UNIVERSITY OF OSLO
Department of Informatics

Application of Open Source GIS in District Health Information Systems

Master thesis

Lars Gunnar Vik and
Trond Andresen

24th May 2005

Abstract

This thesis considers issues surrounding use and development of open source geographical information systems (GIS) as a part of the district health information systems in developing countries. The base for this thesis is field work done by the two authors in three different states in India in cooperation with the Health Information System Program India. The thesis looks into the general concepts of open source and geographical information systems in the Indian context, and present a GIS application created for a district health information system. Issues concerning open source GIS are discussed with observation done by the two authors. Based on the discussion the thesis will conclude that some reflection in the discussion may be of interest for developers of open source GIS.

Index

1. Introduction

1.1 The implementation of open source GIS in DHIS

This thesis will consider issues surrounding use and development of open source geographical information system (GIS) as a part of the district health information system in developing countries. The base for this thesis is field work done by the two authors in three different states in India in cooperation with the Health Information System Program India (HISP India).

The main objective of the field work was to create an open source GIS system that could be integrated with the District Health Information System (DHIS), a management system for supporting the district based health systems. The two authors where heavily involved in all stages of the development.

The thesis will look in to the general concepts of open source and geographical information systems in the Indian context. We will also present the finished GIS application that were developed, and explain the different solutions in the system.

The development of the GIS application was influenced by many factors. Some of these issues were related to technology, infrastructure and programming skills. We will look into these issues, and explain how they affected different aspects of the project. With our observations as a starting point, we will then discuss aspects of open source development and GIS technology for use in public health care.

1.2 Motivation

The software development was conducted in south India by the authors in the beginning of 2005. We wanted to do the development work in India in order to have a closer contact with people from the Indian HISP team, health officials and health workers. The motivation for this was twofold. First, we wanted to get closer to the context where the software was going to be use. We needed knowledge about the Indian health system, information about the end users existing technology and infrastructure and the skills and needs of health personnel. Altogether this would constitute the applications functional and non-functional requirements and its operational constraints. Second, it was important to create continuity in the development and life cycle of the software. The Indian HISP members are both doing system development and implementation and training of software. When we, the authors, would leave the project, it was important that the Indian HISP team could benefit from our participation. By involving staff in the development process, the shearing of knowledge, and training of developers, was thought to be easier and more efficient, making it easier for HISP developers to continue the development process.

1.3 Chapter presentations

This thesis is divided into seven chapters. In chapter two, we review literature concerning health information systems, GIS and open source.

The third chapter introduce the health information system program (HISP) and the district health information system (DHIS).

Chapter four contains the observations we made during our field work in India.

The fifth chapter reviews the GIS application we made in India. The chapter includes information about the requirements of the application, and a review of its functionality.

Chapter six contains the discussion part. Here our observations and experiences are taken into consideration as we look at literature about open source and geographical technology.

The last chapter summarizes the thesis with a short concluding remark.

2. Literature Review

2.1 Health Information System (HIS)

A generalized way of viewing HIS, is to think of it as a generic label for different types of information systems used in health care. For instance a HIS may include computer equipment, procedures and personnel designed, constructed, operated, and maintained to collect, record, process, retrieve and display information specific to the health domain. (CEN, undated)

Heywood, Campbell, and Awunyo-Akaba (1994) argue that it is important to look at the HIS as a tool for improving health care, and not as a solution itself. A HIS will improve the health care by improving health service and effectiveness and efficiency through better management at all levels of health service (Lippeveld and Sauerborn, 2000)

"The ultimate objective of health information systems is [...] not 'to gain information' but 'to improve action'" (ibid.)

In this thesis we will use the term HIS on the routine health information system (RHIS). Non routine health information systems are typically complex clinical health information systems that focus on patient data while the RHIS' focus on; "*“Information that is derived at regular intervals of a year or less through mechanisms designed to meet predictable information needs”* (RHINO 2002)

Examples of routine health information systems are (ibid.):

- health service statistics for routine services reporting and special program reporting (malaria, TB, and HIV/AIDS)
- administrative data (revenue and costs, drugs, personnel, training, research, and documentation)
- epidemiological and surveillance data
- data on community-based health actions
- vital events data (births, deaths and migrations)

2.2 Geographical Information Systems

"A geographic information system (GIS) is a computer-based tool for solving spatial problems. A GIS integrates information in a way that helps us understand and find solutions to problems. Data about real-world objects is stored in a database and dynamically linked to an onscreen map, which displays graphics representing real-world objects. In general terms, a GIS can be defined as a computer-based information system that enables capture, modeling, manipulation, retrieval, analysis, and presentation of spatially-referenced data." (AFT Project, 2003)

"In its simplest form GIS can be described as a map on a computer drawn by simple lines and symbols representing formation in the landscape." (Mark et. al., 1997)

The history of GIS started in the 1950's and was developed mainly for public sector. In the beginning the definition of GIS was very vague, and a big part of the GIS evolvement has been to clearly define GIS:

"One reason why it can be difficult to agree on a single definition for GIS is that various kinds of GIS exist, each made for different purposes and for different types of decision making. A variety of names have been applied to different types of GIS to distinguish their functions and roles."(Foote et al. 2000)

One way of categorizing GIS, is to use the three terms *map*, *database*, a *spatial analysis view*s. These terms reflect the ways in which geographical information systems is used by the GIS community. The map view focuses on cartographic aspects of GIS, emphasizing the ability to produce high quality maps and charts. The database view of GIS emphasizes the importance of a well-designed database system, in which complex analytical operations for many types of geographical data can be preformed. The spatial analysis view focuses on analysis and modeling. GIS is here viewed as a spatial information science (Maguire, 1991).

2.3 Geographical Information Systems applied to Health Information Systems

GIS has many possibilities in public health and can be a powerful tool for analyzing health data (Chung, Yang and Bell, 2004). It can be used for exploring spatial data, analyzing coherence between spatial data or simply visualizing map based reports.

Cromley and McLafferty (2002) distinguish between four different types of spatial analysis for health analysis: Measurement analysis, topological analysis, network analysis and spatial data analysis.

- Measurement analysis allows the user to calculate distances between points, polygons and areas. This is important for looking at interactions between people and places.

- Topological analysis is used to describe and analyze spatial relationships. It could for instance be interesting to find out how many hospitals that is located within one kilometer of a river. Or the how far a new housing area would be from major health facilities.

- Network analysis - investigates flows through a network. A simple, but important, example is determining the distance from one place to another through available roads. (How far is it for a vehicle to reach a destination?)

- Spatial data analysis - is closely tied to spatial statistics. This is functions for performing traditional spatial statistics, for instance finding nearest neighbor and doing interpolation and correlation.

The measurement and topological analysis functions are generally present in all GIS', network and spatial data analysis are often available only as functions developed outside GIS packages (Cromley and McLafferty, 2002).

In a review of GIS software used in public health, Chung, Yang and Bell (ibit.), concludes that most GIS packages, such as ArcGIS or MapInfo, not allows the users to do spatial analysis. The GIS studies in focus had to leave the GIS system to use other programs to make meaningful spatial statistical analysis. Most tools had only implemented functionality for visualization and mapping.

Chung, Yang and Bell (2004) think implementation within GIS of spatial analytical tools will increase the use of advanced spatial GIS in health studies. They found that studies in health field only use GIS as a data organizing system or for visualizing of results.

2.3.1 Getting data into GIS

Getting data into GIS is a work dependent on the raw data. The desired result is most often a link from the data to a latitude and longitude. An increasingly popular tool is geo-coding, which translates location information into geo-referenced values (Chung, Yang and Bell, 2004). Examples of source locations could be postal codes, specific addresses or roads.

One problem with health data that is mentioned by Chung, Yang and Bell (2004) is that health data very often is aggregated count data. The loss of precision when aggregating data makes it less suitable for geographical analysis

2.3.2 Processing data

After the data has been gathered and refined it is time to start processing it. There are four major techniques in the data processing stage: spatial analysis, spatial statistical analysis, smoothing and cluster detection statistic and spatial autocorrelation and regression.

It is necessary to distinguish between the first two: spatial analysis and spatial statistical analysis. Spatial analysis is the study of spatial pattern using basic GIS operations like spatial query and join, buffering and overlaying. Spatial statistical analysis on the other hand is the use of statistical theory on spatial referenced data (Chung, Yang and Bell, 2004).

Smoothing methods are used to eliminate variance instability in the data. Observed rates are often extreme when the population is too small (rural areas), or a disease is rare. The goal is then to control high variance in such cases. There are several different methods for smoothing data. Three popular methods are kernel density smoothing, empirical Bayes smoothing and locally weighted regression. Kernel density smoothing normally aims at creating a continuous map from data referenced as points on a map. Using an inverse distance weighting function, a grid is created between the data points. The weighting of the function can be customized to fit the variance in value and number in the dataset. Empirical Bayes smoothing is often used in disease mapping literature (Chung, Yang and Bell, 2004). The rates are then assumed to be binominal random variables. The method consists of shrinking the observed rates *differently* toward the mean of the distribution based on the number of observations in the small areas (Chung, Yang and Bell, 2004). In other words: in areas where the population is small, the observed values are normalized by smoothing them towards the mean value. Whereas in areas with a large population the smoothing is little and the observed value is kept very much as it is. Locally weighted regression is different from the two other methods because it does not have any parameters. "In locally weighted regression, points are weighted by proximity to the current value in question using a kernel. A regression is then computed using the weighted points" (Cohn, Ghahramani and Jordan, 1996).

After smoothing the data health analysts will be interested in identifying a pattern of clusters. Spatial clustering of health events is often caused by a common factor. This can be a dangerous road crossing, polluted drinking water or an outbreak of a disease. Finding such places of unusually low or high risk can be done by GIS software. By combining information about the involved population, such as age (a risk factor), average incidents for a larger area nearby and other relevant factors, a GIS application can automatically scan an area for clusters.

There are two kinds of spatial clustering (Cromley and McLafferty, 2002). One is where the search window is of a small size and looks for local differences. These are normally caused by local factors such as the examples above - a dangerous road cross, polluted drinking water or an outbreak of a disease. Another type of spatial clustering is when the window is large and covers cities or large regions. In that case the differences might be caused by climate or culture.

The last spatial analysis tool we will comment is spatial autocorrelation and regression. These statistical methods look for associations in data. For a given health indicator it would be interesting to know if there is a relation between the measured value and surrounding elements. To find such relations to neighboring values, or a given variable, one can use spatial autocorrelation or regression.

2.3.3 Visualization and mapping

The advantage of GIS is that it enables the user to visualize and explore health data interactively (Cromley and McLafferty, 2002). One traditional way of doing this in GIS is by combining table data with map data. One view is then showing a table with health data and a map linked to the table. The user can then either select rows from the table, and does queries about them, and get the result (locations) highlighted on the map. Alternatively, the user can select an area on the map, do a query about it, and get the results listed in a table. These are very simple GIS functionalities, but can be quite helpful for health analysts as it enables easy exploring of data.

Visualization and mapping can very well be used for analysis of geo-referenced data. Both as a report tool for the spatial analysis techniques mentioned above and for manually inspecting variables across space. But it does not address the spatial relationship of the data itself (Chung, Yang and Bell, 2004). Statistical techniques are required in order to quantify relationships and expectations for spatial data.

2.3.4 Pitfalls for GIS in public health

Richards et al. (1999) however emphasize the importance of not only having knowledge about spatial and statistical techniques, but also about epidemiology. Principles from epidemiology must be understood when formulating study questions and testing hypotheses about cause-and-effect relationships. Not paying attention to epidemiologic knowledge could cause a series pitfall for the user. A different pitfall lies in GIS' ability to quickly, and convincingly show the results of a complex analysis (Richards et al., 1999). The reason for this is the many possibilities for misunderstanding or misusing simple results created from very complex data. One example of misuse can be altering of data. This could be difficult to discover in a graphical computer GIS environment. Also the graphical visualization of queries in GIS is powerful and can give the user a false impression of correctness. As reports are created, one is often confronted with the need of simplification in order to obtain clearness. Suppressing details selectively to help the user see what needs to be seen, could suggests conclusions not supported by a careful epidemiology analysis (Richards et al.,1999).

2.3.5 Advantages of GIS in public health

Richards et al. (1999) emphasize on five advantages GIS could have in public health:

1. Improve the ability of practitioners, planners, and researchers to organize and link datasets

2. Encourages the formation of data partnerships and data sharing at the community level

3. Provide public health practitioners and researchers with several new types of data

4. As new GIS methods are developed, they can be added to the "toolkits" of epidemiology

5. Help community decision makers visualize and understand a public health problem

The first advantage is argued by geography's ability to provide “a near-universal link for sorting and integrating records from multiple information sources into a more coherent whole” (Richards et al.,1999). The advantage of this is obviously that health practitioners can get more information about a case. This makes it easier to both find causes to a health problem, and plan counter measures. The thought is that more information makes planning easier and more effective.

Most departments within a community are in need of information that data cross different departments and divisions. This will encourage partnerships and sharing of workload. Production of data can in many cases be done many times by different people. This redundancy could be avoided by sharing data. Also, much of community data is non-sensitive data, and could be stored centrally, available to all departments.

The cooperation with different departments gives opportunity for creating new types of data. This data is different from plain health data because it is generated based with the help of information from a variety of departments. Also, the geographical information technology itself is a source for new data by encouraging new ways of working. When health workers get new ways of exploring health data, they might find new interesting connections.

The new working methods, technical tools and GIS methods constitute new methods that can be added as important resources for epidemiology (Richards et al.,1999). New GIS methods can also be developed by the public health community and shared openly as toolkits for the health analysts.

The last advantage Richards et al. (1999) talk about, is the power visualizing: "GIS technology can be an extremely effective tool to help community decision makers visualize and understand a public health problem. In addition, action is more likely when the decision maker can see on a map that a problem is occurring in his or her backyard." The power also lies in the ability to provide quick responses to questions from the community – or even more important: being able to display the answer in an easy to understand way, through maps and illustrations. GIS and computer aided presentations, also enables useful interaction with the community by being able to answer questions right away (understandable to the common man or woman).

2.3.6 Limitations of GIS

Richards et al. (1999) offer six points on the limitations of GIS from a public health perspective. These are:

1. A lack of GIS software for community health planning
2. Communities lack basic street-maps needed by GIS
3. The cost of training and support is high
4. The cost of maintenance and upgrades

That fact that there is very little software for GIS based health planning makes it hard to integrate GIS with community work. Richards miss the ability to link GIS technology with community planning tools. Data entry forms and procedures must be present in order for GIS to be usable and helpful to health practitioners.

