system has to know where the Access file is located. So this was added to the mapping file with the other DSIP variables. This will probably be located in the same place on most machines, and if it is it can be automated with that location.

While working on converting the exporter from using XML to connecting to the Access database I found a bug on the date. The date structure for DSIP is from Monday to Friday, while the weekly period in RHIS 2.0 stretches from Monday to Sunday. To make sure the conversion of dates function properly over months and years I decided to use the Calendar class for modification of the Date objects. I sent another mail on the DSIP export setup-tool and manual thread on got a good response on how to use it. I ended up with the code in Code example 7.11 to get the date two days before.

```
Calendar c = Calendar.getInstance();
c.setTime( toDate );
if(c.get( Calendar.DAY_OF_YEAR ) == 1 || c.get( Calendar.DAY_OF_YEAR ) == 2)
{
c.roll( Calendar.YEAR, -1 );
}
c.roll( Calendar.DAY_OF_YEAR, -2 );
```

Code example 7.11 - From IdspDataValueExporter.java, lines 76-87, no revision

I thought that I should improve the setup solution, so I decided to try what a Norwegian developer commented earlier, but not implementing the full solution. His proposal can be read in E-mail 7.2.

> - Add a small dataset (with these 7 elements) where this information
> could be stored (entered once)

I crazy idea somewhat unrelated to this thread. Data elements and data sets are used to register data about something... What if we used datasets and elements for user data, settings and other things in the RHIS 2 application itself. It would be reuse of existing functionality rather than create something specific. We'd need to be able to separate between "system" datasets and real datasets though.

E-mail 7.2 - From Norwegian developer, to RHIS developer mailing list, subject: [Dhis-dev] IDSP export setup-tool and manual, date: 22.03.2007

So I created four DataElements to the database: DSIP Name, DSIP Worker, DSIP Id, and DSIP Supervisor. When trying to create a DataSet the system created an Exception. Apparently there was a problem with the database. There was no Period with id 0, so I changed the "*two yearly*" Period id to 0 instead of 7, and it worked. With this solution I also lowered the number of needed values from 7 to 4. The three ids that are needed are now subtracted from the long id which is explained in Table 7.5.

An example of an DSIP id 3501020106001 explained	
35	state id (Andaman & Nicobar Islands state)
01	district id (Andamans district)
02	block id (Bardez block)
01	sector code (01 means Government)
06	category code (06 means PHC)
001	id of the Organisation Unit

Table 7.5 – The DSIP Id explained

I implemented the part that read properties from Data Sets instead of from the property files, which now would function only as mapping between the different Data Elements. I figured that I would also need to test the new solution. A bug arose when trying to add Data Elements to the new Data Set in the test. The documentation said the addDataElements() method would accept classes that implemented the Collection interface, which both ArrayList and Vector does. But it returned a ClassCastException, so I decided to look into the Java API which describes all classes, interfaces and methods of the entire Java library. There I found the HashSet class which worked properly.

The path to the local DSIP database was still defined in the property file, so I created a fifth element in the DSIP Setup Data Set. It was now possible to enter all the information needed into the Data Entry screen instead of using the mapping files or the setup tool. I also updated the test to this functionality and tested in a running web server. Since the system connected to the database and put the values directly into the database I had to figure out what to do with the file sent from the server to the user when doing export. This file now tells the user if the export was a success or not. If there is an exception, the text explains to the user what the fault is and what the user can do.

Now it was time to fix the manual/FAQ with the latest changes. I updated it again with pictures and explanation of how to set it up and how to do the export. We still weren't added to the Indian mailing list so I sent a mail to a selected number of Indian and Norwegian PHIS members. The full mail can be found in Appendix H. The mail also got forwarded to the person who is I charge of RHIS in Kerala. This would be the first state to try the DSIP export when it is done.

I got a response from an Indian coordinator with some suggestions to how to store the path for the Access database, but nothing special came out of it. The path could be different in each location and some other information about the DSIP installation also has to be inserted into RHIS 2 in one way or another. The next reply to this mail came from a Norwegian coordinator commenting on how the property reading and total solution were now. The e-mail can be seen in E-mail 7.3.

> *> I have changed it so it can find these values from a data set within > RHIS. This is maybe not a "right" way to do it, since the data sets > aren't supposed to keep this sort of data (am I right?).*
>
> *This is in some ways nice, and it is good to see it working now for the short run. But it is definitely not the right way to do it in the longer run. These values should not be stored in the database, but probably in a property or XML file, which I guess is what you did with the first setup program? Sorry for not seeing this clearly earlier, but we should probably return to that solution.*

E-mail 7.3 - From a Norwegian developer, to a list of coordinators and PHIS people, subject: IDSP exporter anno 26. March, date: 26.03.2007

So I replied that I would go back to the old solution and explained why I created the setup application in the first place with E-mail 7.4.

> *The reasons I created the setup tool was:*
> *- It might be needed in other exporters (RIMMS, ect)*
> *- It makes sure the length etc is okay and that the user does not corrupt the file.*
> *- To setup the files from the first time. The files will not be created until the setup tool or the exporter is run (Depending on which solution).*

E-mail 7.4 - From me, to a list of coordinators and PHIS people, subject: IDSP exporter anno 26. March, date: 27.03.2007

On the basis of this e-mail I converted the solution back to the way of reading the DSIP properties from the mapping file instead of reading them from RHIS Data Sets. This was no problem at all since I kept a backup of the last code. I tested the solution again and it worked like it did before. I also did a good visual eyeballing test of the database after testing the new solution to make sure it exported the correct values.

When I downloaded the new revision from the repository, the new User Role system had been implemented. This caused me to not be able to log into the export module with my Kerala database. The User Role system was created so that the administrators can control what each user has access to do. I tried searching the documentation for a solution, but found none. So I sent a mail to developers list asking for help on how to get it working again and I got a reply from a Norwegian developer. The reply can be seen in E-mail 7.5.

```
01 Stian Strandli wrote:
02 > Hey folks!
03 >
04 > I just started RHIS with my Kerela database. I have a revision from
05 > yesterday. Now my user hisp has no access to do anything. How do I
06 > give my user full access? Is there a page for this on the confluence
07 > (i tried searching)?
08 You'll have to do the initial step manually for now, something like this:
09
10 /* create new role*/
11 insert into userrole vaues ( NULL, 'superuser');
12 /* add module access to the user module*/
13 insert into userroleauthorities values ( [userroleid],
14 'M_dhis-web-maintainance-user' );
15 /* Assign a current user to the created user role */
16 insert into userrolemembers values ( [userroleid], [existing userid] );
17
18 Now start the user web module and you should be able to create new roles,
19 add authorities, and assign users to the roles.
```

E-mail 7.5 - From me, to RHIS developer mailing list, subject: [Dhis-Dev] User role problem, date: 27.03.2007

I tried this for a long time, but I did not get it to work. After a while I decided to write a mail back to the list, and then I noticed that on line 14 there was a spelling mistake which caused the error. I fixed the database and it worked. I also wrote a mail back explaining the problem, so that other people using this as a solution will be aware of it.

Since we finally got added to the PHIS India mailing list I decided to fix up the manual with all the latest changes so I could get some feedback on how the system was functioning now. I decided to split the manual into to two pieces; one for users and one for technical personnel implementing the system. I also added a guide on how to edit the property files manually and a part about how to add the project to the build cycle. The setup tool was also updated so that it only needed 5 elements as input.

I was really optimistic on getting some good feedback from Indians on their mailing list. The hope was that they would be able to test some parts of it and see if anything goes wrong. I constructed a long mail and put in link to the documentation files which I had uploaded to a server. The whole mail can be found in Appendix I.

The first response I got (E-mail 7.6) was from the Indian national coordinator which congratulated me on the accomplishment.

```
01 Hi Stian,
02 Thank you verymuch for the great work. I would really like to test it on my laptop first before
03 taking it to the field.
04 Can you guide me how to get the war file, is it available along with M7?
```

E-mail 7.6 - From Indian national coordinator, to PHIS India mailing list, subject: IDSP Export module, date: 28.03.2007

The second e-mail (E-mail 7.7) was from another Indian coordinator.

01 congratulations stian
02 much appreciated

E-mail 7.7 - From another Indian national coordinator, to PHIS India mailing list, subject: IDSP Export module, date: 28.03.2007

It felt nice to be appreciated, but these were not the answers I was hoping for. I answered the first coordinator with E-mail 7.8.

01 Hello Indian National Coordinator :)
02
03 This is really what is left. Since the source code for this is
04 supposed to not be together with the public code this have to be done
05 in addition. There is a small guide on how to do it inside the
06 documents, but it is not complete. You will have to download these
07 files:
08 http://student.hiak.no/~stian/setup-tool.zip <http://student.hiak.no/%7Estian/setup-tool.zip>
09 http://student.hiak.no/~stian/repository.zip <http://student.hiak.no/%7Estian/repository.zip>
10
11 The first file contains the jar file you need. This is the DSIP
12 exporter. You will have to unzip the folder inside and put it in you
13 user.home/.m2 folder. Then the structure will be:
14 c:\D&S\username\.m2\repository\org\hisp\dhis\dhis-service-export-idsp\2.0-M7-SNAPSHOT\jar
15 file her
16
17 The you add the dependency in web-portal pom.xml file like the guide
18 says and it will be a part of your build. The file will hopefully be
19 located at the PHIS repository server for the release so you dont have
20 to download it yourself.
21
22 The other file contains the setup file and a bat file to start it. Run
23 this by double clicking the bat file and follow the guide. You will
24 have to have DSIP installed and find the FULL path to the database
25 file. Once you have set this up you are set to try it. You will have
26 to have a working Kerela database though. Insert some data to the two
27 DSIP sets and try to export data following the guide. Hopefully the
28 file will report that the data is inserted right.
29
30 Please respond if it doesn't work or something us unclear.
31
32 Stian

E-mail 7.8 - From me, to PHIS India mailing list, subject: DSIP Export module, date: 28.03.2007

It was pleasing that he was going to test it and see how it worked, especially since no one else responded to this object at all. I didn't get any feedback on this until the 31st of March when I got this reply in another mail from the national coordinator, displayed in E-mail 7.9.

01 Issues:
02 Testing of DSIP:
03 Yes I promised to test it, but a bit busy writing some document for NN last two days and I
04 promise that entire sunday is for testing only DSIP.

E-mail 7.9 - From Indian national coordinator, to 13 different people, subject: no subject (about workshop), date: 31.03.2007

We joined an Indian coordinator who was visiting in Bhopal on a meeting with the Commissioner of family welfare in Madhya Pradesh. They discussed the situation here in Bhopal and since the Indian coordinator said DSIP integration was in place and RIMS (which is another Indian HIS software) integration was on the way, the commissioner asked me about the integration. I was a bit startled, but I managed to describe it a bit, being very modest. The coordinator later told me that I did OK, but that I should make it sound like it was really hard to accomplish. At a later time in Bhopal the national coordinator came to visit us, but he did not test the exporter the time he was there. It didn't seem like he had tested it on his own either.

Before we left for out last stop in Noida, we prepared our lectures for the workshop. I made a presentation about the DSIP exporter that I planned to present at the workshop. At the workshop there was no time for the presentation. I had a session with the coordinator for Kerala and presented everything for him the last day everyone was present. I went through the whole flow of building the system, configuring it and doing an actual export. He seemed to understand how it worked. He commented that he had also followed some of the discussions on e-mail from earlier. On the 16th of May when he had gotten back to Kerala I sent him an e-mail with the documentation and everything needed to set it up and test the exporter. This e-mail is seen in E-mail 7.10.

Hello Kerala coordinator!

Congratulation with the engagement I saw some nice pictures on NNs computer!

Here is some of the documents about the exporter. I hope you will be able to look at it and give me some feedback on how it is described.

You will find the exporter in the repository: scm/branches/dhis-service-importexport-idsp and you will have to run mvn clean install in this directory to compile it. Then you will have to add one dependency in pom.xml in web-portal before you build. Most of it should be explained in the manual file.

Files:
http://student.hiak.no/~stian/Manual.zip
http://student.hiak.no/~stian/setup-tool.zip

Thank you!
Good luck!
Stian

E-mail 7.10 - From me, to coordinator in Kerala, subject: IDSP stuff, date: 16.05.2007

I got a reply the next day saying that he would look at it the next day because he had been busy with other tasks. I never got any more answers from him. As of now the exporter has not been used in India and to my knowledge never tested properly either.

What could be the reason(s) for not implementing the module?

After I presented the module for the coordinator for the state of Kerala, I sent him a mail about how to test and implement the software, but nothing more happened. Based on what I experienced in India there could be multiple reasons that the exporter was never tested or implemented:

- Not enough "*need*" or support for it
- To difficult to implement
- To difficult to use
- Not enough time and competence to test or implement it

The need for this software was brought up in the fall of 2006 and was ready for testing by April 2007. When I was in India, the demand for this functionality was variable. In one period it was important to get the functionality ready for a meeting, but it was never used. I was told multiple times that it would be tested, but it in my experience it never was. I think the lack of actual need and support for it somewhat caused the Indian team to neglect it after it was ready for testing in the actual state. Management support is one the important issues for having a successful project according to Warne & Hart (p. 191, 1996). There was also no response on the e-mails I sent to Indian people about the actual software, which means that I was not able to include the actual users into the development, which according to Warne & Hart (ibid) is another important factor in successful development.

Because of the need for multiple data that is DSIP specific, i.e. the id for every unit that uses the program, it was harder to implement because this information has to be stored somewhere. The choice I made for this was to store it together with the mapping between DSIP and RHIS. When using the small Java setup program I created it is possible to enter only four values, but by editing this file manually there are many more lines of code to edit. This information has to be stored on every PHC, CHC, sub-centre etc. they implement RHIS 2.0 with DSIP exporter into, which causes more work for the facilitator. There is already a complicated procedure implementing RHIS 2.0, since there is (yet) no complete installer created for RHIS 2.0. Based on the facilitators I have seen there might a bit difficult to follow my instructions in English, but if that was their main problem they should be able to translate it to the needed language.