Maybe even more important is the need for geographical basis data. This includes street maps and topological maps. “Without an up-to-date base street map [...] a public health practitioner investigating a disease outbreak may have to spend considerable extra time and effort to digitize the locations of cases or may not be able to map all case reports” (Richards et al.,1999). Obtaining good maps is very costly. In a report conducted for the Squamish Nation in Canada, map data expenses was pointed out as the major cost for most GIS projects (Calla and Koett, 1997). It constituted 60-80% of total cost.

In addition to this comes the cost of training and support. “Practitioners, planners, and researchers, and especially state and local public health department staff, need training and user support in GIS technology, data, and epidemiologic methods in order to use GIS technology appropriately and effectively” (Richards et al.,1999). The cost of this training can be a burden. Before expertise has been acquired, training must be bought from commercial GIS vendors.

One more cost factor is maintenance and upgrades. This is closely linked to computer technology and its frequent releases and ever more demand on hardware performance. “GIS software continues to evolve rapidly; typically, a new iteration (or upgrade) is released about every 18 months. Current prices for some GIS products (in particular, for Web-enabled GIS applications and for neighborhood lifestyle segmentation datasets) remain a potential barrier (running as much as $10,000 or more).” (Richards et al.,1999).

The last two points on the list concerns the maturity of GIS, both as a general toolkit and in epidemiology especially. The first is about web enabled GIS, and the possibility of performing spatial statistical analysis over the web. “Full GIS capability on the Web is a considerable technical challenge because GIS software has only recently started to be developed using Web-accessible programming languages, and the size of GIS map images and data files can be large and significantly slow access and display functions over the Web” (Richards et al.,1999).

The second maturity issue is about confidentiality. The confidentiality epic individuals and households must be guarantied by GIS software. Richards et al. (1999) are concerned of whether the databases will keep confidentiality as they are linked within a geographic information system.

2.4 Participation

The term participation is used in information technology (IT) to describe the co-operation between end-users and system developers in the design, implementation and use of IT systems. Participation has evolved as an important research field due to many failures caused by an over focus on the technological aspects of system solutions.

There have existed several kinds of participatory approaches in the systems development research field (Puri, 2003b). We do not intend to move into this debate, but we would like to look at some issues relevant to the Indian context. S. K. Puri (2002b) offers some interesting thoughts on participation and GIS. In his Phc thesis and in three articles he looks at GIS and land management in India. He focus on experiences made from cases in India and discusses the future challenges. In his cases, the motivation for using computer based GIS is the increased land degradation and poverty in the rural areas.

According to Puri (2002a) computer based technologies for management of degraded lands in India, have been used since the early 90's and in these programs there have been four important requirements for the information systems:

- The ability to access spatial data periodically in a cost effective manner
- The ability to integrate spatial data with socio-economic data
- The ability to update data periodically and easily
- The ability to present and display spatial and non-spatial data in simple forms.

To meet these requirements, GIS has been promoted in India since the mid 90's. Geographical tools were considered for design, implementation and monitoring of some of the development projects.

The problem issues identified by Puri (2002) as the key problems for utilization of GIS are as follows:

Technical difficulties:

- Lack of technology awareness at the end user level in the field

Socio-economic difficulties:

- Scientific institutions adopting technology deterministic approach
- Lack of relevant socio-economic data
- Absence of "map-based culture"

Organizational difficulties

- Rigid bureaucratic structures
- Difficulties in achieving inter-departmental coordination

An important obstacle mentioned by Puri (2003b) are the problems caused by the top-down approach adopted within bureaucratic structures of the health institutions in India (Puri, 2003b). This approach makes it difficult for expert and users of the different departments to enter into a common dialog. A common dialog is needed because of the multi-disciplinary nature of the technology (Puri 2003b).

A different obstacle for enabling a common dialog is the cultural challenges. India is still highly marked by the cast system and a pronounced social grouping of people. Walsham (2000) has been looking at how this strict system motivates a strict society with well defined roles for each and every citizen. In a GIS projects this was reflected in the way the participants expected to fill a compartmentalized role and activity. The workers saw little need in looking outside the boarders of their specialized work tasks.

Walsham (2000) also mentions an interesting statement from an employee in a non-governmental organization operating at district level in India: The district level worker says he is interested only in dams, the agricultural scientist in soils and the forester in trees. Everyone says I am fine and no one sits and talks with each other. The statement witness a system that is very compartmentalized. Walsham (2000) says there is a mental barrier among the people.

Participation is also hindered by a lack of technology knowledge. GIS technology developed in western counties can be thought of as reflecting western values says Walsham (2002). The GIS technology might not be compatible with beliefs and attitudes in other cultures. The Indian problem is that the people not have a culture for using maps. Typical Indians will rarely, if ever, use maps in their daily life. Walsham (2000) mentions a situation where he was looking for a map that could help him find an institution. He asked why there was no such map, and got to know that in India, maps were not used for viewing institution areas.

Walsham (2000) think GIS are viewed as an alien in an Indian context. He means Indians do not in general use external conceptualization of space. They do not "think in maps" when they orientate themselves in an area. Some Indians feel discomfort when relating to a GIS map. Also the purpose of GIS reflects a sense of being able to control space and nature through technology. The need to dominate nature feels unnatural for many Indians. Indians typically see themselves as a part of nature rather than standing outside of it.

Walsham (2000) do however see a change the use of maps in India. There is a market increase in the production and availability of maps in India. Much due to import by foreign companies and agencies. In addition, Indian software houses are spreading knowledge about maps by supplying GIS services and software to foreign clients. Walsham (2000) think this increased focus on maps in parts of the Indian society will have a significant impact on Indians use of maps, and more generally, their conceptualization of space.

2.5 Open Source Software / Free Software

Open source- / free software (OSS/FS) is based on an ideology that gives the users and developers an alternative to proprietary system development. The idea of OSS/FS came up in the early 1980'ies when proprietary systems started to rule the development of software. As a counterweight to property development Richard Stallman in 1985 started the Free Software Foundation, an organization to braze the development of opens source software (Openknowledge.org, undated). The Free Software Foundation ideology was build on Richards Stallman's four rules for truly free software (ibid.). The user must be able to:

1. Run the program, for any purpose.

2. Modify the program to suit their needs. (To make this freedom effective in practice, they must have access to the source code, since making changes in a program without having the source code is exceedingly difficult.) They must have the freedom to redistribute copies, either gratis or for a fee.

3. Redistribute copies, either gratis or free.

4. Distribute modified versions of the program, so that the community can benefit from your improvements.

To protect the OSS/FS Stallman created the GNU General Public License (GPL).

"The central idea of the GPL is to prevent cooperatively developed software code from being 'enclosed' or turned into proprietary software."(GPL 1991 URL)

The idea of free licensing with distribution of source code inflict on project development. The organizations set up to support software development are different according to the project (OSS/FS or Propriety).

"The 'standard' way of organizing the production of software has been much like the standard way of building a complex industrial good: a formal division of labour that uses proprietary knowledge, guarded by restrictive intellectual property rights, enclosed within a corporate hierarchy, to guide and govern the process. But this is not the only way to organize the production of software. In the last few years another way of building software, the open source process, has gained notoriety just as the products of this process have gained market share across key segments of the information economy. In fact open source is not a new process. But it is distinctive, and the success of open source software projects demonstrate empirically that a large and complex system of code can be built, maintained, developed, and extended in a non-proprietary setting where many developers work in a highly parallel, relatively unstructured way and without direct monetary compensation."(Weber undated URL)

To day OSS/FS has a great market share, and according to surveys conducted by independent organizations it seems likely that OSS/FS popularity will further increase (Wheeler 2005).

2.6 Open Source in developing countries

"The digital era presents significant opportunities and real risks for emerging markets. One risk is that developing countries could be 'locked out' of the leading edge of the software that runs a digitizing global economy. The combination of Moore's law rapid increases in processing power at declining prices) and Metcalfe's Law (positive network externalities, meaning that the value of the network increases is proportionately as it grows) suggests that markets can grow intensively and dramatically within the developed world, without necessarily having to expand geographically at the same pace. As developed economies increasingly create networked purchasing and production systems that depend on advanced digital systems, countries that are not connected on favorable terms (and firms within those countries) may be deeply disadvantaged. International organizations and non-governmental organizations are increasingly computer-enabled as well, which means they will favor interaction with countries and organizations in the developing world that are similarly enabled and can interact effectively with their information and management systems."(Weber undated)

In order to get in to the technological race and to be a part of the network, developing countries need access to sophisticated technology with in a limited budget. Governments in the developing countries have recent years looked in to the potentials of open source, not just economical factors but also the capability to compete with proprietary systems. For example in Vietnam the government has done a major research on open source, and is now implementing a plan to make all government computers run on OSS/FS. Vietnam and countries like Korea, India and China are also planning that a quota of civilian systems is to be replaced with open source software. Some of the reasons for using open source instead of proprietary software were to;

"...reduce commercial software license fees, freedom from foreign-owned technology, greater security, curbing the number of infections from Windows-based viruses and to gain technological leadership on platforms relatively free of dominance by large multinational corporations."(CNET Asia, 2003 URL)

These reasons/motivations are explained in greater detail by Steven Weber (Weber, undated). He divided the motivations in to three clusters:

- Independence
- Security and Autonomy
- Intellectual property rights and productivity

Independence

The independence aspects of OSS/FS include the cost efficiency by stimulating domestic open source development, in stead of purchasing software from international merchandisers. A South African government council expressed foreign currency savings as an explicit rational for considering OSS/FS deployment, while the Taiwanese government estimated that a strategic open source project in their country could save the government nearly $300 million in royalty payments (ibid.)

OSS/FS represents an opportunity for countries to minimize their reliance on single suppliers who may not be focused on the countries interest, and represent a possible route for more domestic talent to participate in the development of local software (ibid.) This is related to nations interest in precisely where the expenditure on information technology is going (ibid.). To keep up, and not to become reliant on purchasing software, the OSS/FS thru participating local talent:

"[...] *holds the potential for expenditures on information technology to stay at home and contribute to a nascent local software industry. This brings with it the attendant prospect of upgrading indigenous human resource capacity and the technological skill base of the country."* (ibid.)

Security and Autonomy

Further on the use of OSS/FS will, according to OSS//FS proponents, increase security and autonomy of the users (ibid.) Bugs is fewer and attack of viruses is less critical. With open source it is possible for any one to look in to the code to make sure that a system does not violate security issues (i.e. espionage). At a fundamental level, nations must be able to rely on systems without elements controlled at a distance in order to guarantee national security (ibid.)

Intellectual property rights and productivity

"With increased emphasis on and pursuit of intellectual property rights enforcement at the international level, the choices available to software users are becoming more distinct." (ibid.)

This is the main motive for the Indian government to initiate the developing and use of open source (Nagaraj, 2002). Microsoft had 94% of the desktop market, and for India to avoid purchasing of licenses when they start implementing intellectual property rights they have decided Linux as 'platform of choice' (ibit), and created a strategy to promote Linux.

The transition to OSS/FS because of intellectual property might also result in increased in increased productivity:

"In one sense the provision of a freely available technological infrastructure represents by itself a form of wealth transfer to developing countries, but it is a wealth transfer that can have dynamic development effects. To provide real products and services on top of the infrastructure requires an investment of local labor to start. Many emerging economies have a surplus of inexpensive technical manpower. Combining this with free software tools creates the possibility of an interesting kind of comparative advantage that will matter in local markets and in some cases might become important on global markets as well." (Weber, undated)

For developing countries to take the full advantage of open source, and to meet the goals of more none-reliant projects, a certain IT infrastructural and skill conditions need to be met. Weerawarana and Weeratunga have (Weerawarana and Weeratunga, 2004) lists up six factors which are vital for creating economical benefits for developing countries by using OOS/FS.

- Intellectual property, law framework and enforcement
- Low cost, widely available Internet access
- Educational infrastructure
- Freedom of information
- English-skilled developers
- Skilled or trainable developer pool

Weerawarana and Weeratunga emphasize the importance of a good law framework and enforcement to protect intellectual property. If this law is not well enforced the long term economical benefits of OSS/FS will be under mined by people using pirate software which like OSS/FS is distributed free.

To participate in open source the development developers must have easy access to internet, since most communication is done thru internet forums or email (ibid.).

An educational infrastructure will promote the use of OSS/FS systems like Linux, and make people aware of alternatives to proprietary systems (ibid.)

Freedom of information is required for developers and users to access the information they require (ibid.)

Programming languages is based on English, and communication thru open source networks, especially when the pass national borders, is English. This requires English-skills (ibid.)

If a skilled or trainable developer pool does not exist, it will not be feasible to participate in open source development. Such a pool is therefore crucial for open source development (ibid.)

2.7 Open Source GIS-Technology

In 1994 the Open Geospatial Consortium (OGC) was established to simplify and bridge different standards used in GIS (opengeospatial.org, undated).

Much geographical resources are under development by the open source and free software community. This includes software, geo-data and documentation. The resources are also beginning to be quite well documented. Several web portals offer structured overviews with categorization and short comments. Most important of theses are the Free GIS Database (www.freegis.org), Remote Sensing (www.remotesensing.org), The Free GIS portal also provide an index of specialized GIS resource pages.

Among software products, there are several different categories. At the top level we can distinguish between (i) databases for geo-referenced data, (ii) visualization and exploration tools, (iii) spatial analysis tools, (iv) digitizing tools and (v) map servers. The tools can further be divided into web software and desktop software.

The two most important database servers (due to their existing user volume) for geo-references data is MySQL and PostGIS (add-on to the PostgreSQL server) At the time of writing, the latest stable version of the MySQL database is 4.1. With this version MySQL AB introduced spatial extensions. The spatial extensions allow storing and analyzing geo-data in a limited way. Advanced queries for spatial analysis of relationships between data is however planed implemented in future releases (Karlsson, undated). PostGIS is in more or less the same situation as MySQL. They continually implement GIS functionalities for they database (Wagner, 2005), but has still to implement advanced spatial query functions.

Other important GIS software is uDig (udig.refractions.net), MapServer, GeoTools, GeoServer, GRASS (www.grass.itc.it), JUMP and QGIS The first one, uDig is a spatial data viewer and editor. MapServer is a development environment for constructing spatially enabled web applications. GeoTools is a Java library with classes for working with maps and geo-referenced data. GeoServer is a map server which enables accessing and modifying data via web interfaces. GRASS is a geographical tool for data management, image processing, spatial modeling, and visualization. It is currently supported by a project called JGRASS which is porting GRASS to Java. JUMP is GUI-based application for viewing and processing spatial data through build-in functions or custom plug-ins. QGIS is an advanced visualization and exploration tool.

3. Background

3.1 Health Information System Program (HISP)

The Information System Program (HISP) was started after the fall of apartheid in South Africa in 1994. Initially HISP was based at two Cape Town Universities, and was receiving funding from the Norwegian Agency for Development Cooperation (NORAD) during a two to three year pilot project. HISP's role was to take part in the Reconstruction and Development Program launched by the ANC to reconstruct the health service in South Africa (Sæbø and Titlestad, undated). The reconstruction was done by developing a district based health information system including software, standardization of health data, and general approaches for reconstruction of health services.

The open source District Health Information System (DHIS) application was developed in 1997 and is still being further developed. When the pilot phase turned out to be a success, the strategies, processes, and software developed in the pilot areas was adopted by the Department of Health in 1999 as national standard.

The vision of HISP is;

"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" (HISP South Africa, undated)

The principles of HISP:

- Empowerment, democratization & transformation
- Develop an Information Culture at local level
- Action-led District Health Information System
- Integration of health & management information systems
- Computer software supporting processes, not driving them

- Primary focus on district teams, supervisors & facility managers
- Shift power from IT managers to Health managers/workers
- Appropriate training programs
- Educational programs (Certificate, Diploma, M.Phil, Ph.D) key elements in developing professional skills and career paths

The success of HISP in South Africa and the fact that the software is free, combined with the interest of Norwegian researches, has led to an export of DHIS and ideas to many developing countries, such as; Mozambique, Malawi Mongolia, Cuba, Nigeria, Tanzania, China, Ethiopia, Vietnam and India.

3.2 District Health Information System (DHIS)

The District Health Information Software (DHIS) was created to support the health information system in South Africa. The design team came up with five design parameters that would be used (Braa and Hedberg, 2002):

- The application must support the hierarchy of essential data sets, that is, allowing users to add, modify, or delete local data elements, indicators, and so forth.
- The application should be designed in such a way as to support the drive toward decentralized capture, analysis, and use of data – in particular, support the push toward having the facility staff responsible for data collection also doing data capture, quality checking, initial processing, and output.
- The application should be easy to use for new areas (provinces, districts), and should allow users to tailor the geographic scope of their data sets to their needs. This resulted in the use of a front-/back-end solution in Access, where the back-end data files cover different areas and the user can switch between them at will.
- The application should as much as possible rely on the flexible and powerful analytical and display tools already available within Office 97 (e.g., Pivot Tables in Excel), even if this increased the learning curve.

- The application should be free (open source) software – both gratis and with free distribution and redistribution of the source code.

The DHIS software was based on Microsoft technology with a database and a user interface created in Microsoft Access. Since the first launch of DHIS several versions have been developed.

Even though DHIS has proven it self to be a powerful tool for integrating the HISP ideology, there is now a project for remaking DHIS. The new version will be made in Java (DHIS2), and will therefore be platform independent. This is done in order to make DHIS2 independent of proprietary systems. The DHIS is free software, but its dependency on MS windows and MS Access made it necessary to purchase expensive licenses.

A switch to Java does also mean that the project can man make use of the many advanced tools provided by the open source Java community. Currently the HISP team is looking into libraries for reporting, pivot tables, wireless services, data storage and integration with office tools (HISP, 2005b).

3.3 GIS in DHIS

Currently DHIS has support for map functionality through the Report Generator Module. This module has a simple GIS interface that allows the creation of health thematic maps. These maps can be exported and saved for later display in the free GIS desktop viewer ArcExplorer (HISP, 2005).

There is also a GIS project being developed an Indian master student at the University of Oslo. His GIS application, HISP_SpA, is developed for integration with DHIS version 1.3 (Berg and Lewis, 2003). HISP_SpA allows the linking of monthly routine data with the district maps which. The software is developed in Visual Basic 6.0 and just like the maps created by Report Generator Module it supports ESRI maps (HISP India, 2004).

The current GIS functionality in DHIS is made in Visual Basic, but the version of DHIS2 will be made in Java, and there is a need for a Java based GIS module to support the platform independent DHIS2. The application developed by the two authors, HISP GIS is a prototype for such a module.

3.4 India

Our work in India was based on needs from the Indian health workers. We will therefore in this section give a quick review of how the Indian health system is organized and what information technology (IT) that is present. We also look at the work of HISP India.

3.4.1 Indian health system

The Indian health system is divided into five organizational levels:

1. Level: State
2. Level: District
3. Level: Division
4. Level: Mandal
5. Level: PHC (Primary Health Centre)

The main structure of the health system is much the same in all Indian states. We describe here a generalization of the information flow in the Indian health systems.

Detailed data is collected at the two lowest levels, PHC and Sub-centre. Sub-centre level is below level 5 (PHC) and placed under administration of the PHCs. The data is collected and reported to the next level in the hierarchy. This way each level has the aggregated data of all the levels below their own level. The DHIS database is based on this hierarchy. It stores aggregated data for five levels. In addition the PHC has their own DHIS database for the administration of Sub-centre data. This means there are two DHIS database files, one for Sub-centre data and one for data aggregated from health level 1 through health level 5.

The Department of Health is situated at the state level and is directed by the commissioner which is in political charge for the whole state. The commissioner is under the administration of the Government of India. Reports generated at district level are received by the department of health for further reporting to the Government of India. Decision making, concerning the overall health service of a state, is done at the state level based on districts reports and decisions and resolutions imposed by the Government of India.

At district level the District Medical & Health Officer (DMHO) collects data reports from PHC level. The DMHO is the head of the health service provided at the district level, and in the DHMO mandate is the power to make decisions inflicting PHCs and sub-centers. Reports gathered from sub levels is re-reported and sent to the state level.

The district level is further divided in to divisions. Division is a "support level" for the district. In many cases the division level is just an organizational level in the hierarchy without any employees. Employees working for the DHMO to fulfill the DHMOs tasks will typically work in the division level. The extra level between PHC and district is thought to make it easier for the DHMO to support and help PHC staff. The division level helps the DHMO to gather PHC data and to get a stronger dialogue between the PHC and district level.

At the PHC level the Medical Officer is the head chief. The Upper Division Clerk (UDC) collects the data from sub-centre levels for consolidation. The reports generated from the consolidation are used locally and some are sent to the DMHO office at district level. The Medical officer has to a certain extent in the mandate to set focus on health issues and deploy campaigns directed to improve the health situation in the PHC area.

The sub-centre is the root level where Multi Purpose Health Assistants (MPHA) is providing most of the medical/health services to the public. The MPHA makes records of all the services provided to the public (such as ante natal care, vaccination etc) and report this data one to two times a month to the PHCs. In some sub-centers a head nurse is coordinating their operations, but usually the MPHAs work mostly on a solitary basis.

3.4.2 IT in Indian health system

Information in the Indian health system is basically exchange thru paper based forms and reports. From sub-centre level to state level the job of calculation and filling of reports is done with pen and paper, and transportation of the reports from a level to another is done often by the staff them self. Programs (i.e. family planning and National Anti Malaria Program) initiated on national or state level often results in increased reporting and more administrative work for health personnel on the expenses of direct health service for the public. To reduce the administrative workload the state of Andhra Pradesh hired a software company to develop a system to support health related programs. The system was to be called FHIMS, "Family Welfare, Health Information and Monitoring System" (CMC, undated)

The FHIMS would be implemented on PHC, district and state level. Each PHC would have one computer running FHIMS modules supporting each program. On a regularly basis the PHC systems would send data to the district thru the internet and the district further to the state level, also thru the internet (fig 1).

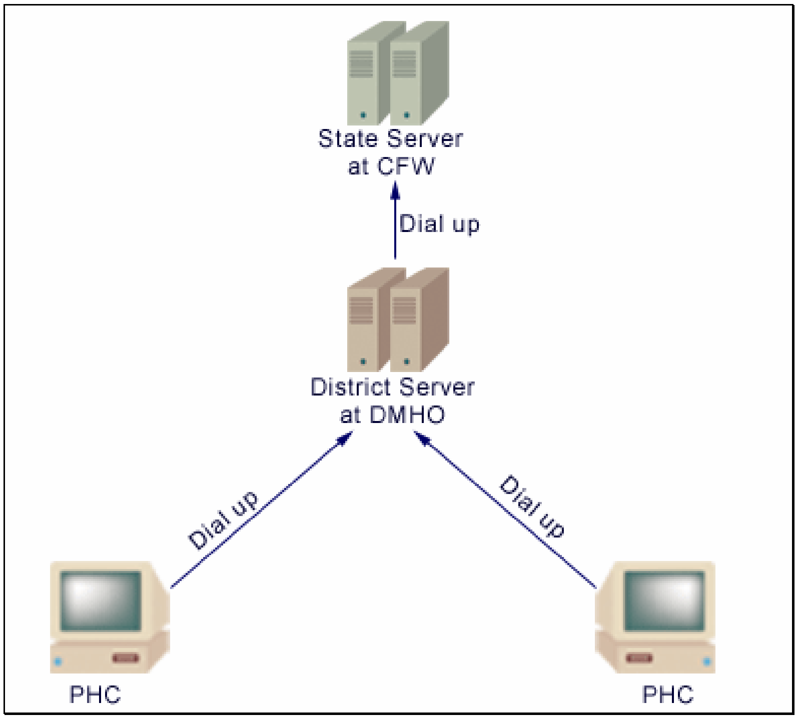


Fig1.

The system is made with Visual Basic 6.0 and uses Oracle 8i/9i as its database. The operating system used for the system is Microsoft Windows Professional/Server.

3.4.3 HISP India

HISP India was started in December 2000 in the state of Andhra Pradesh in cooperation with the University of Oslo (HISP India, undated).The HISP India team consists of people with knowledge and backgrounds in informatics, medicine, public health, anthropology, and development studies.

The piloting part of the project was conducted in the primary health centers in Kuppam division (approximately 300 000 citizens) in Chitoor district. 70-75 staffs were trained on using computers, DHIS software, analyze data, and generate reports which were automated during the process of implementation based on feedbacks from the health staff and officials. After the initial pilot face, the project expanded to include one more division, and to cover 50% of the PHCs in Chitoor district. In the beginning of 2004 HISP India and the State

Health department signed a memorandum of understanding (MOU) to support the health information activities in all the districts (23) of the state.

The MOU project would cover the following components: (ibit.)

- Design, development and implementation of Infant Mortality and Maternal Mortality Monitoring systems in all district offices.

- Creation of district database for health information systems for all districts in the state.

- Stabilization of the Family Health Information Management Systems in Nalgonda district.

- Integrating Geographic Information Systems with the routine health data being collected every month.

- Web enabling of existing health information systems.

- Large scale training and education expenses of health staff at different levels.

- Integrating the health information systems with the broader state e governance efforts of the state.

Along with the development of HISP in Andhra Pradesh the HISP India team has started up a new project in the state of Kerela.

4. Fieldwork & Observation

This chapter is divided in to two sub-chapters. During our development period in India we encountered several issues concerning the co-operation with HISP India and practical difficulties such as lack of infrastructure. In the first chapter we tell about how we experienced working with HISP India as a team, and how this inflicted the development process. In the second chapter we tell about how we had to adjust to practical challenges.

4.1 Working with the HISP

In the beginning of the project period in India the team counted eight people. The group consistent of four people from HISP India, one volunteer, one Indian master student, and the two authors. The first face of the project was conducted at a small camp a two hour drive outside Hyderabad. According to the different skills of the team members, we were divided into two project groups. One group was to create a web based form for entering health data. The other group was given the task to create a GIS application for the DHIS. Both project were important to the HISP India. Team members would basically work with one task, but could also co-operate on cross of project groups. All agreed on the distribution of work.

The next days we, the authors, worked on separate groups. One focusing on GIS, and the other a web framework. After the initial learning phase, we began to make requirements and plan the development phase. At this time, however, we started to realize that the lack of progress made it impossible to continue as planned. We therefore had to reconsider our projects. Our primary goal was to make a stable and functional GIS application. It was important that this was accomplished, both for our theses and for HISP India. As the progress indicated that we would not have time to develop two applications, we chose to focus all resources on the GIS project. This decision was taken in collaboration with the Indian HISP.

At the time when we decided to drop one project, only the two authors and three other remained from the original team. This meant that three persons had left the project in the initial one to three weeks. One person was working with too many other projects, one had plans of quitting the HISP team and was looking for new jobs and one left for reasons unknown. This was the first of several unexpected turns we experienced during our stay in India.

We experienced that plans and agreements often implied a great deal of uncertainty. This was something we where not prepared for. We did know something about this in advance, but were almost always taken by surprise when changes occurred. In some cases we experienced that there were given promises that were bound to be broken. We found the reason for this to be the politeness of the Indian workers. Almost without exception, the answer to a question was positive. It only rarely happened that we were told something was not possible. In stead everything was told to be possible, and that it would be taken care of, without anything happening. This was of course their way of being polite, but to us, who were not used to this behavior, it was a source for misunderstandings.

The politeness of the Indians also made it difficult for us to understand the contents of agreements we made, the importance of them and how strong the motivation was for keeping the agreements. We felt that the politeness compromised the honesty in discussions. This became better as we spent more time working together, but it remained a barrier throughout the development process. For instance, if we asked someone if they could spend the next days on requirements modeling, they automatically answered yes, when in fact they did not have time for it. The same applied for the Internet connection. A connection was often promised for one or two days, when in fact the chances for getting a connection were very slim. Again, it does not mean that the people not were to trust – they only tended to promise more than needed. When we discussed this with our team members later, they said it was part of the Indian culture. A promise of doing something tomorrow, meant that it might be done tomorrow, maybe the day after tomorrow or possible not at all.