From a user-friendly perspective on why the software was not tested or implemented we can use the technology acceptance model (Boddy et al, 2005, p. 197) which models why a piece of software is accepted/used or not. The model defines two different variables: perceived usefulness (PU) and perceived ease of use (PEU). PU is defined as "*the degree to which a person believes that using a particular system should enhance his or her job performance*" while PEU is "*the degree to which a person believes that using a system would be free of effort*" (Davies, 1989, p. 320). Davies's studies showed that the relationship between PU and actual usage was higher than the relationship between PEU and actual usage. If I were to apply this model to me project, it would seem that the user would gain nothing in using my solution. It will not improve the persons work in any way, since the worker is still needed to use the old software to do the actual report and he or she will also have to learn another piece of software (RHIS 2) for data entry.

This might be a sub-reason for point one and four. The way it functions now is better than the XML approach, but it is more difficult than the functionality of the MS Excel export. With the working solution, the user still have to use DSIP to complete his/hers reporting. So this solution does not really improve the work a person reporting with DSIP.

From what I saw in India, there is always something more important to do, and there is always some type of trouble they have to fix. I believe this is a possible reason for not starting to use it, even though I had written as much documentation as I could, I have presented the system for multiple people in India. The person, who was supposed to try it, seemed to me to be competent enough to handle a test implementation and if there were trouble, he should have been able to report them to me or someone else. I could have of course have been more persistent and asked more for testing, but I think I did my part and that it was up to them to give me feedback.

Factors such as top management support, user involvement and the use of effective development methodologies have often been quoted as essential for success (Warne & Hart, p. 191, 1996).

Now that I have presented my main work of trying to develop an integration module between DSIP and RHIS 2.0, I will present some of the other relevant data from India.

7.3 My other work in India

In this chapter I will go through some of the other related work I did while visiting India. The other related work I did in India can be split into two parts. I created a new module where the user of the system can change the colours of the system. In this development project I also utilised different IOs in creating, designing and completing the module in collaboration with people within the PHIS network.

The other important work conducted in India was in training Indian developers in the technologies involved in RHIS 2. This part of the empirical chapter is a bit different and may not fit completely with the main line of integration, but it has some interesting use of IOs so I decided to include it in the thesis.

Creating the System Settings module

During my stay in India, after the work on the export of Indian systems stopped I was assigned to develop the possibility of customising the web interface. This task was one of the issues they wanted to include into the release of Milestone 8.

I started to think about what parts of the interface could be customized and decided to put together a mail about this and get some input on how and what to do. The entire mail can be found in Appendix L, here is an outtake of the mail describing what I think could be changed, how it can be changed and how to store it. I included a part of this e-mail in E-mail 7.11.

06 There is many ways to do this:
07 - Let every user define what he wants through its user object
08 - Let there be one default style and users can have their own
09 - Let supervisor (i.e.) define how it would look for everyone
10
11 How to store the changes:
12 - Define in a new .css file which can be included at the end of all

13 the tweaks and hacks
14 - Define one file for each user based on its name? Where should these
15 files be stored?
16 - Define the css in a String or some kind of object in the User class
17 and add a new <style> and run through it after the tweaks and hacks
18
19 What can be changed:
20 - Background colour (same everywhere? Not black?)
21 - Font colour (same everywhere? what about links?)
22 - Background picture (on first welcome page or soft back on all?)
23 - Background colour in main window (Maybe not all colours?)
24 - Text on the top (loaded from where?)
25 - Anything else?

E-mail 7.11 - From Appendix L, lines 06-25

The mail thread ended up with a total of 23 e-mail replies where two were private e-mails. The first suggestion was from an earlier developer suggesting a framework for this functionality, which was a bit much for this type of functionality. The next reply was from a Norwegian developer. He commented on most of my questions and defined most of how the module was going to be created. In E-mail 7.12 you can read some of the things he suggested.

01 I think there should be a global settings page, typically only accessible
02 to administrators, where one can set this for one particular installation.
03 I don't really see the point of every user being able to tweak his or her
04 view of the system.
05
06 ...
07
08 I'm thinking style changes will be rare (based on my scenario above), and
09 as such the file shouldn't be generated every time. As such I'd lean
10 toward generating a CSS file each time the configuration is changed,
11 storing it in some useful place, then serving that file on subsequent
12 requests. This way the file can also be cached by the client.
13
14 ...
15
16 I'd suggest
17 developing two modules for this, one for user-settings, moving all the
18 stuff on the left side menu from the front page, and one for global
19 settings. A submenu in one or both of these modules could contain style
20 settings.

E-mail 7.12 - From a Norwegian developer, to RHIS developer mailing list, subject: [Dhis-dev] User customization, date: 09.04.2007

So I created a new module based on an earlier maintenance module and added the module to the sandbox part of the subversion repository. This module should really have been created in the incubator folder instead, because it was a new module. Once again I had to edit the database and give me access to the module because of the user role functionality. The first thing to do was to create a menu with two options: user settings and global settings. The user settings were already implemented on the front page of the application, and all I had to do is move the files and definitions into my module. I moved everything and tested the functionality.

When the functionality was somehow defined, I decided to send another mail asking some questions that was more specific on of to implement it. There are always a lot of possibilities on where to put modules and how to do the implementation and there are always different opinions on what is best. I asked how the database should be structured and where to put the implementation. E-mail 7.13 is part of the reply I got from a Norwegian developer.

> *Have a look at UserSetting.java in the API and UserSetting.hbm.xml in dhis-service-user-hibernate. The primary key is the name of the setting (combined with the user), and the value is something serialisable.*
>
> *...*
>
> *I guess you could follow up on the dhis-support-system module I made. Make a org.hisp.dhis.system package in the API and implement it somewhere, maybe in the dhis-support-system module. Create a SystemSetting class or so.*

E-mail 7.13 - From a Norwegian developer, to RHIS developer mailing list, subject: [Dhis-dev] User customization, date: 11.04.2007

Based on the reply I decided to create the database with an ID created by the database, a name and a Binary Large Object (BLOB) which stores a serialisable value which is the setting itself. The setting can be a String or any other Java object that is serialisable. The class SystemSetting object is what is stored in the database.

I did not find any way of creating a style sheet file so I decided to change the CSS specification after all the original style sheets were loaded. This was done in the Velocity files that render the HTML code. The next problem was to get the values stored in and loaded from the database.

I implemented the code for storing the SystemSetting object, but it wasn’t persisted in the database, I could not find anything wrong with my code, so I instant messaged a Norwegian developer and asked him what could be wrong. He explained to me how the transaction management was created and used. Transactions was completely hidden for the code storing the object and was controlled with WebWork Interceptors which started and closed both read and write transactions. Without the use of these Interceptors the Hibernate session was never flushed and nothing was written to the database. The XML code that fixed my problem is represented in Code example 7.12.

```
<bean id="org.hisp.dhis.options.SystemSettingStore"
  class="org.hisp.dhis.options.setting.HibernateSystemSettingStore">
  <property name="sessionManager"
    ref="org.hisp.dhis.hibernate.HibernateSessionManager"/>
</bean>
<bean class="org.springframework.aop.support.RegexpMethodPointcutAdvisor">
  <property name="advice" ref="readOnlyTransactionInterceptor"/>
  <property name="patterns">
    <list>
      <value>.*\SystemSettingStore\.get.*</value>
    </list>
  </property>
</bean>

<bean class="org.springframework.aop.support.RegexpMethodPointcutAdvisor">
  <property name="advice" ref="readWriteTransactionInterceptor"/>
  <property name="patterns">
    <list>
      <value>.*\SystemSettingStore\.add.*</value>
      <value>.*\SystemSettingStore\.del.*</value>
    </list>
  </property>
</bean>
```

Code example 7.12 - From beans.xml, lines 33-37 & 45-64, revision 3291

Now that the storing worked it was time to be able to show the stored data. To be able to implement this I had to read a few chapters about WebWork so that I knew how Interceptors worked. I first created an Interceptor that fetched the values from the database for every request to the server. This worked, but it was a bad solution. I also tried to store the values in the session object connected to each user that is logged in, but it did not work like I hoped it would. The last, best and most complete solution was to create a manager class that could manage all the values stored in the database. This implementation proved to be a good solution. The values are automatically fetched from the persistence layer into a dictionary when the manager class is first created, as seen in Code example 7.13.

```
  private SystemSettingStore systemSettingStore;

  public void setSystemSettingStore( SystemSettingStore systemSettingStore )
  {
    this.systemSettingStore = systemSettingStore;
    this.currentSettings = this.systemSettingStore.getAllSystemSettings();
  }
```

Code example 7.13 - From DefaultSystemSettingManager.java, lines 62-68, revision 3291

There was also a discussion about having the possibility of showing a picture on the front page. We had a discussion over e-mail on the possibilities of storing pictures. One of the suggestions was to upload the picture and store it somewhere on the web server where Java could move the file and display it on the web. The second suggestion was to store the picture in the database. The third and easiest selection was to add an image tag to the main page and let the individual state or country who build the WAR file insert a picture manually. This last solution was chosen.

After much tinkering the module was almost finished, and I had some questions before moving it into the trunk directory and making it a part of RHIS 2.0. I sent a mail asking if the database will be updated automatically and how the order of the menu elements is defined. My new module was listed near the right end of the menu and I thought it would be better to have the settings tab more to the left. The order of the elements where answered in E-mail 7.14.

01 The order of the modules is managed by the
02
03 org.hisp.dhis.webportal.module.ConfigurableModuleComparator in dhis-web-commons.
04
05 The order can be explicitly defined through its beans.xml configuration (in dhis-web-commons).
06 Modules not defined here are listed alphabetically after the defined ones.

E-mail 7.14 - From a Norwegian developer, to RHIS developer mailing list, subject: Customization module summary, date: 10.05.2007

With this knowledge I fixed the order of the elements. I was told that the new table would automatically be inserted in the database by Hibernate, but since I posted my Hibernate database definition for the new table there was a discussion about what to use as primary key. I had originally used the name as the key, because I did not need any integer to be referenced in other tables. Every System Setting would preferably also be unique in name. Code example 7.15 was my Hibernate definition.

```
<class name="org.hisp.dhis.options.SystemSetting" table="systemsetting">


  <id name="name" column="name">
    <generator class="assigned" />
  </id>


  <property name="value">
    <column name="value"/>
  </property>


</class>
```

Code example 7.15- From SystemSetting.hbml.xml, lines 7-17, revision 3291

One developer thought we should use an incremented integer as the id and I agreed since all the other classes except DataValue uses an integer as the primary key. This was also defined in the database API in the RHIS 2.0 documentation (Database API) which is a good example of an Intermediary object describing how to develop software within RHIS 2.0. The mapping file and the Java class were altered to match this change. The Hibernate configuration was Code example 7.16 this after the change.

```
<class name="org.hisp.dhis.options.SystemSetting" table="systemsetting">


  <id name="id" column="systemsettingid">
    <generator class="native" />
  </id>

        <property name="name">
                <column name="name" not-null="true" unique="true"/>
        </property>

  <property name="value">
     <column name="value"/>
  </property>

 </class>
```

Code example 7.16 - From SystemSetting.hbml.xml, lines 7-17, revision 3297

Before committing the final changes to the repository, I also had to change the pop-up windows to include the new title and colours. Norwegian and English translations were also added, the rest of the translations will be added by other people.

This functionality was first somewhat replaced by a theme functionality, but was later replaced by a whole new system for using CSS files to define different styles and a whole new look for the entire system.

Training Indian developers

The second goal of our trip to India was to teach some of the Indian developers the skills needed to participate in the development of RHIS 2.0. This chapter is not very related to the actual integration I have done, but it explains some of the work we did with regard to teaching the Indian developers the skills we thought they needed. It also contains some interesting use of Intermediary Objects.

It is especially important that PHIS India can be able to develop their own specific software themselves. This means integrating what they have already created with other tools and frameworks. It also means that they should be able to develop integration with other Indian software. PHIS India have hired people to do development, but only one developer have been doing any real development, and the development was mostly done on India specific software which was not using the same frameworks and methods as the RHIS 2.0 system used. In this chapter I will give a small account for what I did in with regard to training Indian developers.

While we were in Gujarat the first big discussion about a workshop was initiated. At a lunch meeting with many of the PHIS India coordinators we decided to plan a workshop where they would gather a big number of Indian developers and coordinators for us to train. The initial plan was to hold it in Kerala in the end of April, but was later postponed to May and moved from Kerala to the new office in Noida.

Before the workshop we both worked with two developers while we were in Bhopal. The two developers consisted of their main developer and a newly hired developer which was being trained by their main developer. Their main developer gave us an introduction to the Indian module Graphical Analyser and to their report system. We then tried to change the code for the reports from

Java Server Pages over to using the RHIS 2.0 framework stack instead. We spent a bit over two days doing this while getting some support from the Indian main developer. After this, my travel partner started doing some exercises with the Indian developers to prepare them for the workshop; this was also a good way for us to see if what we had planned would work. The tasks included using Maven, Subversion and Spring. The first task was to set up the right structure for a Maven project, creating two Java classes using Spring to inject dependency between the classes and adding and updating the actual code to a Subversion repository. These were the tasks we were introduced to the system with and we thought they were a good way to introduce them to some of the most important tools in the RHIS 2.0 world.

A big problem while working with the Indian developers in Bhopal was that their main developer had to do a lot of other tasks that was needed for the organisation. There was especially a lot of work with fixing the different databases for the various states. Field note 7.3 is an excerpt from my field notes with my thoughts on this issue at the time.

I feel they are using their resources in a strange manner. They are using their best programmers to solve all these database issues that came from M7. It seems they have spent some weeks on that now. It must be wiser to have someone else do this and let these guys do programming instead. They have other people with CS degrees hired; it shouldn't be a big problem to make them learn how the database works. Hopefully more people will have more knowledge about RHIS after we are done here.