The consequence of this was that we started to make our own agenda for the application. We had very limited time for developing the GIS application, and in order to finish what we had started we had to make progress with the development every day. This resulted in us leading the development process into a track that fitted our own programming skills. The HISP team members acted as resources for information about the DHIS database and the Indian Health structure. This was not beneficial for the one member of HISP India who where keen on participating with programming work. As his Java programming skills was too limited in order for him to contribute, and the fact that we did not have time to train him, he had to do database work with the maps. He did very important work, but this way of separating tasks considerably lessened the build up of knowledge about the GIS software in the HISP team. It was however a necessary approach in order to finish the software.

We would have needed more time for developing process in order to include everybody on the work. It was however difficult for us to do this as we had a strict deadline for our master thesis. We could not spend more than two months doing fieldwork, therefore we had to use all available time on the developing task we had been given. In contrast to us, the Indian HSIP team had work which they had to do parallel with the developing process. This was for the most administrative work that had to be attended in the city, 100km from our work site. In addition to this, the two months we spend in India were very busy with religious and cultural festivals. Most of these festival were conceded holidays, and were spent with family living far away from were we lived. The time the HIPS members were away we could not afford to loose, therefore many decisions had to be made by the two authors without consulting the rest of the project group. This worked ok, but reduced the HISP members' knowledge about the software.

4.2 Practical challenges

In the large cities the internet access was better than expected. Thru internet cafes people could benefit from general services provided thru internet, and we could access the forums our project required. But a great challenge for our project would be the fact that we most of the time lived outside the major cities without internet connection.

Our main camp, Thrive, was cited outside a little village 100 kilometers from Hyderabad. When we first arrived at the camp we were promised access to internet thru a mobile phone connection. The internet access would then be distributed thru a local network. With this promise we settled at the camp and initiated our project. After some organizational problems in the beginning, explained in previous chapter, we started to explore the software packages we would use for our application. A lot of the packages, libraries, and documentation we had gotten the hold of before we arrived in India because we had been informed about the general poor computer infrastructure. But when we started to work on the implementation of the application the libraries we were about to use were dependent on other libraries which we did not have.

When the camp still waited to get internet access we either had to go to Hydrabad our self, or call people in Hydrabad, to download what we needed from the internet and bring it to the camp. The process of getting hold of libraries and documentation became time consuming events. We considered the option of moving from the camp to another place with internet accessibility, but the continuous promising of internet access made us stay at the camp.

During the period at the camp we tried to spend most of the time working on the application, but on a daily basis the electric power lines where down for some time. Using laptops made it possible to work without high voltage lines for a couple of hours, but when the batteries ran out we had to put the work aside. This was a common incident most places in India, but the amount of time the electric net was down varied from area to area, and the best "up time" was in the bigger cities.

The time spent on getting libraries and packages, and the time spent on waiting for electricity, delayed the project considerably. After to thirds of our project period we realized that the camp would not get access to the Internet. At this point we decided to leave our camp, and stay at places with continues supply of electricity and easier access to Internet.

4.3 ICT and human resources in the health system

To get a better understanding of how the work was organized within the Indian health system, and how they related to computers, we visited several facilities at sub-centre, PHC and district level in different states. Computers where not common in the health system, except in the facilities that were provided with systems to run DHIS or FHIMS. In fact the facilities with both the systems would have two separate computers (pic. 1). It seemed strange that some facilities would have two computers while others did not have any at all. We were told that the owners of FHIMS denied other software to be run at their computers, and DHIS had to be run on computers provided by HISP India.

Pic. 1:

When talking to a supervisor (doctor) at a PHC in the state of Kerela, he spoke of problems with storing gathered data (pic. 2), and resources wasted on filling in separate reports containing the same data. He meant for sure that a computer based system could help. People working for HISP told that everybody wants a computer, but they tend not to use it when they get it. In one case where both FHIMS and HISP computers were installed, only the DHIS system was used because of two reasons; lack of resources to operate two systems, and the continued dialogue and support and sometimes persuasion of health workers to keep using the system. Health workers did not see any effect of using a computer based system. For them, in most cases, the computer system would just be an extra task among everything they already did.

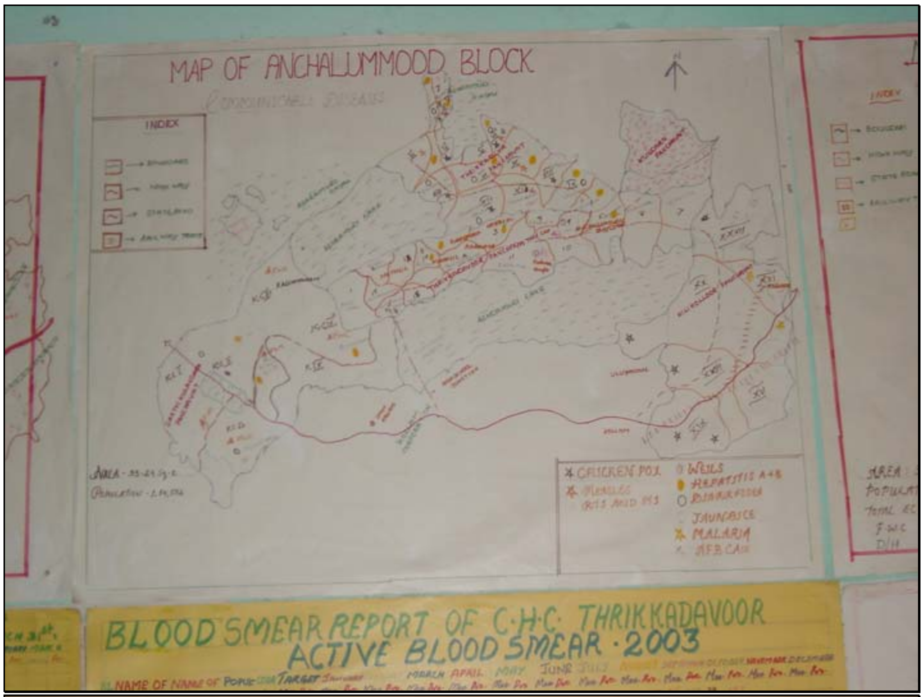
Pic. 2:

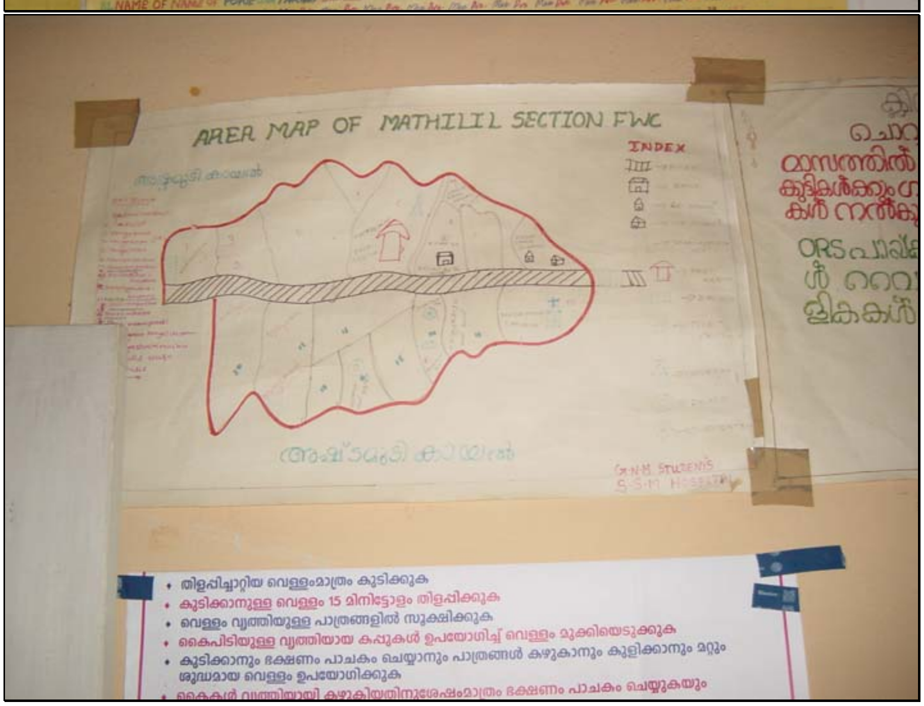
The general computer knowledge seems to be very poor. At district levels typewriters is much more common than computers. We were told by officials and health workers that the lack of resources to buy computers and provide training is the reasons for the absent of knowledge. Even with the presence of a computer the knowledge seemed very poor. When health workers have access to a computer, just a few actually uses it, and it's used for one purpose only. One example is that a computer running DHIS is only used for that. Another example we experienced, on a district office, was a computer used only for running MS Excel, where MS Excel was only used for writing a handwritten style sheet manually in to the program so they then could print it out. The handwritten style sheet is a report scheme where each row is calculated manually. In our latter example the computer operator used a handheld calculator and wrote the values in to MS Excel in stead of using the functionality of the software. To us, it seemed like a big challenges to convince the health staff what a powerful tool a computer can be, and how to provide them with the proper kind of training.

When it comes to geographical health planning, we where amazed by the health personal at the facilities we visited in the state of Kerela on sub-centre and PHC level. At these facilities they planned much of there work using hand drawn maps (pic. 3 & pic. 4), which represented everything from health workers area of responsibility to maps graphically presenting the spreading and density of diseases. According to the staff the drawing of the maps was initiated by them self to make it easier to create action plans to prevent diseases (i.e. malaria) and to plan routine work done in the field. They gathered data in the field and used this in combination with population data to create graphical representations of the health situation to create new action plans. The map use at the other health facilities we visited was however not as extensive. Some used maps only for viewing the villages in their region, at most facilities however, we saw no maps at all.

Like all the other health systems they also used this data to fill out other forms that were sent up one level in the hierarchy. People in HISP told us that even when the health systems architecture is similar in most states the regime may be very different. This can be related to the people with political power in the different states. In some states the authorities main focus was to deliver "good numbers" (i.e. low infant mortality) to gain popularity among the public and their "chiefs" higher up in the hierarchy. This would often make staff at the lowest levels to manipulate the data to fit the goals for the above authorities. Delivering bad numbers could lead to their resignation. In this "good numbers" regimes the goals where sat at upper levels and in most cases this goals where impossible for the lower levels to live up to. Afraid of loosing their jobs the staff at the lowest levels only provided the health care needed from a day to day bases, and the data they gather is more or less useless since they modify it before it is reported. This was not the case we observed in Kerela. We questioned people about the authorities' motivation for gathering health data. The authorities of Kerela made overall goals for the state and the districts. The goals were based on statistical data, and if they were not reached, special tension would be directed to those health issues. Personnel at the lowest levels would set up individual goals to reach the overall goals. No one would get punished if their goal was not reached. The focus on health issues in stead of "good numbers", seemed to us as a motivation for health staffs to perform well, and also to take interest in data they were gathering. They saw a connection between the continuous reporting and the good infliction it had on their job. The use of map, they said, make it easier to get a feeling of the health situation and make plans.

One more observation we made, was the story from a worker who told us that the different departments had limited interest in fixing other peoples problems. He said that if they delivered health reports to the wrong departments at public offices, they would simply be thrown in the litter.





Pic. 3 & Pic. 4:

4.4 Open source

During our project we where invited to participate in conduction of interviewing computer engineer. The development of DHIS2 required more people with skills in Java and people with open source knowledge would be preferred. Our role at the interviews was to question the candidates about technical knowledge. All together we interviewed ten people. Out of ten only one had heard about open source, and he concluded that it was not possible to make money with open source. However all of the candidates had some skills in Java, and about half had worked with web application. One of the candidates had also worked with Apache Tomcat. Since most of the candidates had a master degree in computer science, and all had knowledge in the multi platform programming language Java, we very a bit surprised that just two had heard about Linux. One of them had just heard about it, while the other one had seen a computer running it.

The two candidates that had heard about observe Linux was the only awareness of the system we observed. When we asked the HISP India team about this, by emphasizing that Linux where free and MS Windows expensive, they answered that Indians in general do not pay for Windows either and that most systems were pirate systems. Both private and public computers would run pirate systems. We were also told that it was not uncommon that systems required for DHIS was pirate software because limited budgets could not include purchasing of licenses.

During conversations with other Indian master students it was said that getting a job in a software company for Indians' could be the big 'break thru' in an Indian life. The companies could bring prosperity people often just would dream off, because of the salaries in this companies would be from two or three times to more than forty times the average income in India. He said that this is the main motivation for Indians to study computer science. Further on they believed that the systems they were trained was going to be the same system they would use working in a software company.

4.5 Summary

During our period in India we encountered challenges that we had to solve. Communication problems and poor access to basal computer infrastructure was however challenges we overcame by adjusting to the Indian context. We learned to better understand how Indians communicate and the "Indian way" of doing things. We also realized that access to internet was more a luxury than something common. In our project and other projects alike access to internet is crucial, making it hardly impossible to live in rural areas and develop software based on open source.

We also encountered challenges that were outside our reach. Issues such as providing the health system with computers and proper training, and make the users see the values of those systems and how they may make there day to day work more convenient. If staffs do not see the reason for gathering data, except for providing their authorities with "good numbers" to keep their jobs, why would they be interested to learn and use a computer system?

5. HISP GIS Open Source – The test application

In this part of the thesis we present our GIS test application. We will refer to the test application as HISP GIS (Health Information System Programmes Geographical Information System).

5.1 Introduction

HISP GIS is a combined web and desktop open source visualization tool, developed for testing possible integration of GIS with the DHIS. The project started January 2005. For two months we were heavily involved in creating the geographical information system. The project was collaboration between HISP India and the University of Oslo, but all programming was done by the authors.

The starting point for the HISP GIS application was that HISP wanted a GIS application for working with the HISP database for health indicators. Primarily the GIS application should offer a new way of working with data and health organization units. Through map navigation the user should be able to more easily compare neighboring units, and get statistical information about them. Reports could be geographical in the form of colored maps or in the form of traditional line and bar diagrams. Ideally, the GIS application should be easy to integrate with both existing HISP software and upcoming versions of HISP.

It was important that the GIS application satisfy the needs of health administrators. This means providing health administrators at the district and state levels with a graphical tool for assessing the health care situation and planning day to day working tasks.

5.2 Requirements

We did not have the opportunity to query health administrators or primary health care personnel for their geographical needs. We were however in constant contact with the experienced Indian HISP personnel. They had been working closely with public health employees for a long time, and knew some of their needs. We therefore based the requirement of HISP GIS on their statements.