Field note 7.3 - From my personal field notes, date: 11.04.2007

During our stay in Bhopal there was a lot of discussion on e-mail about how we would undertake the workshop itself. Many of the coordinators where hoping to get the developers to develop something for the actual system as a task, but my travel partner and me decided that would be too difficult. From what we had seen earlier, we knew the level of Java competence was not very high. Based on this we decided to have simple tasks, but still using some of the tools needed in RHIS 2.0 development.

We created slides and presentations of a range of topics: Maven, Hibernate, structure of RHIS 2.0, using the data provider class, the DSIP exporter and more. To be able to create the slides and present some of the frameworks I also had to read a book about Hibernate and a book about WebWork. We gave very introductory presentations of the frameworks, as it was impossible for us to teach them everything needed when even we are not very familiar with them. This would give them a sense of what the framework was used for and why it was used. We only had five days to conduct the workshop.

The workshop in Noida lasted for five days from the 2nd of May until the 6th. We encountered multiple problems during the workshop: power outs, loss of Internet connection, delays and other minor problems. Power outs caused the Internet connection to fail, which meant problems getting source code and Maven dependencies online. It was especially hard to get everyone ready in the morning to the decided time, which was mostly because we were many people.

There were some slides we did not have time to present because of the trouble mentioned above, but the workshop as a whole went pretty good. Another big issue is the language barrier between us and Indians. I believe they did not understand much of what we said, but it is hard to get them to admit they don't understand something in this situation. Both motivation and skill level was very varying throughout the course, which meant we had to push a lot when doing exercises. We held a feedback session in the end where we discussed want was good and what was bad. In general I believe they enjoyed themselves and many of them learnt something valuable. It seemed they got answers to many of their questions regarding RHIS 2.0, but they also got new ones.

I spent the rest of the stay in India and Noida working with their main developer. The Indian main developer and I continued the work me and my travel partner started in Bhopal, with reports. In the old version, everything was defined in one JSP file with SQL queries, JavaScript, HTML and Java code. We decided to create some WebWork Action files which was general and could be used on most different report formats. The plan was to start integrating the Graphical Analyser (GA) into RHIS 2.0 also, but we worked mostly on the reports until I went back home. A few days before I left, we got started on the GA and made it work within RHIS 2.0.

When I got home I continued to support PHIS India's main developer via mail and instant messaging. It was much harder than helping him in real life, much because we couldn't do the necessary small-talk we needed to understand each other and we had to get the same view of the actual problem. It was an absolute necessity that he committed changes frequently to the Subversion repository, so that we both had the code and a working version running. He told me that he had learned a lot working like we did in India, with a sort of a pair programming approach where he wrote the code and I supported him.

8 Discussion

The DSIP export project started out in the fall of 2006 and was to some degree discontinued in May of 2007. During the fall of 2006 the project was initiated and, I had created a working prototype which lacked some important functionality, like the mapping between the elements of DSIP and RHIS. During the winter before the trip to India I changed because of problems with reading files from the WAR file itself. When I got to India it was suggested that the exporter was incorporated into the advanced export functionality of RHIS 2.0 to have the same interface for all exporters in the system, this change was successful.

While in India I got hold of the database that had created the DataElements for DSIP and mapped the elements so that it was possible to export data to a MS Excel file. When this was in place I found an error that would change the functionality of the exporter. The Excel file created by the Java framework was not compatible with the Access importer. The system changed to both XML export and finally an exporter that connected to the local DSIP database and injected the values directly. During the course of the project I used different IOs like e-mail descriptions, discussions, prototypes, presentations, and the code itself. An Intermediary Object is something (written or orally) that represents the current state or a future state of my product. Concrete examples include the documentation space for the exporter, the various prototypes presented for different coordinators and discussion about the functionality, or our discussions of the problem with the Excel format.

In the rest of this chapter I will discuss the different research questions within the different categories based mainly on the project regarding the DSIP export. I will also discuss briefly the creation of the System Settings module and the training of Indian developers where there are relevant findings in my empirical data. The first category I will present a discussion for is the Intermediary Objects.

8.1 IO discussion

In this chapter I will discuss my questions related to my use of Intermediary objects. The use of objects can give a different view on the development project itself and its interaction with other participants. Since this is a qualitative research attempt and the concept of Intermediary Objects allow much subjective interpretation, the objects I present will probably not represent 100% accurately all of the objects used. I believe I have decided to describe the most relevant objects based on my field notes and the interpretation of my work. In the end of the chapter I will present some of the recommendations that could be made, based on my discussion.

Question 1

“Which Intermediary Objects were utilised and what effect did they have on my development?”

In my work on RHIS 2.0 I have created, presented, changed and read a multitude of different Intermediary Objects. The range of objects is from small descriptions, to presentations and prototypes. I this chapter I will discuss what I have used the different IOs for and what effect they had on the project itself.

The project started with a small description of the task we were supposed to deliver. This was a midterm report in the course INF 5750. This was a very open IO used only to describe to the teachers what we had done what we planned to do, and what troubles we had so far. This gave our group teacher a look into what we had done and he was pleased with the report.

The next important IO that was used was the documentation space of RHIS 2.0. For the groups final delivery we had to finalise this information. This information gave knowledge about the DSIP exporter to anyone who wanted read about RHIS 2.0. The final delivery in INF 5750 was also a presentation of our project. So we created a Power Point presentation of the module and went through what we had managed to do and what troubles we had encountered. This gave the coordinators information about what we had done and what we needed help with. It did also lead to a discussion about help with the database and we were supposed to work with some Indians who were coming to Norway in January, but they never came.

When I asked a question on why my mapping files were not read, I didn't get any answer from anyone on the developers e-mailing list. The fact that no one responded to my IO, meant that I had to create another object as a question directly to one developer. This led me to have a discussion with another developer instead, and the project changed from reading the mapping files from within the WAR to reading it from another place in the file system.

To make the exporter more user-friendly I decided to use the new exporter plug-in system created by a Norwegian developer to gather all exporters under advanced export. I created an IO describing my system and needs and sent it to the developers e-mailing list. The object described the state of the project now and I was asking if it would fit in the advanced export. With this change, I had no need for a separate web module to maintain and the interface for the exporter was changed to a more familiar one.

In Indian context the most important IO was the prototype itself. I showed it to the Indian national coordinator in a meeting at the hotel. This was the first time the Indians saw the application. Everything was described to him so that he could be able to present it in a meeting the next day. The prototype was never displayed in the meeting, but it gave a small part of PHIS India a view of how it worked. The fact that the object was never used to anything gave it very little value and did not drive the project forward in any way. Why the object was never shown was never explained to me. Because we never got to set it up properly I also sent him a long description of how to set it up. This time I showed the prototype, the functionality was in its most user-friendly state.

According to Kechaï & Choquet (2006, p. 3) a discussion can also be viewed as an IO. The IO which had the most effect in changes to the DSIP export module was the description and discussion of the module after I figured out the Excel solution did not work. This discussion started privately, but involved many different people with different interests. I explained the situation and proposed two options which created an Open IO, doing an export from RHIS and importing into the Access database or trying to connect to the Access database and inserting data straight into it. After some discussion, E-mail 8.1 was sent to me by a Norwegian coordinator, the replied text is written by me.

> The way it works now is that I export a XML file from RHIS which
> imports _directly_ to the database without errors. I also managed to
> export a Excel sheet from the database.I can even edit the data within
> DSIP. So it seems this works like it should.

So we also have the option of first importing the data directly to
Access, and then use the DSIP export to generate the Excel files. I
think this may be the least problematic strategy, at least in the
short term.

> I think this will be the way we will try to implement it for now. It
> seems the report tool they use is extremely sensitive with regard to
> changes in the Excel file. So I think I will try to write an extensive
> guide on how to use the exporter in RHIS and how to import the file
> into the Access application and then export via DSIP. This way the
> state level will hopefully see no difference, but there will be some
> more work for the person at PHC/district.

I think this sounds smart, Stian. We must of course assist as much as
we can to make it smooth.

E-mail 8.1 – From Norwegian coordinator, to 5 coordinators and me, subject: Problems with the IDSP export format, date: 19.03.2007

Based on this I started working on changing the module from exporting a MS Excel file to exporting a XML file which is compatible with Access import. This wasn't a major change since it already had all the variables ready, they had to be wrapped differently. When I had a solution for this with a manual and everything ready I started a new discussion about this functionality. Now I got a reply from a Norwegian coordinator shown in E-mail 8.2.

Glad to see this working, Stian.

But would it be possible to take the needed parameters from RHIS 2
rather than requiring the user to enter them in a MS-DOS window?

Do you see any possibility of shortening this procedure? There are
very many steps involved now. Also, it would be good if the
screenshots were in English.

E-mail 8.2 – From a Norwegian coordinator, to RHIS developer list, subject: [Dhis-dev] IDSP export setup-tool and manual, date 22.03.2007

Like it was in the case of Bojout & Blanco (2003, p. 215) where the CAD model did not provide all the information about the situation, my description of how the XML solution worked might not have been sufficient enough. Therefore it was decided that I should develop the XML export, but the solution itself wasn't good enough. In this process of choosing which of the solutions, me thinking the XML approach would be a simpler approach, might have become a *cognitive trap* (ibid) where I felt this was the best solution. The process of exporting and importing the XML document could have seemed easier and more user friendly than it turned out to be. But when my IO of the whole project was finished (with guide and pictures) it presented the whole picture and the solution was labelled too complex and it was decided to try another solution.

Because of this the discussion went back to the other solution, and we decided that I should try to implement a solution that connected to the actual MS Access database and injected the values directly. With a better IO describing the XML functionality we might have concluded differently the first time we had it up for discussion.

With the new functionality in order I was ready to get some feedback on this solution. The functionality was much better than the XML export, but not as user friendly as the first solution with MS Excel export. I updated every type of documentation like I did before sending out the last mail. Then the object that documented everything was sent to the PHIS India mailing list. This would present the IO to a new group of people, those who would actually use it and implement it. At least this was my intention and hope for this object. There were no answers from any users which affected the system in any way and there were no response on the actual content of the descriptions or the guide. Lack of involvement in the IO from Indian members caused the project to halt at this point.

The most useful approach for getting feedback from Indians based on my objects was for them to use and test the software itself. When showing the prototype IO of the System settings module I got feedback which resulted in changes.

While training and working with the Indian developer in Noida, we were sitting next to each other and had no need for use of any transferable IO. This means that we did not have the need for an object that had to be written down to be used. We mostly had oral discussions about the future of the product and the code itself was viewable right in front of us. But when I was back in Norway trying to support his work it was much more important to have a common object to use for support. He sent me descriptions of what was wrong and sometimes it would help me help him, but this was not the ideal solution. So the distributed cooperation worked best when he committed his work to the repository and I could view and test the code along with helping him, using an instant messaging client. Using the code repository as an Intermediary Object between us was very useful, if not a necessity. By this we clustered together code, discussion and problem description together to a very usable object. This gave us a better shared knowledge of the problem and I was closer to sitting next to him. We continued with this a few sessions during the summer of 2007. This was also a good thing for the project as a whole, as it taught the Indian developer to commit code regularly.

In my case there were no times any IO was irreversible like in the case of Bojout & Blanco (2003, p. 215) where the sketches became irreversible. My IOs, especially the code itself and the manual, were closed, because they were highly defined, but they had to be changed multiple times. The manual was dependent on the source code, and when the source code changed the documentation also had to be changed. The documentation space on the website was also highly coupled to the source code, but was neglected by me. I thought of this more as an IO for later use, documenting the module and how the technical parts worked. In project like RHIS 2 where the developers are mostly Master students that work for a limited time, it is important to have proper documentation of how different parts of the system works. This goes for me also, when I leave the project, it should be possible for another developer to continue or use my work easily. This documentation was not used during the development of the project and was not updated as often as the code was.

In the context of which country responded differently to different IOs it seems that the Norwegian context responded mostly to all kinds of IOs. While in India, the best way to present the project was with a live prototype with the person to present it to. When presenting the IO over e-mail it seemed it was not responded to in any degree, but while showing the prototype I got feedback and a discussion from it.

The documentation space is maybe the most important IO in regards to the whole project. In a research/university driven project like RHIS 2 there are a lot of people involved in a short period of time. This means that there have to be defined standards for everything for the project to function properly. In my DSIP export project I utilised the documentation multiple times. I found the description of the standard way to make an exporter and make it plug itself automatically into the advanced export tab. The documentation was also used to find a standard way to design a new database table in my system setting project. I designed my table without an integer primary key which according to the standard was wrong, so I implemented a primary key with auto generation instead. Some standards are well defined while other are more in an immaterial form, creating an IO that is not written, but it exists within the project in the minds of the participants.

Since I did not have the possibility of testing the prototype on actual users when I were in Bhopal I tried to use a different IO to present the project. Since source code and prototype was not possible I tried to use the documentation as an IO describing the state of the project at the time to get feedback from other people in the organisation. Unfortunately there were no responses to this type of IO from anyone on the Indian e-mailing list. This might have been caused by the way organisations work in India, where I had no “power” over them to give me feedback. There might also be the problem of understanding the text and/or e-mail since it was written in English. The best possible IO at the time did not give me any effect.

In an Open source project it is important that the utilisation of most Intermediary objects is public. When the developers use issue trackers and e-mailing lists to describe a state into an object everyone have access to the information. When people use private IOs like IM discussion and face-to-face discussion between only small parts of the team, the other participants in the project are left out. When they don’t have access to the private objects, they might not understand a change or disagree with a decision already made. This can be a problem for PHIS when they have a big core of developers placed in Oslo. I was careful about this issue when I developed my modules in India and tried to keep the use of private IOs to a minimum; only when I needed help with a very specific problem. I tried to present my objects in a public way using the e-mailing list in both Indian and global context.

The intermediary objects can at any time explain in what state the project is in and why it is in this state. The objects can function as log of the history of the project. The sum of all the IOs should describe the project at any given time. All the technical discussions can describe many of the choices taken in the code of the project. Discussion about the format of the export caused the code in Code snippet 7.2 (p. 59) to change to Code snippet 7.7 (p. 67). The IO and the source code caused all the documentation to change accordingly.