The geographical needs for health administrators are:

1. Practical map use:

- Guide new employees – generating maps with road, street and settlement information.
- Make routes for where health workers shall do health work
- Get an overview over the health region and underlying/neighboring administrations

2. Managing health data

- Analyzing geographical referenced health data
- Plan prevention work
- Look for causes for disease outbreaks: still water (where malaria spreading mosquitoes can lay their eggs), pollution sources, sewage network, etc.

3. Managing human and non human resources

- making maps with location of health centers and their vehicles, important health equipment and patient capacity
- locating personnel with special competences

We had to consider these needs, the requirement of making a stable and usable application, and the very limited time we had for development. All of the requirements could not be implemented. We therefore limited the requirements to include functionality covered by these two points:

- Navigate geographical regions by zooming and panning maps
- Selecting multiple regions and get statistical health information about them

This means that we left out one of the major functionality:

- Viewing semi permanent data for health units

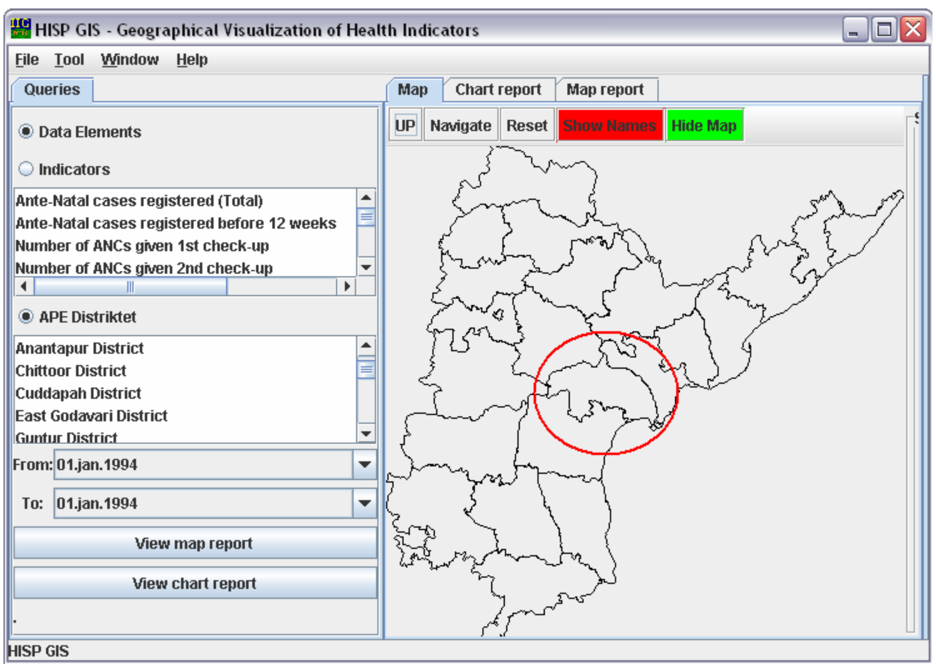
Semi permanent data means human and non human resources. Leaving this out limits the use of HISP GIS, but we saw it as more important to get a stable and well functioning application, than implementing all needed functions.

5.3 Functionality and appearance

Separation of model, view and controller layers was the starting point for the application. Such a separation we considered as a vital key for later changes in HISP-GIS's working environment. This is especially important for the HISP project as it is in the process of switching technology for its DHIS application.

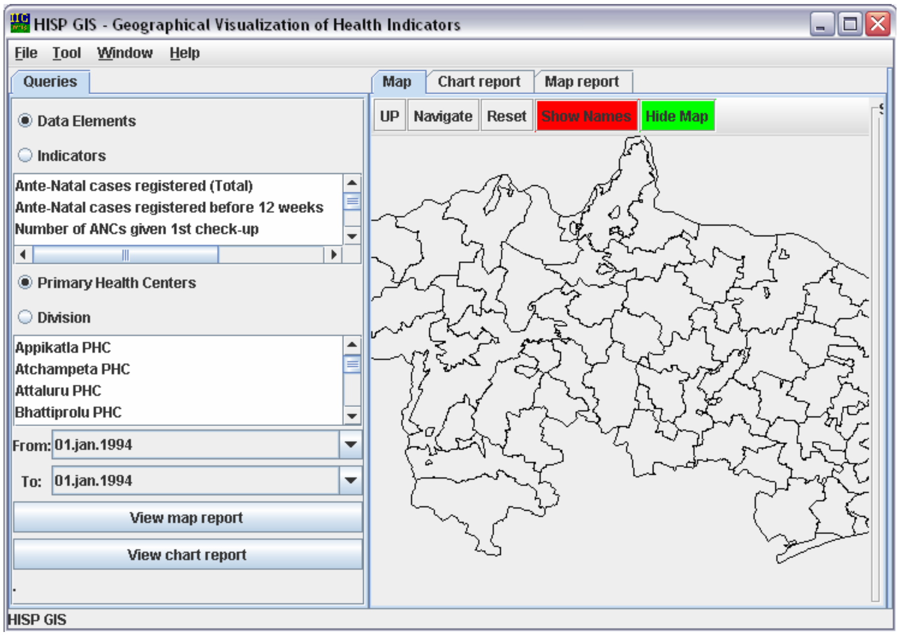
The application GUI must be simple to use. It should not have the common GIS interface with terms such as layers etc.

As a mantra thru the development period, the simplicity of the application was repeatedly said to be of highest priority. It should be simple to use, and yet offer advanced functionality. The graphical user interface is therefore made with as few buttons as possible and a plane graphical area to display maps and graphs (pic. 5)



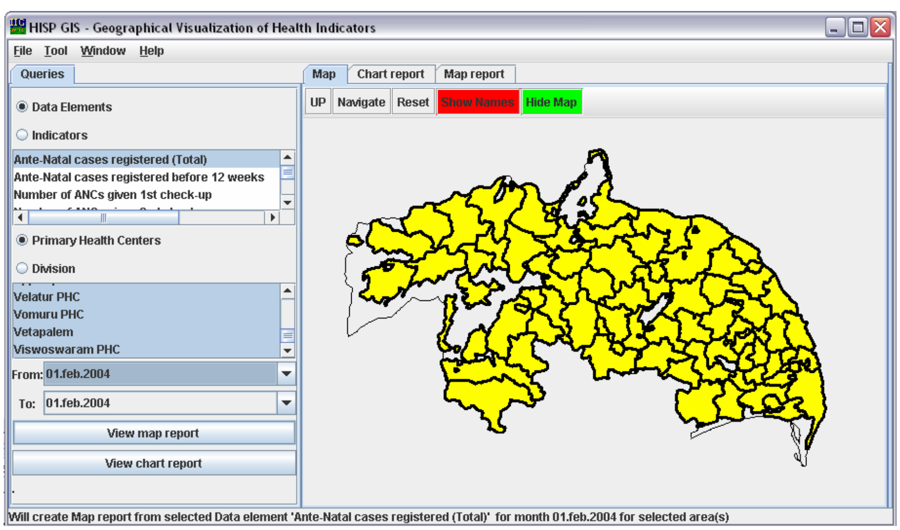
Pic. 5

The workbench of HISP GIS is divided in to two spaces. On the left side parameters are listed from which the users can choose to generate reports. These parameters will adjust according to which level the users are looking at. To choose which level to look at the users use the right space and double click on the map to "zoom" in to the requested level.



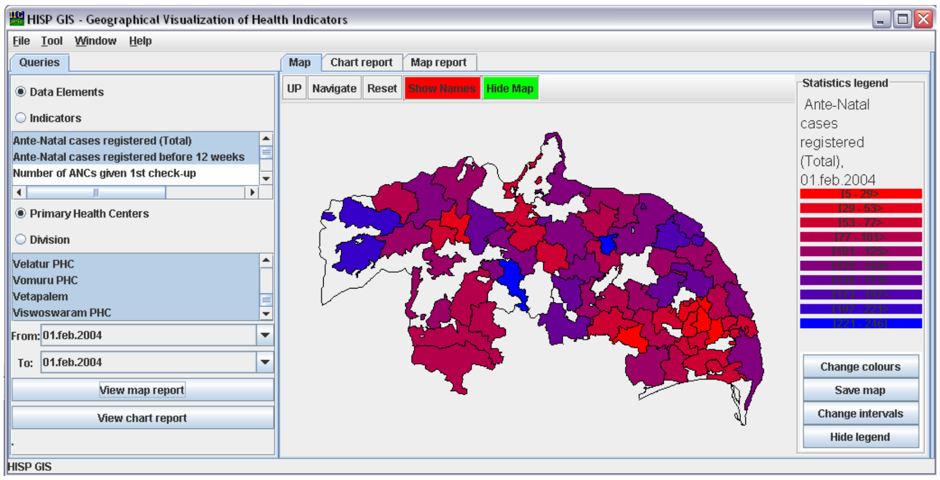
Pic. 6

For example: If the user double click a district (in this case the Guntur District marked with a red circle in Pic. 5) the HISP GIS loads the map of Guntur District (Pic. 6), and change the parameter data on left side of the workbench to correspond with report options for this level. To support different kind of reports the HISP GIS "figures" out how to generate the reports by looking at the parameters the users have chosen, and it also display what kind of report the user is about to generate when the user holds the mouse over the action buttons ("View map report" and "View chart report")(pic. 7).

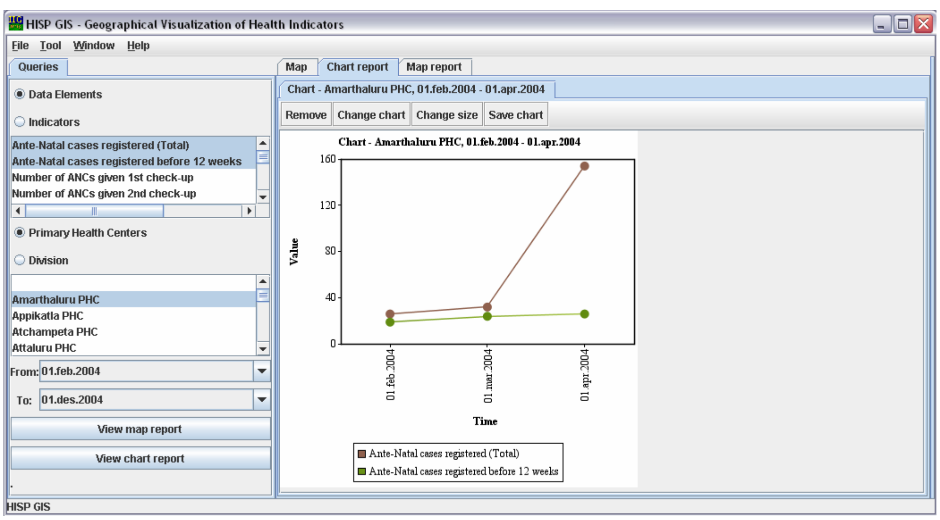


Pic. 7

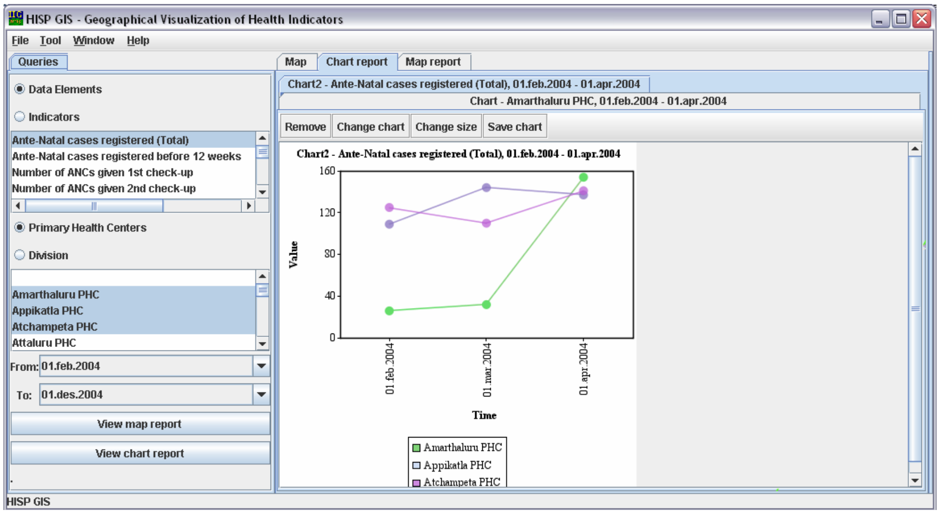
The information displayed for the user at the bottom of the workbench makes it easy for user to see what they are about to do, and makes the HISP GIS intuitive and easier to understand when the user is not familiar with the program. The next three screenshots (Pic. 8, 9, 10) show how the application deals with three different requests from the user:



Pic. 8: The user has chosen to view a map report for all primary health centers in the Guntur District.



Pic. 9: The user has chosen to see two data elements in a primary health center over a certain time. The user has chosen the primary health center by either clicking on it once on the map, or by selecting it from the scroll box.



Pic 10: The user has here chosen to see several primary health centers compared to each other by one data element over a certain time. The user only use the two action buttons to generate the report, but the HISP GIS have to translate the requests and then generate a report that correspond best to the users chose of parameters. This is the application in its simplicity for the end users. The end users can generate a numberless set of reports without having other concerns than how he will utilize the generated reports. When a report is generated the user can change the look of the report. For a map report he can change the interval of the legend and colors used in the legend, and for a chart report he can choose what kind of chart he want to use (area, line, bar, and etc). All the reports generated can be exported as pictures, for later to be used in any kind of presentation.

5.4 Architecture

The HISP project saw benefits with both a web based and a desktop GIS application. A web based application makes centralization of data easier. This makes it easier to create reports containing data from several region, and check that the data is new and consistent. A central database also means that data does not have to be distributed. Making the GIS module web based is also making it more in line with the rest of the HISP which aims at making the DHIS project increasingly web based. Both to ensure platform independency and eliminate the need for installation work and maintenance of software and hardware.

The two most important benefits when creating a desktop application is greater speed and better user interface. The possibility of having the database stored locally, will give much more rapid responses. And the desktop user interface has far more ways of interacting with the user; this decreases the number of needed dialogs and increases usability.

It is also important to consider the availability of an Internet connection. In most developing countries an Internet connection is only available in major cities – not in remote areas where many primary health care centers often are located. In such cases a geographical web service would not be a good option.