The term Intermediary Object is used very widely in different researches. It can be a *shared white board* (El-Kechaï & Choquet, 2006), *job allocation tables* (Papadimitriou and Pellegrin, 2007), *CAD models* and *calculation results* (Boujut & Laureillard, 2002). This brings me to what I see as a problem with the term Intermediary Object. Its definition might be too broad and it might become too subjective to distinguish between what is an IO and what is not. This may lead to problems with both external and internal validation. Are the arguments and findings in a study relevant for other cases (external) and what would the outcome of the study be if you changed the researcher, but kept the exact same data for analysis (changing one dependent variable - internal). The collection of different IOs can also be seen as an object, which causes the creation of multitude of possible objects. Should there be a term for a cluster of objects or will it be too difficult to separate what is a cluster and what is only one? These are issues that should be explored in more research.

I would argue that there needs to be a more specific condition or pattern for labelling something an IO, at least within the different research fields; industrial design, software engineering etc. Can every discussion related to the project be an IO, if so how would one gather the data? Another aspect is if an oral object should be classified as an IO. Is it an object when only orally presented or does it has to written down to be considered an IO? In a research perspective I would argue that it would be especially hard to undertake a valid research if orally presented information should be considered an IO. It will not be possible to collect everything, when a project is bigger than my case, which I believe it should be.

Even though I have argued that the definition of Intermediary Objects is too wide I would still like to propose two new classifications or attributes they can be described with. The first one is the relation or dependency between different objects. Most of the objects within a design or development process will have dependencies between each other, also of variable strength. In my case I have seen that my product will depend on the standards that the creators of the system have agreed on (ref. chp 7.2, p. 62, pg. 7, chp 7.2 p. 69 pg. 6, chp 7.3 p. 82 pg. 3). If standards change, my product will also have to change. Because of changes in the requirement my code changes also. This is evident in my empirical findings; Code example 7.2 (p. 59) changed to Code example 7.7 (p. 67) which later changed to Code example 7.11 (p. 71). When the product itself changes, the documentation might have to change, it depends on the change and which type of documentation (ref. chp 7.2 p. 72 pg. 4). This means there is a dependency, but it is weaker than the dependency between the requirements and the product. This is an attribute that can be investigated further, which might also help define the concept of Intermediary Objects in a clearer way.

Another classification that can help describe the object is the degree of specialisation of the content or the degree of competence one would need to understand or utilise it. This can range from high, to medium, and to low specialisation. In my case the different types are all found and have different functions. The source code is a highly specialised IO, where you need knowledge of Java development and the related frameworks to be able to understand it, while presentations and prototype displays would require much less specialisation to take advantage and developing an understanding of the object itself. The documentation and requirements can be defined as either one of the classifications, based on the content of the object. This classification might also be somewhat hard to define, similar to my argument that the action perspective discussed earlier. This term will also have to be studied in greater detail. Another similar classification is field of expertise required to use the object. In my case this might be split into development; which would be required to use the code (ref. chp 7.2 p. 71 pg. 2 & Code example 7.10, chp. 7.3 p. 79), facilitators (the person who installs and supports the installed system); which would have use of the documentation (ref Field note 7.2 p. 69) for the software. In the case presented by Boujut & Blanco (2003) they have three different fields of expertise; the forgers, manufacturing engineers and the designers.

Then why would these attributes or classifications be useful? I will present this briefly in the next part where I discuss the way that El-Kechaï and Choquet (2006) use the term of Intermediary Objects.

El-Kechaï and Choquet (2006) propose and use the definition of IOs as a way to better understand the design process itself and the use of communication between different participants. They (ibid, p. 5) conclude with "*This will lead the designers to perceive their design process, to transform it, to develop it, and to refine it during the process and thus, will enable them to build together the final product*". I would argue that a lot of software development/engineering is done without the organisation evaluating their process regularly. Writing documentation, creating models etc. is just a means to reach the final product; a working piece of software. Which it also is, but I would argue that using the concept of IOs to classify different objects used in the process and analysing them in a way they (ibid) have done, can help analyse and improve the process. They argue that this is done by "*If we give them the possibility to clarify collectively their design process according to their context and the organization they adopted, this will bring them to build a collective design approach in situ*" (ibid). I will argue that if you see the flow of the objects and how they interact with different people, you can plan meetings better because you know who will have to attend depending on which objects will be discussed. The clarification of which IOs are needed to continue the work can be defined and the process can be improved to match these findings.

In this part I will try analyse some of the aspects of my development process using my IOs with the same approach as I have described above. My system engineering approach should be undertaken in an agile and test-driven way. This is the way the RHIS 2.0 should be developed and the way we were trained to develop in INF 5750. If I look at the progress of the code in my development, I can clearly see that I develop in an agile and iterative way, creating small functionality piece by piece and changing the code when the circumstances are changing. But when it comes to developing it test-driven I can see that I never create the test before I create the implementation, which breaks with a part of the principle. Going back in my objects, I can also see that I totally lack a risk analysis where I would plan for what could go wrong under the development. The thought of the Excel format not being compatible didn't even strike me until I encountered the problem. Another object which could be evaluated was the discussion around, and my definition around the change to export to XML instead of MS Excel. If had described my solution more clearly I might have gotten asked not to implement it, and I would have saved the time it took. I would argue that if the IOs also would have gotten the attributes I discussed earlier, then the process could be even more analysed and improved. When you know the interdependencies between objects it is easier to plan for which one should be stabilised first and what happens if one changes. Given the attribute of speciality, or field of expertise, you would know which people should be involved in creating, discussing and finalising the different objects in the development process.

I will now try to sum up the different approaches to improving the concept of Intermediary Objects and recommend new features to the notion of IOs. I strongly believe the concept of IOs need more clarification of what can be defined as IO, as it is being used differently in multiple research. I believe that my two properties used to describe the objects can be an interesting way to enhance the concept. Both concept of dependencies between objects, and the property defining its degree of specialisation can improve the way IOs are used in understanding, describing and improving a design process. The way El-Kechaï & Choquet (2006) uses the concept is an interesting way of utilising the concept to improve a process, and I believe this should be investigated further.

Question 2

"How does my Intermediary Objects relate to the concepts created by Papadimitriou and Pellegrin?"

Papadimitriou and Pellegrin defined a set of IOs that they found in their research. The framework has only been tested in one case and as they write it has not been yet properly validated by other research. I think it is interesting to see if some of my IOs can be put into the same categories and see if they fit the timeframe they found them in. My case differs somewhat from their case in especially size, and also my project was never completed in a proper manner, it is still on hold, because it needs to be tested properly by PHIS India before it can be fully implemented.

The first IO they defined was the *Oral ideal* object which represents an IO that should be able to support oral interaction and describe a general principle. In my project this type of object was most present in the beginning, but was also brought up in later parts of the project when needed. First of all everything in the RHIS 2 system should follow the standards decided for the project; coding style, everything should have unit tests, follow the rules of the code repository etc. The user interface was supposed to be user friendly and use the same design principles already inscribed into the system. Most of these objects were from the INF 5750 course, from the PHIS website or presented to me by members of the team. They are all ideal in the way they present information about how something should be done in a perfect or best solution. This fits with their research that these objects appear in the beginning. Under the way there were also some IOs that had the same intention and were of the same type. It was at one time proposed that it should be incorporated into the advanced export to follow the standards of RHIS 2.0. At another time it was commented that it was not easy enough to configure and there were too many steps in the process of exporting values. In both cases the change was conducted, only in the case of user-friendliness the object was up for discussion. Another *Oral ideal* object that was really just orally presented was the need for a separate license and repository for the Indian specific code. The reason was so that it could not be misused by other Indian companies taking the RHIS 2.0 code and selling it as their own. This requirement object was orally presented at a meeting as an *ideal* standard, but the object was never followed up by neither me nor the Indian coordinators. It will and might be an important *oral ideal* if they implement this export and also are able to integrate other Indian systems.

Written discourse was the next defined OI by the researchers. The group of IOs I used that fit into this category is the more concrete information about implementing the Oral Ideals mentioned earlier. In the case of the advanced export, it was the help from core developers and the information about how to implement it from the RHIS 2.0 documentation that actually caused the implementation to occur. Another example was the case of creating an ID in the database; the *Oral ideal* is to follow the standards provided, while the *Written discourse* itself was the documentation page describing how to create a new table and the rules for the technical implementation. This last *Written discourse* IO did not come into my project in the beginning of the actual start of development, but in the beginning of the change to this solution. This is because my project went through different stages where some of the development started from scratch, while other aspects did not. The project went through different iterations where the new functionality was introduced but the objects describing the logic behind the application remained the same. This was not planned, but because of the different troubles and different *oral ideals* described earlier made it necessary. Another *Written discourse* object used by me was the definition on how to change the database between the different releases. This was crucial in maintaining my Kerala database up to date on the changes in the database structure.

The third statistically found object Papadimitriou and Pellegrin found was the *Codified technique*. This object can specify a rule, method and/or a performance criterion and is has a well established meaning. One important IO created and used by me was the manual and FAQ for the exporter. This defined the methods for implementing it and how to use it. In their case this object usually occurred late in the project and this matches my case where this was created when I thought the exporter was close to finished. Unfortunately it had to be changed multiple times in accordance with how the exporter itself evolved.

Most of the e-mail discussions fall into every category Papadimitriou and Pellegrin have defined. Some parts are very technical information and resemble *Codified technique* and other are closer to *Oral ideal* objects. The comments on making the export easier to set up is an *Oral ideal*, while help on how to connect to a MS Access database is a *Codified technique*.

Papadimitriou and Pellegrin also found two other types of IOs that were of significance. This was the *Quasi-technical* object which glues together information between the technical and business actors. In my case I have used multiple objects of this kind. The first one was the documentation on the RHIS 2.0 page which contained a lot of information for all kinds of user and wired together both technical and more descriptive information. One other IO was the presentation of the module in the course INF 5750, which presented information for all types of viewers. We had both a technical and a high lever description of the state of the project.

The last IO they (ibid) found was the *Pilot application* which is a production ready code tested in a real setting to ensure the project is complete and follows the requirements established. They are used in the border between project and the permanent implementation. My prototype object is ready for a pilot implementation now, but this has not been conducted yet. Like in their (ibid) case, my pilot application will also be used in the last face of the project. My pilot is also closed in the sense that I have explored multiple different solutions and come to this state of functionality, but it is also open in a sense that it can be massively changed after a pilot implementation and after receiving feedback from actual users.

As mentioned in the literature chapter and above, Papadimitriou and Pellegrin (2007) try to validate a set of perspectives to classify their objects. These perspectives are defined by Boujut and Laureillard in their paper "*L'approche du processus de conception de produit par les objets interme´diaires: le cas des pie`ces forge´es a RVI*", which sadly is written in French. But in the way Papadimitriou and Pellegrin present the action perspective, it is based on the subject of change for an object; it can be open or closed. Open meaning they are allowing interpretation, and closed when they are considered definite and not subject to change (ibid). The way they have defined the action perspective, means it must be one of the two. This is how they have categorised the list of their IOs, with the object being either open or closed.

One of the special findings they present is the *Pilot application* which can be both open and closed. This is not found statistically, but they find it from studying the research data. They found that the Pilot application had impact on the process and I agree that this will not be found in the statistical data because it will probably happen very few times during a project. This brings me to my point; is it possible for every object to be only one of the two? I would argue that some of my objects appear to be closed, but must allow for interpretation and change at a later time. The early requirements in my project were early defined. It was to integrate the functionality of DSIP into RHIS 2.0. Which meant the need for using DSIP would disappear. My requirement (ref. chp 7.2, p. 51, pg. 4) was to export a MS Excel sheet which replicated the report function of DSIP. These two are under my impression the documented requirements, which mean they form an IO, and the requirements are not up for interpretation. In the same requirement object there are also *open* requirements like it should be incorporated into the look of the rest of the system and it should be user friendly. If you don't split the requirements object into two different objects the same object can be seen as both *open* and *closed*.

My documentation along the way was also perceived to be *closed* near the end of every solution. Then because the product itself changed, the related *closed* IOs had to be changed back to open again. Papadimitriou and Pellegrin argue that the objects evolve from *open* to *closed*, while I would argue that the objects are interrelated and may change back to open again. I will argue that the action perspective of the objects defined by Papa is not clear enough, because both they and I found objects that can be both. One aspect similar to this aspect is the aspect of time or maturity.

Recommendations

I would argue that there is a need for more clear clarifications for the Intermediary Objects. This becomes a problem in all research, because of the subjectivity of choosing what an object is and what it is not. The problem can be described with the term *anectodtalism* described in the methods chapter, where the researcher will have problems convincing both himself and the reader that their findings are indisputably based on a critical investigation of all data, and not only of a small well chosen examples.

I would also like to propose adding more properties to them term, which can describe the object in a better way. Describing the way the objects are interrelated will give the researcher a better overview of the whole design process. Another property I would recommend further studies of is the need for classification of the degree of specialisation or the degree of knowledge in a field to understand and use it.

I believe the way El-Kechaï & Choquet (2006) use the concept is another interesting way to use the Intermediary Objects in research. The use of IOs to improve a process can also have an advantage of the newly proposed properties also.

Now I have presented my discussions and recommendations of the concept of Intermediary Objects. The next chapter will contain the discussion about software development.

8.2 Software development discussion

In this chapter I will discuss my research question which is related to Software development. The chapter contains only one question, regarding distributed development.

Question 1

“What measures does Programme of Health Information System take to support distributed development of their software?”

In PHIS there is a high degree of temporal distance. There are people working in Vietnam, India, Norway and multiple African countries. This means that there is a time difference of 5 hours between Norway and Vietnam. When the Norwegian office hours begin, the Vietnamese only have 3 hours left of their day. Because of this the possibility for synchronous communication is lowered and the need for asynchronous is bigger. PHIS does not support any synchronous communication software, but most of the developers and coordinators use Windows Live Messenger, Google Talk or Skype. They also conduct meetings in real-time which have been conducted on both Skype and Windows Live Messenger. The most important asynchronous communication is e-mail and the e-mail mailing lists, which are used frequently and well supported by the organisation. The documentation space is also well suited for sharing information in an asynchronous way. When the geographical distance is high it is important to utilise the asynchronous communication in a good way. One way I was proposed to do is to plan to send requests for help at the proper times, so that you get the best help. There is no point of asking for help before the other part is at work, but communicate at the end of you work hours and hopefully you will receive help by the time you start again.

In PHIS there is also a big geographical distance between coordinators and developers. Even within one country there could be big geographical distances, for example India. PHIS is good at bringing different people together to undertake team working. Developers travel between countries to help, learn and get to know each other. Good personal relations between developers in different countries generate better understanding and communication. All master students that would like to participate in the PHIS project is asked to travel to another country where PHIS needs support. Within PHIS India the workshop conducted with us present was a way for many different developers in India to meet each other and develop relations. Because I developed good relations with their main developer, I supported his work also after I left India.

The social-cultural distance in PHIS is also large. There are big differences in how the organisation within each country is managed. The way management and structures are handled in countries like India and Vietnam is different than in Scandinavia and Norway. This can lead to problems when coordinators and developers from different social-cultural places interact and communicate. Therefore it is important for PHIS members to be aware of this distance and they help lowering it by allowing people to travel to other cultures and see how they work. When there is a stable workforce in one of the countries, they will become more and more accustom to the culture of the other countries involved and this will lead to a better understanding and development. Utilising Intermediary objects in this context can help lower the distance. Using documentation, prototypes or other written objects in communication can minimise the tension.

The RHIS 2.0 project does not have clear distribution rationale which is recommended by Lings et al (2007, p. 4-8), there is always a lack of developers, especially outside of Norway. Before releases they try to delegate some tasks between the different developers, but it doesn't always work out like planned. RHIS 2.0 is very modular and is created this way because of the distribution of developers in the project. This works well for the distributed collaboration needed in this project.

Lings et al (2007, p. 4-8) also proposes the need for cultural mediation, which is a person who can function as a link between the different dislocated teams. In the RHIS 2.0 project this is often done in collaboration with the point of temporary collaboration. When Norwegian developers travel to other countries, they can develop a good relationship with people in the country they travel to. There are also people from other countries coming to Norway working with developers here and joining courses at the university to develop their skills. The link between different countries can be somewhat problematic because of the different organisational structures in the different country. In many of the developing countries they are involved in all the communication go through the leader instead of directly communicating with developers or other people involved in the organisation.

RHIS 2.0 have a common platform in its website storing and presenting at least about the teams and the goals for the project. Information about the progress might not be stated clearly in the web site, but in the e-mailing list discussions and in the code repository itself, which is also a part of the common platform. The tool base is also defined on this page which guides and documentation about how to replicate (i.e. set it up) it on every site. There is also heterogeneity in methods and tools, but everything might not have been well planned ahead. It has been held meetings using both Windows Live Messenger and Skype.

Two points I feel the RHIS 2.0 project does properly is having a team leader which has regularly contact with the different locations. This can be lowered by for example having a common Intermediary object which is updated by different coordinators asynchronously. This can lower the bar for sharing information because it can be done when each and every person have the time. The second point is "*all understandings, clarify and agree on goals before the project starts*" (ibid). The last point is really not possible in the type of development style RHIS 2.0 is developed in.

In general I believe the team around RHIS 2.0 is delivering good support for doing distributed development for their employees and students.

This was the end of my discussion chapter. I have discussed both the concept of Intermediary Objects and one question related to software development. In the next chapter I will conclude my whole research.

9 Conclusion

In this chapter I will try to summarise the results of my research and to conclude what I have learnt from exploring the use and effect of Intermediary objects in my system development. Being that my research as many limitations I will also touch upon the validity of the research.

If I was to undertake another research project of this size I would definitely do some actions differently. I regret that I did the research before I had taken all my courses because there was a big gap of time between the field trip and the analysis and writing. Another thing I would do differently is to undertake a class in research methods before I travelled which would give me a better picture of how to take notes, what to focus on and how to conduct a good research project. If I were to do another research I would also have had a more defined research question, even though my effort showed me that this isn't always doable. My effort to focus heavily on the integration failed because of reasons to at least some degree out of my control. This happened because some of the software they had hoped to integrate was not possible to integrate.

One big issue in the research of Intermediary Objects, including my research, is the broad definition of IOs. It can be defined to a multitude of things in a development project where multiple people are involved. This causes the definition of what can be an IO to be very subjective and may cause very different results based on the researcher's background and his or hers personal perception of what an IO is. Field notes and what to write in the notes are also affected by the researcher's idea of what is interesting and important.

Using Intermediary Objects to see a development effort gave me a new and different view on the communication, documentation and decision making. One aspect where the IOs are important is when a developer joins the project; the person needs to learn about the system. The different types of objects can introduce him or her in different ways to the standards, code and principles. In a project like RHIS 2.0 where students come and go often, as most developers are only with the project for maximum two years. This is why it is important for me create complete and different objects for all different purposes; implementers, users, new developers etc.

Another point that I discovered was that different objects work for different people. In India they reacted to different objects than the one reacted upon from Norway. This can have some relation with the organisation and culture, but I believe it is mostly because of the position the people I interacted with. In Norway I had most contact with developers, while in India there were mostly coordinators, who at least responded to my objects.

Since there is not much research on Intermediary Objects, it is not much for me to compare research with. My main research question was very open and it is hard to compare to other findings. The second question comparing my findings with the findings of Papadimitriou and Pellegrin is more comparable even though their case is not in the same field as mine. Most of the objects I found that I was able to their four/six types happened in the same period of the project as in their case. My project had a very long last part where there were changes in the actual functionality. The action perspective is the one I think is not good enough, as I would argue that if there is a point of clarifying the property, it should only be one of the two.

I made some propositions with regard to the theory of Intermediary Objects. The addition of new properties like interdependencies and degree of specialisation can bring more quality to the concept. The way El-Kechaï & Choquet (2006) used the concept, is another direction I think should be more researched. I believe these should be studied further to make the concept more mature, and so that it can be used properly in research.

This question answered if PHIS follow the guidelines set for doing distributed development. The basis for answering this question is being a part of the RHIS 2.0 development team for a period of time and doing development both in Norway and India. The limitations in answering this question are (1) I have only used a small part of the theory about the subject to generate the guide for developing in a distributed way and (2) the relative short and little participation in the organisation. This question is also limited to just this organisation. With these limitations present I still feel that PHIS have a good platform for doing distributed development.

I tried to undertake a research project where I was assigned to create a new module for the RHIS 2.0 system and explore what IOs where used and what effect they had. Even though the integration module failed I believe that I to some degree answered my research question in a decent way. In addition to what has been presented in this thesis, I have learned incredibly much from participating in an OSS project, and from all my experiences from India.

10 Abbreviations

AI	= Artificial Intelligence
AOP	= Aspect Oriented Programming
API	= Application Programming Interface
BLOB	= Binary Large Object
BSD	= Berkley Software Distribution
CAQDAS	= Computer Assisted Qualitative Data Analysis Software
CDC	= Centre for Disease Control
CHC	= Community Health Centre
CVS	= Concurrent Version System
DFID	= Department for International Development
DSIP	= Disease Surveillance Integration Project
EDS	= Essential Datasets
FAQ	= Frequently Asked Questions
FHMIS	= Family Welfare and Health Information Monitoring System
FOSS	= Free and Open Source Software
FSF	= Free Software Foundation
GA	= Graphical Analyser
GNU	= GNU's Not Unix
GOI	= Government of India
GPL	= GNU General Public License
HMIS	= Health Management Information System
ICMR	= Indian Council of Medical Research
ICT	= Information and Communication Technology
IO	= Intermediary Object
IRC	= Internet Relay Chat
IS	= Information System
IT	= Information Technology
JAR	= Java Archive
JSP	= Java Server Pages
JSY	= Janani Suraksha Yojana
LPGL	= Lesser GNU Public License
MOHFW	= Ministry of Health and Family Welfare
MS	= Microsoft
MVC	= Model, View, Controller
NICD	= National Institute of Communicable Diseases
NIH	= National Institute of Health
NORAD	= Norwegian Development Agency
NRHM	= National Rural Health Mission
ORM	= Object Relational Mapping
OS	= Operating System
OSI	= Open Source Initiative
OSS	= Open Source Software
PHC	= Primary Health Centre
PHIS	= Programme of Health Information System

PHP	= Hypertext Pre-processor
PIP	= Project Implementation Plan
RHIS	= Rural Health Information System
RIMS	= Routine Immunisation Monitoring System
RE	= Reverse Engineering
SVN	= Subversion
UEM	= University Eduardo Mondlane
UiO	= University of Oslo
USAID	= United States Agency for International Development
USB	= Universal Serial Bus
WAR	= Web Application Archive
WHO	= World Health Organisation
XML	= eXtensible Markup Language

11 References

Absolute Basics; http://folk.uio.no/torgeilo/annet/DHIS_2.0_-_Absolute_Basics.pdf, 2006, downloaded 2008-03-25

Atlasti.com; read 2008-01-15; http://www.atlasti.com/

Baskerville R. & Wood-Harper A. T., Diversity in information systems action research methods, European Journal of Information Systems 7 p. 90-107; 1998

Boddy D., Boonstra A., Kennedy G.; Managing Information Systems – An Organisational Perspective; Pearson Education; 2005

Boujot J. F. & Blanco E., Intermediary Objects as a Means to Foster Co-operation in Engineering Design, Computer Supported Cooperative Work 12 p. 205-219; 2003

Braa J., Monteiro E. & Sahay S., MISQ Quarterly MIS Quarterly Vol. 28 No. 3 09/2004
Networks of Action - Sustainable Health Information Systems Across Developing Countries

Câmara G & Fonseca F; Information Policies and Open Source Software in Developing Countries; Journal of the American Society for Information Science and Technology; 2007

CIA; https://www.cia.gov/library/publications/the-world-factbook/geos/in.html; read 2008-04-01

Confluence; http://www.atlassian.com/software/confluence/; read 2008-03-06

Crystal Reports, read 2008-02-14,
http://www.co.multnomah.or.us/dss/info/crystal/ce8_introductory_tips.pdf

Database API; http://www.hisp.info/confluence/display/DOC/Database+API; read 2008-04-15

Davies F. D.; Perceived usefulness, percieved ease of use, and user acceptance of information technology; MIS Quarterly; p. 319-340; 1989

Eclipse1; http://www.eclipse.org/; read 2008-03-06

Eclipse2; http://www.opensource.org/licenses/eclipse-1.0.php; read 2008-03-06

Edwards M. & Lippeveld T.; Decision Support Systems for Improving the Analytic Capacity of Decentralized Routine Health Information Systems in Developing Countries; Proceedings of the 37th Hawaii International Conference on System Sciences; 2004

El-Kechaï Hassina & Choquet Christophe; Understanding the Collective Design Process by Analyzing Intermediary Objects, Proceedings of the Sixth International Conference on Advanced Learning Technologies; 2006

Fogel K.; Producing Open Source Software: How to Run a Successful Free Software Project; http://producingoss.com/; 2005

GNU; http://www.gnu.org/philosophy/categories.html#ProprietarySoftware; updated 2007-12-03, read 2008-03-03

Hammer M. & Champy J., Reengineering the Corporation: A Manifesto for Business Revolution, 1993, Harper Collins, New York

Health workers operational manual; read 2008-01-23; http://www.nicd.nic.in/IDSP_docs%5CMANUAL%5COperational%20Manual%20for%20Health%20Workers%5CHealth%20Workers%20Manual.pdf

Hibernate1; http://www.hibernate.org/; read 2008-03-05

Hibernate2; http://www.hibernate.org/344.html; read 2008-03-05

Java; http://en.wikipedia.org/wiki/Java(programming_language); updated 2008-03-05; read 2008-03-05

Jetty; http://www.mortbay.org/; read 2008-03-06

JSY, read 2007-04-30, http://www.hindu.com/2005/09/11/stories/2005091107130700.htm

Larson A.; Making Sense of Collaboration: The Challenge of Thinking Together in Global Design Teams; 2003

Lee C.; Boundary Negotiation Artifacts: Unbinding the Routine of Boundary Objects and Embracing Chaos in Collaborative Work; Computer Supported Cooperative Work 16 p. 307-339, 2007

Lings B., Lundell B., Ågerfalk P., Fitzgerald B.; A reference model for successful Distributed Development of Software Systems; International Conference on Global Software Engineering; 2007

Lutters W.G. & Ackerman M.S.; Achieving Safety: A Field Study of Boundary Objects in Aircraft Technical Support; Computer Supported Cooperative Work, CSCW 2002; New Orleans, Louisiana, USA: The Association for Computing Machinery (ACM); 2002

Mailman; http://www.list.org/; read 2008-03-06

Mambrey P. & Robinson M.; Understanding the Role of Documents in a Hierarchical Flow of Work; Group 97, Phoenix, AZ: ACM; 1997

Marshall C. & Gretchen R. B.; Designing Qualitative Research; Sage Publications; 1998

Maven, http://maven.apache.org/what-is-maven.html; read 2008-03-06

Miles M.B., Huberman A.M. ; Qualitative Data Analysis: A Sourcebook of New Methods; Newbury Park, CA; Sage; 1984

Monteiro E, Østerlie T, Rolland K, Røyrvik E; Keeping it going: The everyday practices of open source software; submitted for reviewing; 2004

NRHM, read 2007-04-29, http://mohfw.nic.in/NRHM/Documents/NRHM%20Mission%20Document.pdf

NRHM progress, read 2007-04-29, http://mohfw.nic.in/NRHM/NRHM%20%96%20THE%20PROGRESS%20SO%20FAR.htm

NICD; Project implementation plan; 14.05.2004; http://www.nicd.nic.in/idsp_Docs%5CNPIP.pdf

Nordal K., The Challenge of Being Open; Master thesis UiO; http://www.hisp.info/confluence/download/attachments/2374/KristianNordal.pdf?version=1; 2006