The HISP GIS solution has been to use Java Web Start (JWS). JWS applications can be started by anyone with a connection to the Internet. When pointed to the applications configuration file, JWS download all the needed files and launches the application. JWS is part of the Java Network Launching Protocol (JNLP) mime type (application /x-java-jnlp-file) and the application's configuration file is a JNLP-file. When a link to JNLP is clicked in the web browser, JWS lunches and downloads the requested application, the application is then cached locally so that you don't need to download it the next time. A JWS application can also be loaded independently of a web browser by invoking JWS on the client machine and pointing it to the JNLP file. Currently JWS is available for Windows 98/NT/2000/ME/XP, Linux, and the Solaris Operating Environment. This ensures platform independency. All Java Runtime environments above 1.2.2 support JWS applications.

When working with a JWS application, you are by default working in a protected environment (sandbox) with restricted access to local disk and network resources. Privileges such as reading and writing for local computer, and accessing remote servers must be granted by the user. This however, requires that the application is digitally signed. HISP GIS needs too be digitally signed only when the database is not located on the same server as the HISP GIS application was downloaded from.

A JWS application can also be written so that it does not have to be changed in order to be run as a desktop application. This makes it perfect for locations without an Internet connection. The application will work in the exact same manner weather it is run standalone or with JWS.

5.5 Design

In addition to the choice of making the application support JWS, in order to satisfy the need for a GIS web service, other decisions also had to be made for the application. The core problems were:

1. The MS Access database (data mart file) must be accessible to the application. We have the choice of migrating it to an open source database system, or hosting the file on a Microsoft Windows server

2. How can the program handle changes in the relation model and tables of the database

3. How can the map features be linked to health organization units in the HISP database

4. How can we make the application easy to extend

5.5.1 Database (Questions one and two)

The HISP database contains all data entered by health personnel. From this database, the HISP software can generate a data mart file containing all the elements and indicators aggregated for each organization level. It is this data mart file which HISP-GIS are working with. Currently the format of the HISP database is MS Access. We had the choice to migrate the database to an open source system or working with the MS Access version. We did not see the need for migrating to a different database system. MS Access use SQL queries for retrieving data or manipulating the database. To change database for the HISP GIS is done by changing the settings file for the program. Several database drivers are bundled with the application, and other drivers can simply be added to the java source-path. This way the choice of database type and database location is left for the user. At present however, the database source is only configurable in the settings file. When ran under Java Web Start, this means only the administrator of the web server can change database location. In standalone mode all users can change the settings file.

A different problem is a change in the relation model. This will most likely require new SQL queries because table names, table content or relations have changed. SQL statements are often hard coded in a programs source code. To make HISP GIS flexible to changes in the relation model, we store all SQL statements in the settings file. This way no source code has to be recompiled.

5.5.2 Connecting the HISP database to map features

Configuring maps

An important job when implementing a map system, such as HISP GIS, is obtaining and configuring maps. We already had maps[1], but we needed to prepare them for HISP GIS.

The maps we used when developing HISP GIS in Andhra Pradesh was on the ESRI SHP (shape file) format. ESRI (Environmental Systems Research Institute) is a large American company which develops a variety of GIS software (see www.esri.com). A shape file stores map features and attribute data as a collection of files with the same prefix. There can be several different files included in this collection, but three are required (ESRI, 1998):

- Main file (.shp) – storing features geometry.
- Index file (.shx) - stores the index of the feature geometry.
- Database file (.dbf) - stores the attribute information of features.

The main file has all information about points, lines, lines and polygons which the map consists of. The index file has information about how the points, lines and polygons are stored in the main file. For instance what data type is used and where a specific feature is located in the file. The database file stores one record per feature (one-to-one relationship) in the main file. A record can have multiple attributes. The format of the database file is dBASE.

In our case, the preparation of maps only included the database files (.dbf). The database files contained names for districts and mandals. These names did not correspond to those in the HISP database because they differed in spelling. We therefore first decided to use the organization unit number present in the dfb file and the HISP database for linking the database to the maps. This was however not possible since these numbers change each time the data mart file is generated. Because of this we decided to make the connection based on the organization names. An alternative way to do it would be to make a linking table that had to

[1] *The maps we used during development in India, had originally been prepared for John Lewis, an Indian master student at the University of Oslo, Norway. He experienced some difficulties which we find valuable to mention here. Maps in India are classified documents for the reason of national security. Publishing of any maps had to be clear before they could be used. The reason for this is probably India's turbulent relationship with its bordering countries. Fortunately for us, the HISP India had done all the work with obtaining maps for us*

be updated with the correct organization unit numbers every time a new data mart file was being generated. Performance testing of the MS Access database showed that the lookup time was equal for strings and integers. There where in other words nothing to gain in performance by using a linking table. Avoiding the linking table also made the configuration of HISP GIS tidier.

Linking health organization levels

In order for HISP GIS to support the Indian health structure and with that the HISP database file, we had to obtaining maps for all the organizational levels. This meant either having different maps for each level, or using the same maps for several levels. For the Andhra Pradesh case, this meant getting a set of maps that included these health regions:

1. State level
2. District level
3. Division level
4. Mandal level
5. Public health center level

In addition, there are also sub-centers. This level is below level five and is not a part of the HISP Andhra Pradesh setup. Sub-centers do however play an important role as it is here all data for the HISP database is gathered.

The Andhra Pradesh levels are of different types: state level means the entire state of Andhra Pradesh. District level is administrative regions consisting of several divisions. Divisions are logical regions (without administrations) consisting of mandals. Mandals are administrative regions consisting of villages and towns. Public health centers are geographical health regions consisting of villages.

The challenge with the organization of the Andhra Pradesh levels is that mandals and public health centers are overlapping regions: they both consist of villages. And further, villages within one mandal can belong to several different public health centers. This implies that level five in the structure not is a drill down of level four. This does not constitute a problem for normal use of HISP, which base its work on aggregating data up the hierarchy. But it adds an extra dimension for spatial exploration of HISP.

Normal map tools bases its navigation on panning and zooming. HISP GIS has both these functions. Panning and zooming is implemented as normal, with extra functionality for moving up and down the organization hierarchy. The problem that occurs however, is that when moving down the hierarchy from level tree, you must have the option of either going to level four or level five! This is because, as explained, level four and five are *geographically parallel* levels. You can not do a drill down from level four to level five.

One map used for several organization levels

Another issue occurs when there is no map available of a organization level. In our case this happened for level tree, divisions. The reason for this was that division only is a logic term for a collection of mandals. There are no maps with division borders. The lack of maps for selected levels is solved by defining the relevant level as a *cluster level*. A cluster level uses the map configured for an organization level below the cluster level.

5.5.3 Extending the application

It is important for the HISP GIS application to be compatible with present and future HISP software. It should be easy to integrate with a wide variety of services. One popular HISP service is pivot tables. When starting to develop HISP GIS, we knew that there was a great desire for integrating pivot table results with a map application. We therefore aimed at making HISP GIS easy to extend and implement in other environments by developing the map functionalities as a separate library, and then let the HISP GIS program be an implementation of this library. We focused on separating map functionalities, graphical user interface and data layer and building the application based on a traditional model view controller (MVC) design (Deitel and Deitel 2001).

5.6 Solutions to design issues

When designing the model for the HISP GIS, we had to consider three factors that where not explicit requirements, but implicit in the form of the environment the application would operate in. The first two where the issues of variation of health organizations and map resources. The second issue was the change of database type or location, and changes in the relation model of the database. These issues were solved by the use of an XML based settings file. We talked about these issues in the previous section and we will here show how these problems were solved in HISP GIS. First we will explain the problem of health levels and the availability of maps further.

5.6.1 Health levels and maps

The health systems levels of hierarchy are different in all countries, and can also be different within in the same country. Logical health levels are in the DHIS system referred to as orgunits where each orgunit type symbolizes a logical level in a health organizations hierarchy. The DHIS is constructed for dealing with the variation of health systems, meaning some organizations that might have three levels are represented as three orgunit types, while others might have four or more levels. This lead to a database looking different according to the health systems they represent (fig.2). One of the targets for the HISP GIS is to be integrated into the DHIS, leading on to the fact that when the DHIS database change the HISP GIS must adjust. By fulfilling the requirement of HISP GIS adjusting according to the database the application could operate in all health systems using DHIS.

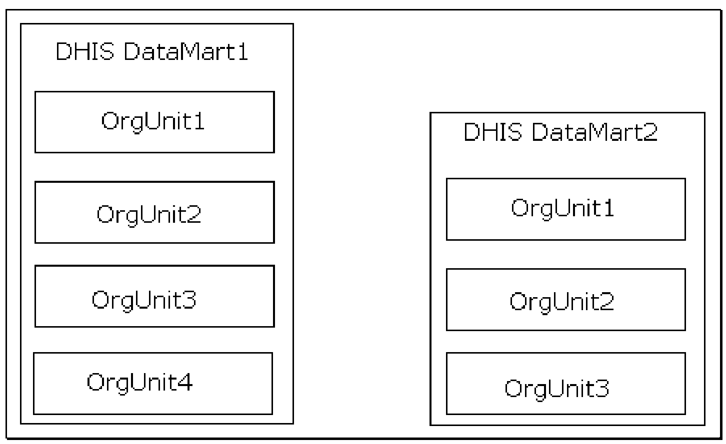


Fig. 2: A simplified illustration of a DHIS data mart files. The number of tables can be various in numbers, depending on the health organization they represent.

In the HISP GIS this is made possible by taking the SQL statements out of the source code and in to the XML file. When separating the statements we developed a simple quasi SQL scripting language the HISP GIS will translate into real SQL before running queries against the database.

During start up of HISP GIS the application reads the quasi SQL statements and put each in to a wrapper object were each object will represent one specific task predefined in HISP GIS. For example; one task could be to get all indicators in the database. HISP GIS would then use the object for querying for indicators. How the query looks is not a concern for the application, it expect to get a certain result when using the object for a specific task.

To create a geo-referenced user interface the HISP GIS uses electronic maps of the SHAPE format. These are used to display the logical health boundaries on maps. Since the current data base used by DHIS is MS Access the maps could not be integrated directly in to the data base. The map files and the database are therefore linked together thru a separate logic built in to HISP GIS (fig.3). Given electronic maps representing a distinct health level, each map could be linked to a distinct health level record in the database. During the development it came clear that the logical health levels represented in the data base were not supported on all levels in the available maps. It also came clear that maps viewing the geographical and political borders are various in details, geographical referenced points, and availability.

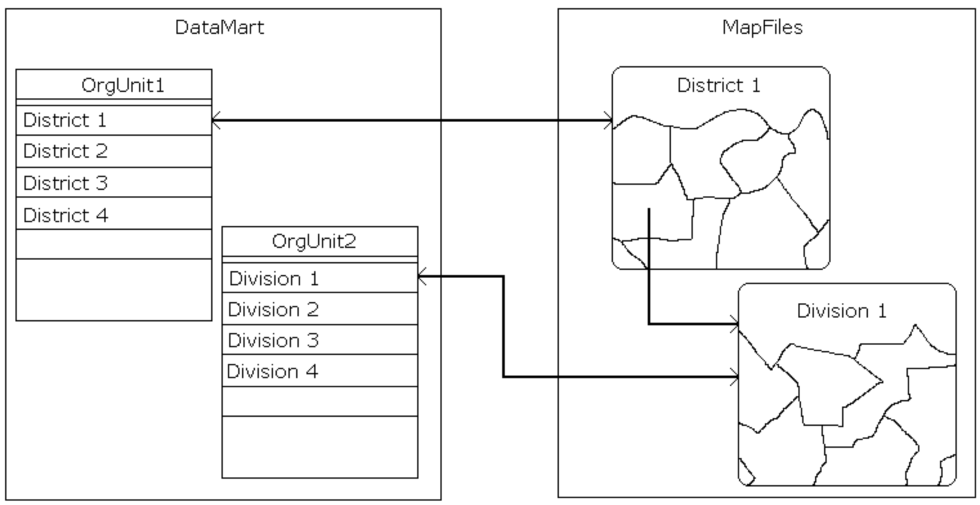


Fig 3: By using logic built in to the XML file the HISP GIS can link the DHIS data mart file with the map files. It also link map files with each other, making it possible to navigate from one map to another.

To solve the issues concerning the poor map quality and the lack of maps to represent a logical health level, we created a separate logic to let clusters of geographical areas represent distinct units within a health level. This way of approaching the challenge let the HISP GIS adjust to the quality and availability of the maps.

5.6.2 The HISP GIS configuration program and settings file

To overcome difficulties as the transition between MS Access and MySQL, poor map qualities, availability of maps, and the combination of remote and local functionality, and at the same time make HISP GIS simple to use for the end users we separated the administrative part of the application away from the end users. This was a result of the consideration of computer knowledge of the end user, that in most cases were restricted. In consultation with employees of HISP India it was decided to let people with computer experience to set up the HISP GIS before it is distributed. In our case the technical employees of HISP India would do that work.

The HISP GIS configuration application let the administrators to set up a list of parameters which is used by the HISP GIS during execution. The parameters are:

- Database
  - Name of the database
  - Localization of the database
  - Username and Password of the database
- Maps
  - Name of level
  - OrgUnit name (Used for linking with database)
  - Cluster name and OrgUnit (if necessary)
  - Name on maps belonging to this level
- Linking
  - Name on the map
  - Name on geometric area
  - Name on the map the geometric area should point to
- SQL (quasi SQL)
  - SQL name (A unique ID , used for wrapping the SQL to an object)
  - SQL Description
  - The quasi SQL string

All of these parameters are written in to a XML file called settings.xml. A typical settings file for HISP GIS will be more than 500 lines, and it would be very time consuming to write it manually. In addition all the map files to be used with HISP GIS must be put in to a .JAR file along with the setting file. To speed up the "setting up" process, we created a separate application to deal with the writing of the XML file, and the copying of all the files in to one .JAR file.

The result of the administrator and end user separation is that end users only have to learn a minimum set of skills to use the functionality of the HISP GIS while people with more advanced skills deals with the more advanced part of adjusting and setting up the application. Setting up HISP GIS

HISP GIS is traditionally organized in two Java archive (JAR) files. One JAR contains the application and the other holds maps, images and a configuration file. The name of the files are by default hispGis.jar and resources.jar. In addition to these two files, the application needs third party libraries. These libraries also come as JAR files and needs to be accessible to the Java runtime environment. The default configuration and setup of HISP GIS allows it to be started as both a desktop application and a JWS application without any form of reconfiguration.