Organisation homepage -> Vision; read 2007-04-27; http://www.hisp.org

Organisation Newsletter; Gianluca Miscione; read 2007-08-13; http://www.hispkerala.org/latest_downloads/HISP%20Newsletter%201st%20issue%202007.pdf

Papadimitriou K., Pellegrin C.; Dynamics of a project through Intermediary Objects of Design (IODs): A sensemaking perspective; International Journal of Project Management 25; 2007

Pawlowski S.D. and Robey D. et al.; Supporting Shared Information Systems: Boundary Objects, Communities, and Brokering; Twenty First International Conference on Information Systems; Atlanta, GA: Association for Information Systems; 2000

Proprietary Software; http://en.wikipedia.org/wiki/Proprietary_software; updated 29-02-2008; read 03-03-2008

Raymond E.S.; The cathedral and the bazaar: Musings on Linux and open source by an accidental revolutionary (Rev. ed.); O'Reilly & Associates; 2001

Reverse Engineering; http://en.wikipedia.org/wiki/Reverse_engineering; updated 13-03-2008; read 13-03-2008

Riding P., Fowell S. & Levy P.; An action research approach to curriculum development; Information Research, http://InformationR.net/ir/1-1/paper2.html; 1995

Spring, http://www.springframework.org/about; read 2008-03-05

Star, S.L. and J.R. Griesemer (1989): Institutional Ecology, ‘Translations’ and Boundary Objects: Amateurs and Professionals in Berkeley’s Museum of Vertebrate Zoology, Social Studies of Science vol. 19, 387–420; 1989

Starring K. & Titlestad, O.; Networks of Open Source Health Care Action; 2006, UiO

Silverman D.; Doing qualitative research 2nd edition; 2005; SAGE publication

Silverman D.; Interpreting qualitative data 3rd edition; 2006; SAGE publication

Sæbø J. & Titlestad O.; Evaluation of a Bottom-up Action Research Approach in a Centralised Setting: HISP in Cuba; Proceedings of the 37th Hawaii International Conference on System Sciences; 2004

Track; http://trac.edgewall.org/; read 2008-03-06

Vinck D.; Ethnographic studies in design activities: looking at the objects too; Ethnographic Organizational Studies, St Gallen, Switzerland

Warne L. & Hart D.; The Impact of Organizational Politics on Information Systems Project Failure - A Case Study (The 29th Annual Hawaii International Conference on System Sciences 1996)

WebWork; http://www.opensymphony.com/webwork/; read 2008-03-05

Weerawarana S. & Weeratunga J.; Open Source in Developing Countries; www.sida.se/publications; 2004

12 Appendices

12.1 Appendix A – Table of Intermediary Object use and effect

Date	Description	Type	Effect	Context
Sep 06	Was presented the RHIS project in class session	Documentation	Was introduced to the system	Norway
XX Dec 06	Presented the project on the doucmentation part of hisp.info	Documentation	Everyone interested can find information about the project	Both
XX Dec 06	Held a presentation for the PHIS coordinators and class of INF 5750	Presentation	The PHIS team now knew our achievements and troubles so far	Norway
23 Jan 07	Desribed my problem with the property files	E-mail	No answer, the use of IO had no effect	Norway
24 Feb 07	Described my solution and asked how it could be incorporated into advanced export	E-mail	The software design changed and it was incorporated into advanced export	Norway
28 Feb 07	Showed Indian national coordinator the software in a meeting	Prototype	Understood how it worked and was ready to present it	India
1 Mar 07	Sent a mail with explanation of what to do at presentation	E-mail / Documentation	No answer, the exporter was not shown	India
16 Mar 07	Read the design documents about DSIP	Documentation	No information, no effect	India
17 Mar 07	Sent a big description about the problem with DSIP in a mail	E-mail	Agreement that old solution is not working, new solution proposed	Both
22 Mar 07	New descripton of the new solution presented	E-mail	Solution not accepted, new solution proposed	Both
26 Mar 07	New description of how the system worked was sent to some coordinators	E-mail	Some changes required by the coordinators back to another earlier state.	Both
28 Mar 07	Description of the system as it was sent	E-mail / Documentation	Only answers from coordinators, wrote more	India

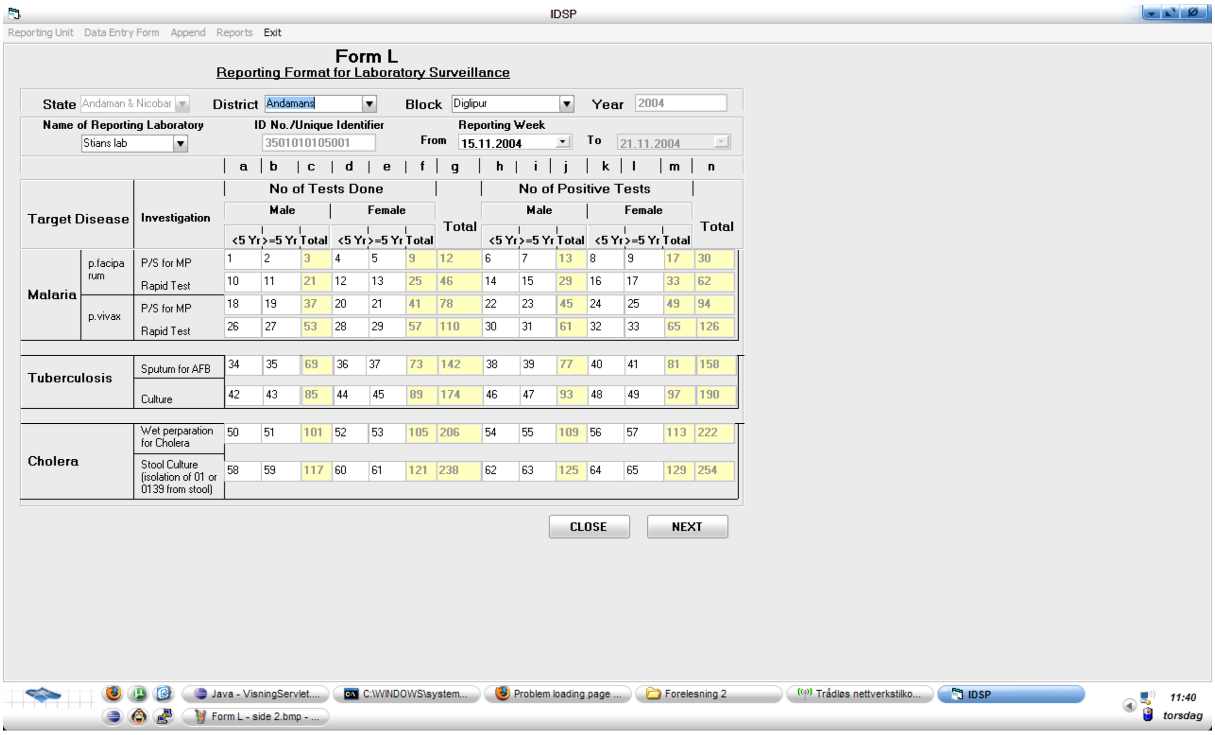
	to the Indian mailing list. It contained all the documentation I had written		about how to add the module to the build. No effect on the software itself.	
Mar 07	Created a manual describing the functionality of the system	Documentation	No effect since it was not read by anyone relevant	India
XX May 07	Presenting the working version to coordinator for Kerala state	Prototype / Presentation	The person was able to see how it worked and how to set it up	India
16 May 07	Sent a mail to the coordinator for Kerala with more information	E-mail / Documentaion	Answer, but no other effect	India

12.2 Appendix B – Table of problems

Occurred	Problem description	Type	Solved
Oct 06	The course group teachers went to SA and could not support our work in class	Issue	Yes
Oct 06	The export module that was in place was not properly made and we could not base our exporter on this	Issue	Yes, Hans made an Exporter interface
Oct 06	A group of seven people working in one branch caused trouble	Organisational	Yes
Jan 07	The RHIS software is still under development and a lot changes in the core creates problems	Issue	Yes
Jan 07	PHIS India representatives were supposed to come to Norway before we travelled to India	Problem	Yes, we travelled
Jan 07	No Indian database to test the exporter with	Problem	Yes, 24 Feb 07 got Kerala database
19 Jan 07	The solution with reading properties files from within the war file stopped working	Problem	Yes, 29 Jan. Put the files in user home instead
23 Jan 07	No answer to mail about properties files.	Organisational	Yes, se last problem
31 Jan 07	Forgetting to commit changes to subversion caused problems when working in multiple places	Organisational	
24 Feb 07	When making a new module I didn't know which modules to include because of changes in core	Issue	Yes
24 Feb 07	Changes to the database in the new version caused my Indian database to not function so I sent an e-mail	Issue	Yes, e-mail reply 24 Feb 07
25 Feb 07	Trying to build the system offline would not work.	Problem in India	Yes, remove the report tool module from building
01 Mar 07	PHIS India wanted to have their own license on the DSIP exporter and store it in another repository because of other NGO in India	Problem	No
01 Mar 07	PHIS India wanted a their own tab in RHIS 2 with among others the DSIP exporter	Issue	No
03 Mar 07	Big databases causes my machine to crash when importing	Issue	Yes
03 Mar 07	Most Indian machines are infected with viruses	Issue	
06 Mar 07	PHIS India wanted to showcase the exporter before it was done	Issue	
07 Mar 07	Database foreign keys must be deleted before the system would start	Issue	Yes

08 Mar 07	PHIS India had made an error in the DSIP data set for RHIS, two data elements were switched	Issue	Yes
08 Mar 07	DSIP will not import the ID number from my Excel sheets	Problem	No
08 Mar 07	DSIP will not export my imported values	Problem	No
09 Mar 07	DSIP will not properly import my values	Problem	No
18 Mar 07	It took me 25 minutes to delete my RHIS folder	Issue	Yes
19 Mar 07	We could not have the source code for the DSIP software	Issue	No
19 Mar 07	Pressure to do other tasks for PHIS (ref Knut mail)	Issue	
21 Mar 07	No feedback on my work on the FAQ and the setup tool	Organisational	No
22 Mar 07	Database update script is not compatible with my Kerala database	Issue	Yes
22 Mar 07	XML export/import was not good enough after it was already settled for and created	Problem	Yes, direct connection to database
22 Mar 07	Figuring out how MS Access date to work with no feedback on errors	Issue	Yes
24 Mar 07	DSIP variables must be stored somewhere	Problem	Yes, created data set
26 Mar 07	We have still not been given access to the PHIS India mailing list	Organisational	Yes, 27 Mar 07
26 Mar 07	Solution with DSIP information in data sets was not good enough	Problem	Yes, going back to properties
27 Mar 07	A new User role system was created and I could not access my own system	Issue	Yes, help from Hans
28 Mar 07	No feedback received on mail to the Indian PHIS mailing lists	Organisational	No
24 Apr 07	John was bragging about how DSIP was already integrated and RIMS integration was under development	Organisational	
16 May 07	No reply from Jeswin on my mail to him about DSIP	Problem	No
29 Jan 08	DSIP export module is still not used	Problem	No

12.3 Appendix C – Screenshots of DSIP



Form L

Reporting Format for Laboratory Surveillance

State: Andaman & Nicobar   District: Andamans   Block: Diglipur   Year: 2004

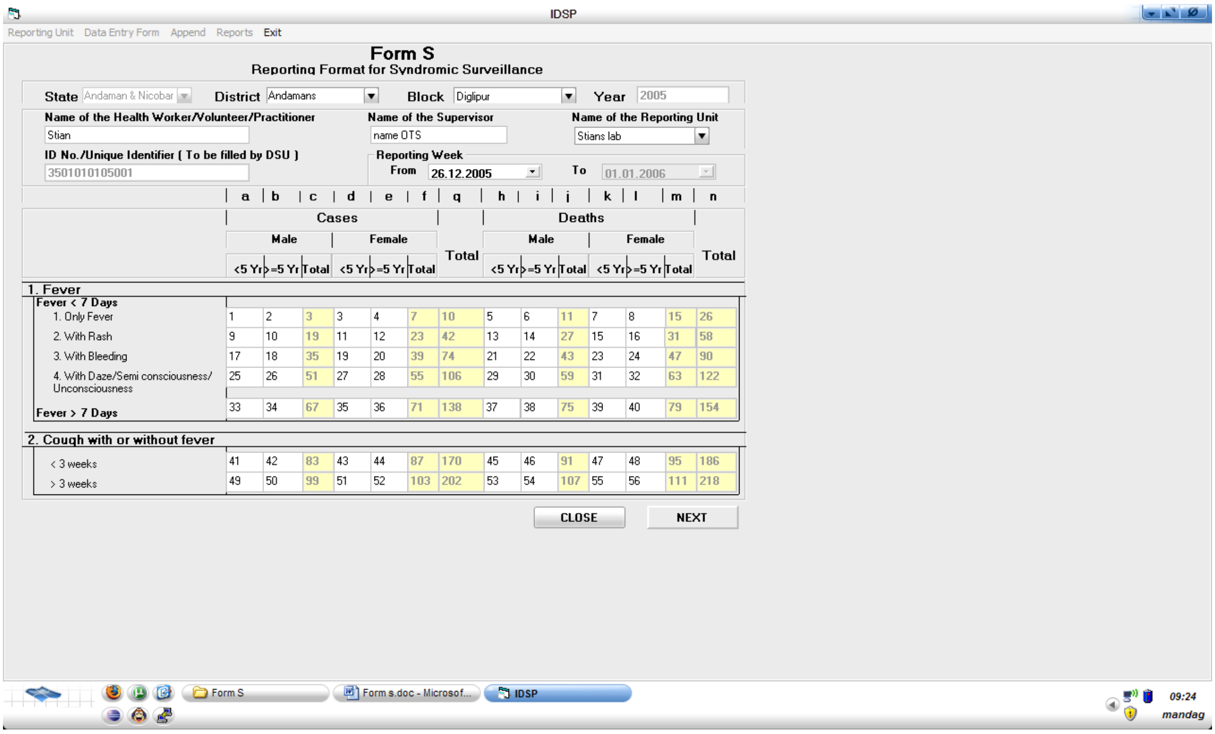
Name of Reporting Laboratory: Stians lab