This is the contents of the two JAR files:

hispGis.jar:	
`./images/gis-logo.GIF`	image used in the application
`./images/startup.PNG`	image used in the application
`./META-INF/HISP-GIS.DSA`	DSA certificate with information about the signature holder*
`./META-INF/HISP-GIS.SF`	digest information for all application files*
`./META-INF/MANIFEST.MF`	manifest file with listings for the attributes “Class-Path” and “Main-Class”
`./org/hisp/hispGis/[Java class application files for HSIP GIS]`	HISP GIS application files

resources.jar:	
`./[map files on the ESRI format with extensions DBF and SHP]`	maps used by HISP GIS
`./settings.xml`	the HISP GIS settings file created by the HISP GIS configuration program
`./META-INF/HISP-GIS.DSA`	DSA certificate with information about the signature holder*
`./META-INF/HISP-GIS.SF`	digest information for all application files*
`./META-INF/MANIFEST.MF`	manifest file without any specific configurations

* = this file is only present if the application has been digitally signed

For the time being HISP GIS depends on these third party libraries:

```
gt2-main.jar
gt2-shapefile.jar
jai_core.jar j
Charts-0.7.5.jar
JTS-1.4.jar
looks-1.3-snapshot.jar
mysql-connector-java-3.0.14-production-bin.jar
opengis-legacy-0.1.jar
opengis-tiger.jar
opengis.jar
RmiJdbc.jar
units.jar
vecmath.jar
```

In the previous chapter about the HISP GIS configuration tool, we saw how the resources.jar file was created and populated with map files and a configuration file. In the next two sections we look at what is required of setup in order to run HISP GIS as a JWS application and as a standalone application. Illustrations are given in appendix A.

5.6.3 Java web start

We already mentioned that HISP GIS needs to be digitally signed if it shall be able to access a database which is located on a different server than HISP GIS was downloaded. This means that both hispGis.jar, resources.jar and all the third party jar files must be signed by the same certificate. In order to ensure safety for the user, this certificate should be issued by a trusted third party such as VeriSign.

To create a certificate you can use “keytool”, a software included in Java Software Development Kit (SDK). First you create a new keystore with the command: “keytool -genkey -keystore [myKeystore] -alias [myself]”, and the you can create a self-signed certificate with the command: “keytool -selfcert -alias [myself] -keystore [myKeystore]”. When the certificate is made, you can sign all the jar files with the application jarsigner (also included in the Java SDK). The command for signing a jar file is: “jarsigner -keystore [myKeystore] [test.jar] [myself]”. For more information please see the documentation bundled with Java SDK.

Signing the application is not needed if the database is located on the same server. In that case you only need to configure the JNLP file already mentioned. This is an example of a HISP GIS JNLP configuration file:

```
<?xml version="1.0" encoding="UTF-8"?>
     <jnlp codebase="[www.website.com/location/of/jar-
files/]">
          <information>
               <title>HISP-GIS</title>
               <vendor>HISP</vendor>
               <description>Geographical Information
System for Health Indicators</description>
               <icon href="images/icon.gif"/>
               <offline-allowed/>
          </information>
          <resources>
               <j2se version="1.2+"/>
               <jar href="hispGis.jar"/>
               <jar href="resource.jar"/>
               <jar href="gt2-main.jar"/>
               <jar href="gt2-shapefile.jar"/>
               <jar href="jai_core.jar"/>
               <jar href="jCharts-0.7.5.jar"/>
               <jar href="JTS-1.4.jar"/>
               <jar href="looks-1.3-snapshot.jar"/>
               <jar href="mysql-connector-java-3.0.14-
production-bin.jar"/>
```

```
                <jar href="opengis.jar"/>
                <jar href="opengis-legacy-0.1.jar"/>
                <jar href="opengis-tiger.jar"/>
                <jar href="RmiJdbc.jar"/>
                <jar href="units.jar"/>
                <jar href="vecmath.jar"/>
          </resources>
     <application-desc main-
class="org.hisp.hispGis.application.Manager"></applica
tion-desc>
     <security>
          <all-permissions/>
     </security>
</jnlp>
```

For a full documentation of JNLP used in combination with JWS, we advise you too look in the Java software development documentation (java.sun.com). Here we will limit our comments to a few tags important when configuring HISP GIS:

`<jnlp codebase=”www.hisp.org/hispGis”>`	The codebase atribute defines the base location for all further file references. The location of resources are given relative to the codebase path.
`<offline-allowed/>`	This allows for HISP GIS to be executed locally by JWS after it has been downloaded once.
`<resources>`	All jar files needed by HISP GIS are defined here. This include hispGis.jar, resources.jar and all third party dependencies.
`<application-desc main-class=””>`	The “main-class” tag tells JWS what class contains the main method for executing the application.
`<security>`	By adding an “<all-permissions/>” tag, the user can grant the application access to disk and network resources.

After creating the JNLP file and the HISP GIS JAR files, you typically create an HTML page pointing to the JNLP file. With a client machine installed with JWS and a browser configured to handle the JNLP mime type, all you have to do is to point the browser to the JNLP link.

The next section describes how to make HISP GIS JAR files for executing the application without JWS. If you choose to digitally sign the JAR files after applying the configuration presented in the next chapter, you will have a distribution suitable both for standalone execution and JWS execution.

5.6.4 Desktop application

When using JWS, the JNLP configuration file tells the runtime environment about the library dependencies. This is done when listing the JAR files inside the resources tag. When running the application without JWS, the location of the JAR files must be given by the Class-Path parameter. To make this easy for the user you can configure the class path in the manifest file inside hispGis.jar. This is an example of a manifest file when all JAR files are located in the same folder as the application:

```
Manifest-Version:.0
Class-Path:resources.jar gt2-main.jar gt2-
shapefile.jar jai_core.jar jCharts-0.7.5.jar JTS-
1.4.jar looks-1.3-snapshot.jar mysql-connector-java-
3.0.14-production-bin.jar opengis-legacy-0.1.jar
opengis-tiger.jar opengis.jar RmiJdbc.jar units.jar
vecmath.jar
Main-Class:org.hisp.hispGis.application.Manager
```

In addition to setting the class path, the JNLP file takes care of telling the runtime environment where the main class is located. This is done in the application-desc tag. When running HISP GIS standalone the main class must be set in the manifest file. In the code example above, this is done in the last line by setting the parameter Main-Class.

After making hispGis.jar and resources.jar, and having configured the manifest file, the application can be executed with the command: java -jar hispGis.jar.

5.7 Improving HISP GIS

We are in general very pleased with HISP GIS. Most of the functionality requested has been implemented, and the system design and programmatic design is satisfactory. We do however see many possibilities for improvement, both to the systems design and to the number of available functions. The next two sections is about of such changes.

5.7.1 Needed changes for HISP GIS

There are many system changes that we would like to do. In the interest of future development of HISP GIS, we have therefore made a list of important improvements. The changes that we consider will increase use and usability the most, are listed at the top:

1. Internationalization
2. New GUI for the HISP GIS configuration software
3. Fully implement Java Logger

Internationalization

Textual elements such as status messages and the GUI component labels should not be hard coded in the program. Instead, they should be stored outside the source code and retrieved dynamically. Internationalization (i18n) is the process of designing an application so that it can be adapted to different languages and regions. By moving text messages such as status messages and GUI component labels out of the source code, this can be achieved. HISP GIS, all text messages meant for the user is hard coded in English. All these messages should be moved outside the source code.

Java has very good support for internationalization through the use of resource bundles. By placing all textual messages in resource bundles, the applications language can be configured in the settings file or it can be chosen inside HISP GIS.

The reason why we did not implement internationalization was prioritizing of developing time. We did not want to spend time on something which was simple to refactor later. Therefore we would now like to go through internationalization and how it can be applied to HISP GIS.

Internationalization is done by creating Java property files. One file for each supported language. The name of these files should be on this format: “MessagesBundle_fr_FR.properties”. The first part of the file name, “MessagesBundle”, you can choose yourself. The second part, “_fr_FR”is reserved, and indicates what language and country the property file is written for. The suffix of the file name is the standard suffix for Java resource files.

This means that the given example, “MessagesBundle_fr_FR.properties”, is a Java resource file named MessagesBundle which contains text strings for France (FR is a country code) in the language French (fr is a language code). Resource files are placed along with the rest of the program's class files, so that they are accessible by the runtime environment. Lets look at an example of how a resource file can be used. This is the content of our resource file “HispGisMessages_en_EN.properties”:

```
newMapLoaded = A new map has been loaded
mapsLoaded = {1} {2} from {3} has been loaded
newChart = Create chart
```

The simple way of showing text messages is to hard code it like this:

```
Button newChart = new Button(“Create chart”);
```

If you want the user to choose the language of the GUI, this is the way to do it:

```
Locale currentLocale = new Local(“en”, “EN”)
ResourceBunde messagesFromPropertyFile =
ResourceBunde.getBundle(“HispGisMessages”,
currentLocale)
Button  newChart = new
Button(messagesFromPropertyFile.getString(“newChart”))
;
```

Now all you have to do is to implement functionality so that the user can choose language file. Alternatively you can use Locale.getDefault() to get language and nationality from the computer setup.

The previous example was showed a simple text string. It is however also important to internationalize culturally-dependent data such as dates and currencies. A more complex string example is in the second line of our property file:

```
“mapsLoaded = {1} {2} from {3} has been loaded”
```

This line has tree numbers which can be substituted at runtime. Here are three examples of how the string could look:

1. 1 map from Norway has been loaded

2. 1 map from India has been loaded

3. 5 maps from India has been loaded

The DHIS software is used in many different countries and supports several languages. The spatial application should do the same. Supporting different languages is important for usability. In addition it appeals to the goodwill of the user.

New GUI for the HISP GIS configuration software

The configuration program is meant for administrators and never intended for the end user. For this reason, the GUI for the HISP GIS configuration software was not prioritized during development. The prototype GUI should as soon as possible be completely redesigned. Although it is the end user of the spatial module who has the least knowledge about computers, and therefore needs the best user interface, there is no doubt that the HISP GIS configuration program needs to be easier to use. Moreover, administrators are often consulted when deciding on software and it is likely that they have some focus on what is convenient for them.

A good thing about the configuration software is that it supports all needed configuration of HISP GIS through it's GUI. No manual setup is needed. It is only the user interface that has to be improved. It should not be a complicated job.

Fully implement Java Logger

HISP GIS has partly implemented logging. For this we have used Java's own Logging API. The package name of this logger is java.util.logging. Information about Java's logging implementation can be found in its API and in the documentation that comes with Java Software Development Kit (java.sun.com).

This is the first logging done by HISP GIS:

```
Logger.getLogger( "hispGis" ).info( "Starting
application" );
```

This logging statement has a message: «Start application», and a status: «info». The logging can be configured with a configuration file. Based on the message status and the configuration file, the log messages can be printed to the consol or appended to a log file.

Logging is useful for many reasons:

1. It makes it easier for an administrator to discover wrong configuration
2. It eases development by making the application actions more transparent
3. Makes it possible for the user to send bug reports
4. Can provide useful insight for the end user

More importantly it can:

- Reveal wrong or illegal use
- Show use statistics/patterns for the applications functions

5.7.2 New functionality in HISP GIS

We have made a list of functionality that we did not manage to implement to HISP GIS.

- Charts
    - Changing the color of legends. Now all colors are generated at random
    - Manually setting the interval of the y-axis
    - When a date is selected instead of a time period, the organization unit or the indicator should be used as label on x-axis
    - The suffix should be added automatically when saving chart as images
- Maps
    - It should be possible to have map reports showing the sum of a time period. Now it can only show single dates or every date in a time period.

    - The placement of names for geographical regions is not optimal. First of all, GeoTools has not implemented functionality for avoiding overlapping of text features. Second, the text is always placed at the mass-center of a region. This sometimes results in names being places outside the region (for instance if the region is shaped as half-moon.

    - Forest areas (regions which does not belong to any health regions) should be shadowed

    - Support for topological layers

- map legend

    - Manually set interval ranges. Now you can only choose the number of intervals

- Application

    - Change data source inside HISP GIS. This could be convenient for advanced users. Default setup could be stored in the settings file. Then all changes could easily be restored.

    - Maps should be restricted.

    - It should be possible to restrict access to the database. Maybe users of one region should not be able to look at other regions

- Sessions

    - At present one can choose to save all queries made to the database in a session file. On our developing list we also had the ability to save all charts in the chart folder. This is easy to implement, but was not prioritized.

5.8 Problems

5.8.1 Solved problems

The single issue that stole most time from our development, was a simple problem of data format. We discovered the problem when trying to display dates from the DHIS database in HISP GIS. To make a long story short, we did not know the format of MS Access "short date integer". After hours of trial and failure we discovered that the format counted number of days from 30. November 1899. This meant that the result "36800" will be a date in the middle of year 1999 (36700days/365days=100.58 years). The work of finding this error in our application, combined with finding the date 30.11.1899 took us almost two days. Now, what's really interesting was to find out that we where not the first one to meet this problem. A member of the Indian HISP team later told us that he used the same amount of time solving the issue. Also, when we came back to Norway we talked to developers of the DHIS2, and he had spent even longer to figure out this problem.

These statements indicate that there is a great problem of documentation for the HISP project. To confirm this, we have tried to find documentation about the DHIS software base. We have however not been able to find any DHIS documentation. Several DHIS persons have also confirmed this fact.

We are certain that the people behind the next DHIS, version two, will have more conscious relation to documentation and sharing of knowledge. Hopefully the newly established HISP server (hisp.info) can be a base for this. Important is also sharing of knowledge through developer mailing lists. DHIS has such mailing lists.

5.8.2 Known bugs

These are known bugs in HISP GIS:

- The map legend is barely visible after startup. It should not be visible at all at this time. After a map report has been generated and removed, the legend disappears completely as it should.

- When the application is rescaled, the graphs paint themselves on the application frame and scroll bars.

5.9 User Manual

We have not created a user manual. This was part of the agreement with HISP India. As a part of learning HISP GIS they shall write a user manual as they see fit.