ID No./Unique Identifier: 3501010105001

Reporting Week From: 15.11.2004   To: 21.11.2004

			a	b	c	d	e	f	g	h	i	j	k	l	m	n
Target Disease		Investigation	No of Tests Done							No of Positive Tests						
			Male			Female			Total	Male			Female			Total
			<5 Yr	>=5 Yr	Total	<5 Yr	>=5 Yr	Total		<5 Yr	>=5 Yr	Total	<5 Yr	>=5 Yr	Total	
Malaria	p.facipa rum	P/S for MP	1	2	3	4	5	9	12	6	7	13	8	9	17	30
		Rapid Test	10	11	21	12	13	25	46	14	15	29	16	17	33	62
	p.vivax	P/S for MP	18	19	37	20	21	41	78	22	23	45	24	25	49	94
		Rapid Test	26	27	53	28	29	57	110	30	31	61	32	33	65	126
Tuberculosis		Sputum for AFB	34	35	69	36	37	73	142	38	39	77	40	41	81	158
		Culture	42	43	85	44	45	89	174	46	47	93	48	49	97	190
Cholera		Wet perparation for Cholera	50	51	101	52	53	105	206	54	55	109	56	57	113	222
		Stool Culture (isolation of 01 or 0139 from stool)	58	59	117	60	61	121	238	62	63	125	64	65	129	254

Screenshot of Form L page 1



Form S

Reporting Format for Syndromic Surveillance

State: Andaman & Nicobar   District: Andamans   Block: Diglipur   Year: 2005

Name of the Health Worker/Volunteer/Practitioner: Stian

Name of the Supervisor: name OTS

Name of the Reporting Unit: Stians lab

ID No./Unique Identifier (To be filled by DSU): 3501010105001

Reporting Week From: 26.12.2005   To: 01.01.2006

	a	b	c	d	e	f	q	h	i	j	k	l	m	n
	Cases							Deaths						
	Male			Female			Total	Male			Female			Total
	<5 Yr	>=5 Yr	Total	<5 Yr	>=5 Yr	Total		<5 Yr	>=5 Yr	Total	<5 Yr	>=5 Yr	Total	
1. Fever														
Fever < 7 Days														
1. Only Fever	1	2	3	3	4	7	10	5	6	11	7	8	15	26
2. With Rash	9	10	19	11	12	23	42	13	14	27	15	16	31	58
3. With Bleeding	17	18	35	19	20	39	74	21	22	43	23	24	47	90
4. With Daze/Semi consciousness/ Unconsciousness	25	26	51	27	28	55	106	29	30	59	31	32	63	122
Fever > 7 Days	33	34	67	35	36	71	138	37	38	75	39	40	79	154
2. Cough with or without fever														
< 3 weeks	41	42	83	43	44	87	170	45	46	91	47	48	95	186
> 3 weeks	49	50	99	51	52	103	202	53	54	107	55	56	111	218

Screenshot of Form S page 1

12.4 Appendix D – Excerpt from DSIP exported file

The name in export	The value of the element	The text in DSIP	Coordinate in DSIP
CMLF1	Number	Fever >7 days / >5 y Only fever	1x1
CMLF2	Number	With rash	1x2
CMLF3	Number	With Bleeding	1x3
CMLF4	Number	With Daze	1x4
CMLF5	Number	Fever < 7 days / > 5y	1x5
CMLFTotal	1+2+3+4+5	SUM	1x9
CMGF1	Number	Fever >7 days / <=5 y Only fever	2x1
CMGF2	Number	With rash	2x2
CMGF3	Number	With Bleeding	2x3
CMGF4	Number	With Daze	2x4
CMGF5	Number	Fever < 7 / <= 5y	2x5
CMGFTotal	1+2+3+4+5	SUM	2x9
CMLGFTotal1	CMLF1 + CMGF1	SUM	3x1
CMLGFTotal2	CMLF2 + CMGF2	SUM	3x2
CMLGFTotal3	CMLF3 + CMGF3	SUM	3x3
CMLGFTotal4	CMLF4 + CMGF4	SUM	3x4
CMLGFTotal5	CMLF5 + CMGF5	SUM	3x5
CMLGFTotal	CMLFTotal + CMGFTotal	SUM	3x9

12.5 Appendix E – Mail to Indian coordinator describing the current state of exporter

Hello

I just thought I would write some things to you that might help if the software wont work.

-Make sure the user defined in Doc&set/yoiourname/DHIS/hibernate.properties has access to the database

-Make sure the database name is ke or change it in the hibernate.properties file

You log in with hisp/hispindia
The OrgUnit which has data is Kerela State
Period is Jan 05 (01.01.05-31.01.05)

When you try to export go to advanced and you have to put any element in the Data elements list. There has to be one added or it will not work. Just choose anyone.

You will have to change the name of the file to values.xls before you press the export button. Or you will have to rename it once it is done.

When you choose which export type to use you have to select the one called "I18.string*($value)" or something. Thats the name because the Internationalization files arent updated yet.

I wish we had more time to set it up, but hopefully you will be able to set it up :)

The two first sheets in the Excel files will only show 0s because that dataset isnt used yet. Show them the rest. The last one is supposed to be empty as I undertand it.

I will not be at the internet cafe much longer. We only went to a local one because My travel partner wasnt feeling well.

Good Luck

Stian

12.6 Appendix F – Mail about MS Excel solution not working

Hello guys!

I have some bad news about the Excel file created by the DSIP exporter; it will not import into DSIP. I tried fixing this a week now. I have no idea what is wrong. It fails somewhere in the compiled VB code in DSIP (I think) so I cannot see what causes the error.

The error I get is:
Run-time error '91':
Object variable or With block variable not set

The error comes from Access I believe. I have tried to Google it and found some places where they ask for it and they solve some of the errors in the example. But I have no idea how Access code works and I cannot see the code in DSIP.

Everything but one of the fields gets imported into the database and that is the Idno, the id number of the OrgUnit. Every other value seems to import okay even though it crashes. I have gotten it to accept the id number also, but it has been shorter than the value needed for a legal number for id's that don't start with 0. I have also tried setting the value to 0. Once imported in the database it says 00.

It really bugs me that it does not work, but since I haven't gotten it to work until now I ask for another solution to solving the DSIP export. It is possible to import lots of types into Access databases. I think I will be able to pretty quickly be able to export to a CSV or DBF format which can be imported directly into Access. That means that at the state(or block) level they have to import this way instead. I can write a guide which explains in detail how this is done and there arent many steps (I am in the process of writing a guide for using the exporter also), but they HAVE to have Access installed.

Is this a possibility? Or do you have any other tips solutions? One other way to do it is insert into RHIS, export to CSV at PHC and they import into DSIP and export, it is quicker than entering into DSIP and exporting at least.

Stian Strandli

PS. Can we also be added to the Indian mail-list (if we are not allready)?

> I have some bad news about the Excel file created by the DSIP

This should say RHIS, the file created with Java and the POI framework. DSIP exports works perfectly in the import function.

12.7 Appendix G – Mail to the developers mailing list

Hello people!

We have been working on the DSIP exporter for a while now and it is hopefully ready soon. I have created a manual for the users and a small Java program that helps the user set up the mapping files needed.

It would help me a lot if some of you could take a look at either one of them. Both files is located in a zip file (350 kb). The manual is a MS Word file. The program can be started by starting the bat file. Try to be a little rough with the program.

Link: http://student.hiak.no/~stian/IDSP.zip

Any thoughts or comments is appreciated!

Thanks in advance

Stian Strandli

12.8 Appendix H – Mail about new state of DSIP exporter

Hello guys!

I send this privately to you guys so it seems more personal and hopefully I will get your attention. We still haven't gotten put on the Indian mailing list which would be nice. Now to the point.

I have altered the DSIP exporter again, so that it does not need the setup tool. I have also reduced the values needed by the exporter from 7 to 4. The exporter now also injects the data directly into the DSIP Access database so there is no need for the person to import a XML file manually, which makes it easier. Now the path to the database is also needed so the number of elements needed is 5 (instead of 4).

I have changed it so it can find these values from a data set within RHIS. This is maybe not a "right" way to do it, since the data sets aren't supposed to keep this sort of data (am I right?). The data set has to be called DSIP Setup and you have to enter these data once. I have revised the manual to fit these new tasks and attached it to the mail.

How is this solution? Is this doable? Is it possible for the dataset to be added by someone technical? Is it possible for someone technical to add the DSIP information?

Can this mail also be forwarded to someone working in Kerela, so they can say what they think about it?

My travel partner has just left Bhopal and is on the train to Delhi to meet his parents. So I am here alone in Bhopal with main developer, another developer and facilitator. Depending on what happens My travel partner might stay in Delhi for up to two weeks and come back here id the Delhi office is still not ready.

Regards
Stian Strandli

12.9 Appendix I - Mail to Indian mailing list about final state

Hello everyone!

I am a Norwegian student working on the RHIS 2 software. Me and one other Norwegian student are currently in Bhopal, M.D., India. We will stay in India until the end of May. We have been working on an DSIP export module for some time now. Now it is hopefully ready for use. There has been many changes along the way, but now we have a solution where data is directly inserted into the DSIP database and then they have to be exported from DSIP itself. If anyone have good knowledge about how the DSIP application is used I am very interested in that. Different versions? How they use the forms? Does everyone report upwards every week? Etc.

I have written some documentation no how to use it and I was hoping some of you can look through it and see if it well enough or if it needs more info, more pictures or anything else? I have made two different documents; one for the actual user and one for the facilitator setting up the system. I have also written a small setup tool that can be used to set up the needed stuff. Both are explained in the document.

Here is a link to a zip file with both documents, they are both MS Word:
http://student.hiak.no/~stian/Manual.zip <http://student.hiak.no/%7Estian/Manual.zip>

Please give me your honest opinion!
Thank you.

Best regards
Stian Strandli
stianstrandli@gmail.com <mailto:stianstrandli@gmail.com>
+919971445676

12.10 Appendix J – Final version of DSIP export manual/FAQ

DSIP export setup manual

For the RHIS software

This document is written by Stian Strandli for PHIS India

How to include the DSIP exporter to the war file

The source code for this project will not be put together with the other source code. Before you build the system you will have to edit the pom.xml file located in the dhis-web-portal folder. Under the dhis-support-hibernate dependency insert:

```
<dependency>
  <groupId>org.hisp.dhis</groupId>
  <artifactId>dhis-service-export-idsp</artifactId>
  <version>1.0</version>
</dependency>
```

This will make Maven download the DSIP exporter module and include it in the war file. Now the DSIP exporter will register itself together with the advanced exporters.

Installing the DSIP exporter with tool

If the version you are using is M7 or newer, then the exporter should already be installed. This means that what you have to do to make this exporter work properly is to configure what should be exported. Not all the same data you enter in DSIP is found in RHIS so this has to be set up properly. This document will help you with these steps.

Step 1
Find the DSIP setup start file located in

Step 2
Start the setup by double clicking the setup .bat file. A window will open.

Step 3
This program will show you a list of Indian setup programs. DSIP may be the only one or there might be others there. Enter the number represented by DSIP and press the enter key.

Step 4
This will start the setup itself. You have to enter some details about DSIP. Here is a small table that show which attributes you need and how they should look:

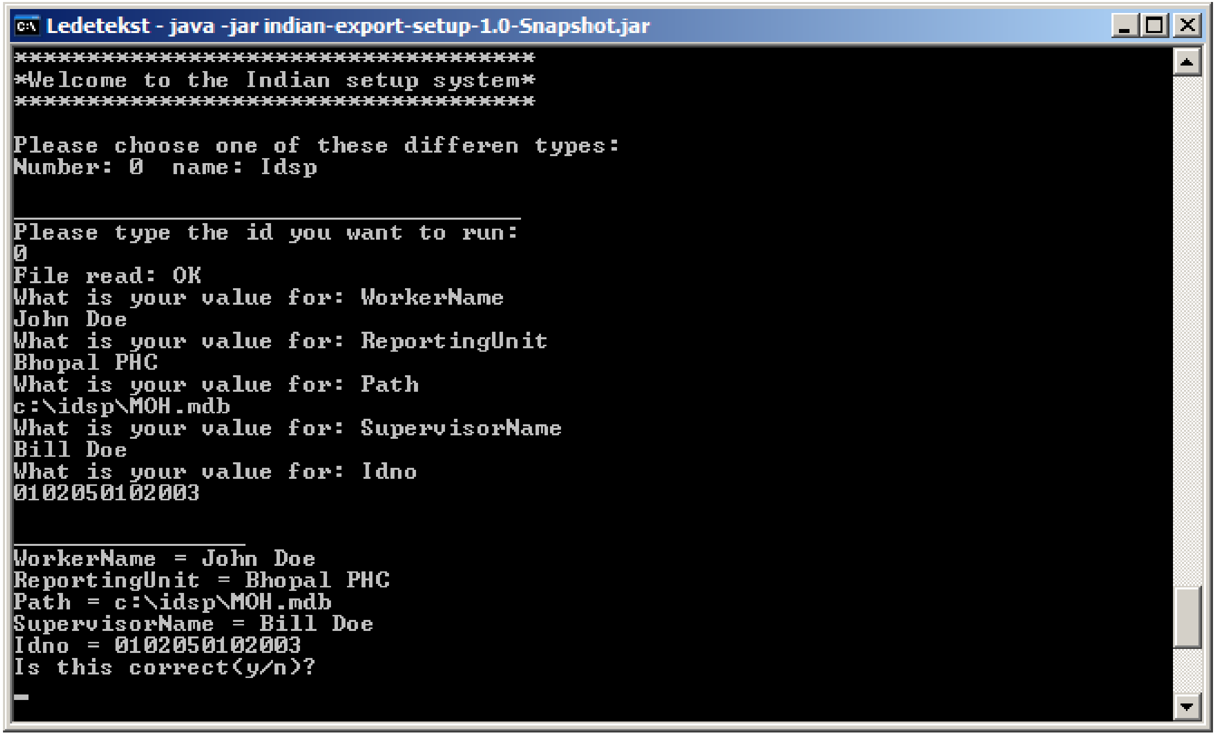
Name	**What it means**	**What it should be**
WorkerName	This is the name of the worker who enters DSIP data. This is also entered into DSIP when a form is entered.	Like a name, only text and not too long.
ReportingUnit	The name of your reporting unit. This has to match how this was entered in DSIP. You can find the name of your unit in the form entry.	Like a name, only text and not too long.
SupervisorName	The name of the supervisor in the reporting unit. This is also a value entered every time data is entered into DSIP.	Like a name, only text and not too long
Idno	This is the identification number of your reporting unit. This can be found in the data entry page in DSIP after you have selected your reporting unit.	This is a number. It should be 13 characters in length: 3501020503002
Path	The path of the database file located in your DSIP folder. If you have trouble locating your file. See the FAQ at the bottom of the manual.	A path should look like this: "*Driveletter:\folder\folder\databasename.mdb*" . One example is: C:\idsp\MOH.mdb

After you have entered all this information, you will be prompted with it and you have to select if this information is right or wrong. If it is correct you press y or if it is wrong you press n and the enter key.

For the exporter to be able to function, this data has to be correct. If you are unsure of any of the elements contact someone who knows the DSIP software or someone in the PHIS organisation.

If everything is set ok the setup program will tell you so. If you don't want to set up anything else you just press enter when you are done and the program will exit. Now you are ready to use the exporter via RHIS. Take a look at next page to se how it is done.

Here is a picture of the tool running:



```
************************************
*Welcome to the Indian setup system*
************************************

Please choose one of these differen types:
Number: 0  name: Idsp

________________________________
Please type the id you want to run:
0
File read: OK
What is your value for: WorkerName
John Doe
What is your value for: ReportingUnit
Bhopal PHC
What is your value for: Path
c:\idsp\MOH.mdb
What is your value for: SupervisorName
Bill Doe
What is your value for: Idno
0102050102003

________________
WorkerName = John Doe
ReportingUnit = Bhopal PHC
Path = c:\idsp\MOH.mdb
SupervisorName = Bill Doe
Idno = 0102050102003
Is this correct(y/n)?
_
```

Installing the DSIP exporter without tool

What the tool does is create the files needed for the export to work. They have to be created either by running the tool or trying to do an export in RHIS. If you try to do an export it will fail because mainly the path to the database is not set and the right values are not set. The property files are located in the user home folder in the subfolder RHIS. On windows this means Documents and settings / username / RHIS. There are two files for the DSIP exporter, one contains which data set that should be looked up and the other contains all the value mappings. The file you have to edit is: IdspDataValue.properties.

You can open the file in WordPad by right clicking and choose open. Choose WordPad on the list that shows up. The line numbers you will have to edit are:
1. Insert the path after the #, see the example in the file

6. WorkerName is the field name. Change the name of the DSIP worker inside the ¤'s. Make sure the name is surrounded by ¤'s.

7. SupervisorName is the field name. Change the name of the DSIP supervisor inside the ¤'s. Make sure the name is surrounded by ¤'s.

8. ReportingUnit is the field name. Change the name of the Reporting unit from DSIP between the ¤'s, make sure you spell it exactly the same as it is in DSIP. Make sure the name is surrounded by ¤'s.

9. Idno is the field name. Change the value to match the id number this reporting unit has in DSIP. The number should be 13 digits long. If the first number is a 0 you still have to enter it.

165. Idno is the field name. Change the value to match the id number this reporting unit has in DSIP. The number should be 13 digits long. If the first number is a 0 you still have to enter it.

217. WorkerName is the field name. Change the name of the DSIP worker inside the ¤'s. Make sure the name is surrounded by ¤'s.

218. SupervisorName is the field name. Change the name of the DSIP supervisor inside the ¤'s. Make sure the name is surrounded by ¤'s.

219. ReportingUnit is the field name. Change the name of the Reporting unit from DSIP between the ¤'s, make sure you spell it exactly the same as it is in DSIP. Make sure the name is surrounded by ¤'s.

220. Idno is the field name. Change the value to match the id number this reporting unit has in DSIP. The number should be 13 digits long. If the first number is a 0 you still have to enter it.

Check over the changes you have done and save the file. Please look at the table in the last page to see what the possible values for each element are. The exporter should now be ready.

Using the exporter

See explanatory picture in the middle of this particular guide.

Step 1

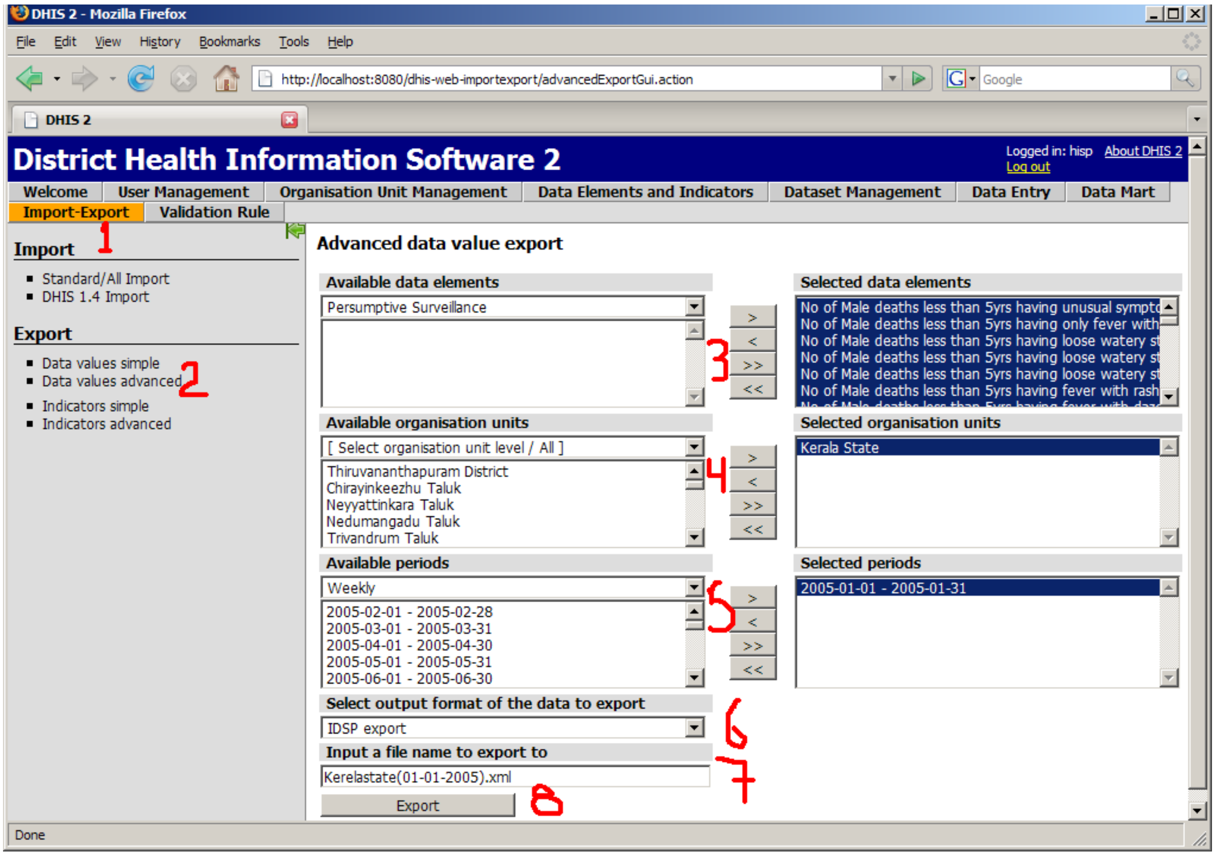
The exporter is located under the Import-export tab. When you click this tab you will get a new menu on the left side. Here you will find a link called Advanced Export, click it. Now the program will show you a new window.

Step 2

Here you will have to select the data set Presumptive Surveillance and Syndromic Surveillance and add all the data elements for each of them. By clicking the >> button you choose all the elements from this data set. Now all the elements will be listed in the right panel.

Step 3

Next we have to select which organisation unit to export from. Select you organisation unit and click the > button. Your organisation unit will now be listed in the right panel. Only select one organisation unit; choosing multiple will corrupt you export.



Advanced data value export

Available data elements: Persumptive Surveillance

Selected data elements:
No of Male deaths less than 5yrs having unusual sympto
No of Male deaths less than 5yrs having only fever with
No of Male deaths less than 5yrs having loose watery st
No of Male deaths less than 5yrs having loose watery st
No of Male deaths less than 5yrs having loose watery st
No of Male deaths less than 5yrs having fever with rash

Available organisation units: [Select organisation unit level / All]
Thiruvananthapuram District
Chirayinkeezhu Taluk
Neyyattinkara Taluk
Nedumangadu Taluk
Trivandrum Taluk

Selected organisation units: Kerala State

Available periods: Weekly
2005-02-01 - 2005-02-28
2005-03-01 - 2005-03-31
2005-04-01 - 2005-04-30
2005-05-01 - 2005-05-31
2005-06-01 - 2005-06-30

Selected periods: 2005-01-01 - 2005-01-31

Select output format of the data to export: IDSP export

Input a file name to export to: Kerelastate(01-01-2005).xml

Step 4
Now you have to select which period you want to export from. First select Weekly in the first box. Now the system will only show you the periods used to enter DSIP data. Select the right period and press the > button to choose the selected period. You can only select one period; choosing multiple will corrupt your export.

Step 5
Choose the "DSIP export" from the box where it says "select output format of the data to export".

Step 6
Press the filename box and remove the text. Write instead the name of your PHC and the today's date followed by ".txt". This will text file will contain a report of how the export went.

Step 7
Take a look at your choices. If everything is correct, then press the export button and download the file presented to you. Save it somewhere where you can easily locate it. If you do not know where the file is stored, then the file will probably be located at you desktop.

Step 8
Locate the file exported and double click it. It will say if the export was okay. If there is a problem that you don't understand contact a developer.

Frequently asked questions

Question: I made an error in the files in user home?
Answer: Just delete both the IdspDataValue and the file IdspDataSet. You will now have to run the setup tool again and these files will be created and set up configured.

Question: I deleted the files in user home?
Answer: See the previous question.

Question: The export will not start, what can I do?
Answer: Are you sure you have selected everything like it says in the guide earlier in this document? Have you chosen the right exporter in the menu on the bottom? Have you edited the configuration files lately, then see question 1.

Question: The program shows me a white page with an error, what do I do?
Answer: Contact your PHIS contact with information on how this happened. See other questions on how to solve errors.

Question: Why isn't form L being exported?
Answer: This form is not yet implemented in RHIS. The exporter software needs to be upgraded to handle this form.

Question: Why is mapping for element 22/21 wrong in Form P?
Answer: There was an error in the data set, so the mapping fixes this error. If the error in the data set is corrected, the mapping also has to be fixed. Contact PHIS developers.

Question: Why can I not import a file?
Answer: This functionality has not been prioritized. The export function is more important. This functionality might come in a later version of RHIS.

Question: My question is not being answered here or the answer does not help me, what do I do?
Answer: Contact you closest PHIS contact or a developer in PHIS.

Question: How do I find the path to my DSIP database?
Answer: There are multiple ways of finding out. The easiest is opening My Computer and double clicking your C drive. DSIP may have been installed directly to your C:\ drive. If it not installed here you can double click you Program Files folder and see if you have an DSIP folder here.

If DSIP is not installed in any of these folders you can find the link you use to start DSIP. Then you right click the icon and select properties. There you can find the “target” line. It will say where the start file is located. Open this folder and see if you find the MOH file there. Now you can copy the address from the address bar and copy it into DSIP and add MOH.mdb at the end. You will en up with something that look like this:
C:\DSIP\MOH.mdb
C:\Program files\DSIP\MOH.mdb
C:\NHRM\DSIP\MOH.mdb
D:\DSIP\MOH.mdb

Question: The exporters jar file will not be downloaded from the RHIS repository with Maven, what can I do?
Answer: Download the DSIP-exporer-jar.zip from hispindia.org. This file can be extracted into your local Maven repository located in user home /.m2. Try to run the builder now.

12.11 Appendix K – Link to Confluence Documentation

The page could be reached at:

http://hips.ifi.uio.no:8080/display/DHIS2/IDSP+Export+Module

12.12 Appendix L – First mail about settings module

Hello people!

I have some questions about how to do the user customization of the web interface.

There is many ways to do this:
- Let every user define what he wants through its user object
- Let there be one default style and users can have their own
- Let supervisor (i.e.) define how it would look for everyone

How to store the changes:
- Define in a new .css file which can be included at the end of all the tweaks and hacks
- Define one file for each user based on its name? Where should these files be stored?
- Define the css in a String or some kind of object in the User class and add a new <style> and run through it after the tweaks and hacks

What can be changed:
- Background colour (same everywhere? Not black?)
- Font colour (same everywhere? what about links?)
- Background picture (on first welcome page or soft back on all?)
- Background colour in main window (Maybe not all colours?)
- Text on the top (loaded from where?)
- Anything else?

I guess there is no trouble just changing colors after all the tweaks and hacks are run? Or do these changes have to be incorporated in the original css file?

I think what Knut said about a logo or a picture as a soft background would be a cool feature. I don't know how to do it though, but I guess it can be done in css somehow?

If there is changed based on user name stored in the object itself or in the database, then the login page will not be changed at all. Is this an issue? I think this is the easiest way to implement it. The possibility to change the text shown on top is also an issue. Should this be read from a file also or from the object. Maybe there should be created a "settings" class which can contain this info?

Then there is the picture. This should be uploaded and stored where in the file structure? This could be stored in the database also and shown from there. I have done this via a servlet before. I don't know if this is possible with an Action?

Any other thoughts on this? I know these things may seem kind of irrelevant but I don't want to do it the "wrong" way and have to do it all over again.

50
51 Regards