6. DISCUSSION

6.1 GIS' in DHIS: Opportunities, Requirements and Limitations

6.1.1 GIS in the Indian health system

Walsham (2000) states that, typical Indians rarely use maps in daily life, and that they in general do not use maps as key aids to spatial awareness. According to our observations, the use of maps in Indian health facilities differed a lot (4.3). The Kerela case, where maps were actively used (4.3), is however an exception to Walsham's observation. The Kerela workers used maps daily for both planning and analysis. Tasks which they used maps for included analyzing the spreading and density of deceases, making action plans to fight deceases and planning daily work and field work. This observation of fairly advanced use of maps is relevant to the discussion of health workers geographical needs, and the scope of a GIS application. It shows that the health workers both want to, and are able to, systematically analyze their own gathered health data in a spatial context.

Several geographical analysis methods are outlined by Cromley and McLafferty (2002) and Chung, Yang and Bell (2004) for dealing with health data. Some of these methods are simple geographic methods for measuring distances while others look at relations between different data types. Some also include fairly advanced statistical functions - such as various kinds of smoothing, regression and correlation. Using the needs of the Kerela analysts and planers as a starting point, we want to review some analysis functions. The maps in Kerela were used for:

1. Analyzing the spread and density of deceases

2. Linking workers to areas of responsibility

3. Planning routine work

Measurement analysis, network analysis and topological analysis are primary tools for working with spatial data (2.3). Measurement analysis calculates the distance between two points, network analysis calculates the length of map features such as roads and rivers and topological analysis looks at relationships between different map features.

The geographical distance between points, which is calculated by measurement analysis, forms the basis for many spatial functions. In the Kerela case, it necessary for manually measuring the spreading of deceases. Network analysis is useful for planning routine work as it can calculate the length of a route through a network of streets and roads. Topological analysis can be used both for decease analysis, by analyzing the relationship between deceases and surrounding topography, and for administrating health regions by looking at how many villages that is near a health facility.

More advanced functions are tools that combined the functions above with statistical methods, or calculate area sizes and spreading automatically, instead of just providing the user with raw numbers of distances and lengths. The tools for automatically calculating the size of decease areas are easy to implement, and would replace manual work done at the Kerela facilities.

Statistical methods (2.4) where not used by the analysts we observed, but we would like to look at some of these tools anyway. The statistical methods described in chapter 2.4 looks at trends and relationships in the health data data. Clustering tools looks for unusual density of health cases (Cromley and McLafferty, 2002), spatial regression analysis helps understand the relationship between the variables of spatial phenomenon (Chung, Yang and Bell, 2004) and spatial autocorrelation indicates the degree to which a distribution is influenced by a phenomenon (ibid). The results of such analysis, which reveals trends in the spreading of diseases and relations between different health data and topological data, is helpful for making efficient and useful action plans for future health work.

Our observation of extensive use of maps, makes us believe that if the functions mentioned above was made available, they would be used. This is encouraging to the future development of HISP GIS. Further, we have so far only been looking at use at PHC and Sub-center level. Manual inspection of health data, which was done at the health offices in Kerela, is easy to do for small numbers of data. The higher levels have however a much larger amount of aggregated data. This gives a large potential for decreasing manual work. With computer aided spatial analysis tools, they could cover large regions fast and accurate, with little need of manual work.

What we have so far not been looking at from the Kerela case, is the job of linking workers to areas of responsibility. This involves combining health data with employee data and is partly outside the scope of a geographical health application. One solution is however to generated maps with the needed health data and topography, and then making use of printouts for planning. If the geographical application supported it, one could also make custom map layers with the workers areas of responsibility. The case of combining different data sources, like employee data and health data, in a GIS application, is the topic of the next paragraph as we look at our observations and the prospect for integrating data from other disciplines than public health care.

The potential of a GIS for public health depends much on the sharing of data between departments and public offices (Richards et al, 1999). Regarding this multi-disciplinary use of GIS, there are some challenges which are specific for the Indian context. Puri (2003b) mentions how the organizational challenges caused by a top-down approach in the health structure, and the strict cultural and religious grouping of people makes a common dialog difficult. Walsham (2000) also comments that this compartmentalized society, where people have little or no interest in others work, makes co-operation difficult. The only experience we had of this, is the statement of from Kerela workers saying that tuberculosis reports would be littered if delivered to the wrong office (4.3). This witness of little interest for the work of others, and is in our opinion a barrier for enabling the sharing of data and knowledge on cross of departments.

A different obstacle for enabling sharing of data and utilizing the skills on cross of departments is the combination of poor communication infrastructure and large distances. This is especially relevant to the Indian case where we saw a strong decentralization of the primary health care services. The decentralization meant that PHC and Sub-centers were located far away from other public offices and departments. With the need of sharing information manually, due to a lack of internet connections, this decentralization can make multi-disciplinary use of GIS difficult.

6.1.2 Integration of GIS in DHIS

In the previous sub-section, we looked at some tools for meeting the demands of the health workers in Kerela. These can be implemented into HISP GIS. The library that provides the map functionality for HISP GIS, GeoTools, does however not provide this functionality, so all functions must be coded by the HISP GIS developers. We do however see before us possibilities for avoiding this programming job.

First of all the expansion of free databases such as MySQL and PostgreSQL (with its PostGIs extension) to include spatial queries will provide much of the functionality needed for geographical analysis (2.7). Second, we believe there will come more free open source tools in the near future which provide spatial analysis functions. One which tool it might be interesting too look at is JGRASS which aims at porting GRASS to Java.

We will now look at some options for future HISP GIS. One option is to leave HISP GIS as it is and not implement support for computer aided analysis. This would imply a clear separation between a GIS for exploring, visualizing and manual inspection, and a GIS for spatial analysis. HISP GIS would then only enable use of topological layers for manual inspection and print out of maps for various simple planning purposes.

The next option is to integrate geographical functions such as measurement analysis, network analysis and topological analysis, but leave out spatial analysis In other words limit the toolkit to functions supported by a spatial database. This way some of the needs outlined by the Kerela health workers could be satisfied, while still limiting the programming work. This does however mean that the health data from DHIS must be exported to a spatial database.

A third option would be full integration of geographical analysis and spatial tools. We do however not see this as realistic considering what an extensive task this would be. Instead we believe a connection to other open source GIS tools would be preferable. This could either be done by using to separate software, such as in the first option: a clear separation between a GIS for exploring, visualizing and manual inspection, and a GIS for spatial analysis, or by linking other tools through program interfaces so they can exchange information. One problem outlined by Chung, Yang and Bell (2004) however is that there are very few geographical tools which includes meaningful spatial statistical analysis for the health care context.

Designing a full featured geographical system is a serious task since one can expect that many health decision will be taken on the basis of results given by the software. Most important is securing patient confidentiality, avoiding misuse and getting proven right result from the spatial and statistical functions (Richards et al.,1999). This is not a given, but depends on good spatial methods, well trained health analysts and above all good software. The complexity of spatial functions therefore makes us emphasize the need of quality software. If this quality can not be guaranteed, we suggest finding well documented libraries with spatial functions, or using good open source geographical software in combination with a HISP GIS. In the case of using ready made GIS packages, some kind of integration would be preferable in order to avoid exporting and importing of data from HISP GIS.

The issue of using multiple data sources for health analysis, as explained by Richards et al. (1999) was not an issue when we developed HISP GIS. Partly because it was beyond the scope of a prototype such as HISP GIS, and partly because there was no other data available than the health data from the Indian DHIS database. As even basic geographical analysis are not yet covered by HISP GIS, and in the light of the many problems revealed in the previous sub-section concerning multi-disciplinary use of GIS in India, we think limiting HISP GIS to only using health data would be right for the time being.

On the other hand, we note Richards et al's (1999) point that GIS can be an enabling technology for forming new partnerships at community level. As the Kerela health workers expands their use of map use, it is likely that they will see the need for data and knowledge from other departments than those from public health, and then motivate multi-disciplinary GIS use.

Richards et al's (1999) also makes an interesting point of that multi-disciplinary use of GIS as a source of new working procedure and analyzing methods. This he says, is positive since it makes it possible to get more information out of the health data, but he also express a concern for the possibility of neglecting traditional epidemiological principals for the studying of health data. To meet this problem, we suggest a close co-operation with health workers already using maps in their work.

When looking at the possibilities and constraints of HISP GIS, it is also important to consider how the software could fit as a module into the rest of the DHIS software package. Important issues are integration analyzing tools such as pivot tables, and integration with report tools. Primarily we see before us that pivot tables and report generators can request image files with customized map data. One could for example select data elements, organization units, and time periods in the pivot table software and the request a map with the same data. The user could then be confronted with options of topologic data and then presented the map as an image. The same could be done for reports in the report software. Maps from HISP GIS, tables from pivot tables software and raw data from the database could be combined to form informative and presentable analysis.

It could however also be useful to work the other way around, using HISP GIS as a navigation tools for selecting regions and data and then have the option of continuing the inspection work in pivot tables, or other suitable analysis tools. Just how the integration should be done technically is a question concerning the design of DHIS as a hole and what options that are provided by the DHIS framework. We do not want to go into this discussion, but find it valuable to mention the choice of using Java Web Start for HISP GIS. This choice of platform might prove disadvantageous if the future components of DHIS aims at a tighter integrating to web through standard html technology. We then see before us a separation between desktop GIS and web GIS so that maps can be provided to web users, and geographical analysis tools to desktop users.

Last, we would like too look at some constraints to GIS that relates to the way the DHIS database is organized. The problem is that there are two different DHIS database files, one for PHC and one for the rest of the health levels (3.4.1). This implies that only the health workers on PHC level can analyze health data referenced to Sub-centers. Health workers on the levels above, which uses a database with data aggregated only from PHC level, don not see Sub-center details. This constitutes challenge for geographical analysis as a lack in precision limits the uses of the data (Chung, Yang and Bell, 2004).

Even detailing to Sub-centre level is not very precise. According to Chung, Yang and Bell (ibid), most spatial analysis tools require point data, and the accuracy for point data should at least be on street level. As one sub-center usually covers one or more villages, the data recorded in the DHIS database is polygon data, not point data. In other words, there is no information about where health cases origins, one can only track health events to the sub-center where it was registered.

The fact that health cases not are registered to persons living at a specific location, also makes it possible for people to use other sub-centers the one they “belong to”. If they prefer a doctor at a sub-centre outside their own village, or use doctors provided by their workplace, those health cases will not be mapped to the place they live. This is a source of error when analyzing health data.

6.2 Challenges of Open Source GIS development

Weerawarana and Weeratunga have emphasized six basic requirements for open source development in developing countries (Weerawarana and Weeratunga, 2004). A part of the ideology of HISP is to increase local knowledge and skills in order to decentralize the health system, and gain improved efficiency (3.1). By scaling down Weerawarana and Weeratunga six requirements and put them in to our HISP India context, we will discuss each requirement in relation to our experience and observation.

The first of the requirements is the need for a law framework to protect intellectual property. The motivation for this is the widely spread pirate software which undermine the economical benefits of open source development (Weerawarana and Weeratunga, 2004). The problem of pirate software were something we observed many times during our stay in India (4.4). All the computers we saw were running MS Windows, and none of the users we talked to had legal licenses. This indicates a need for a stricter legal framework and/or a stricter enforcement of existing laws (Weerawarana and Weeratunga, 2004), especially if India shall be able to fulfill their commitment to impose intellectual property regulations (Nagaraj, 2002). The government of India has since 2002 chosen to promote the open source operating system Linux as the 'platform of choice' instead of the proprietary system Windows (ibid.). The intellectual property regulation issue was seen as the main focus for this transition to Linux. We did however not see any systems running Linux, neither on private nor public systems. In fact, most of the newly educated computer master students which we interviewed had never heard about Linux (4.4). We have to consider the fact that the open source plan of the Indian government is ambitious, and to set it in to full action will take time and effort.

In HISP context open source has been used for developing DHIS (3.2). But even though DHIS is open source it is built on a proprietary system and can only be used in MS Windows (ibid.). For users to legally use the DHIS they need to have registered licenses for both MS Windows and MS Access. But because of al ready limited health budgets in developing countries, the use of piracy software is preferred over purchasing licenses (4.4). Since the users have to pay for licenses the DHIS can not be called free software even though it is open source. In our opinion the proprietary dependent DHIS to a certain extent contribute to undermine the potential of open source to succeed in developing countries. Much of the work conducted by HISP is to educate personnel to develop and work on the DHIS software (3.1). This mean that education of developers and user of DHIS, before the work on DHIS2 started, has led people to relay on a proprietary system, and also stimulated to the use of piracy software.

The DHIS2 will be a truly open source and free system (3.2), and HISP will promote the use of Linux in the health system to run DHIS2. This will probably in the end solve the problems of stimulating to use of piracy software within the health systems using DHIS, but much effort must be used on reeducation of DHIS developers and users before that is a fact.

The second requirement listed by Weerawarana and Weeratunga is the need for low cost, widely available Internet access. During our project we experienced problems and delays related to the absence of internet access (4.2). Despite the problems, we managed with just a few people to create an open source GIS application. There are probably two reasons for this; the application is small and only has a few basic functionalities (5.3), and we did not need internet to cooperate implementation between developers because all participants worked on-site (4.1). Further development of the system will however depend on increased integration with DHIS. This integration requires communication between developers at the different sites in order to manage how data and methods are exchanged between the different modules. If internet access is not widely available to the developers, the ability to participate in such an open source project will be severely limited (Weerawarana and Weeratunga, 2004) This is not critical just to further development of HISP GIS but open source development in general.

The third requirement listed by Weerawarana and Weeratunga is the need for educational infrastructure. A network of good educational institutions is needed to teach basic skills and to stimulate and disseminate the use of open source systems. We did not observe such an educational infrastructure, this does not mean that such educational institutions do not exist, but our observation might indicate that the educational infrastructure in India basically consist of institutions basing their training on MS Windows;

MS Windows had 94 % of the desktop market when the Indian government initiated their open source plan in 2002 (Nagaraj, 2002). We observed training courses for teaching the use of MS Office, but none for teaching in open source systems such as MS Offices' open source counterweight OpenOffice.org This training is the right opposite of Weerawarana and Weeratunga third requirement of education in open source; it increases the use of MS systems in stead of open source. It also conflicts with the Indian plan of emphasizing the use of Linux, but when Microsoft has such an impact on the Indian desktop marked, it is understandable that users want to learn a system they can use elsewhere. This "strong relation" to Microsoft systems is also reflected thru higher computer educated people. During the job interviews we conducted with Indian computer scientist we asked questions in relation to knowledge of open source and what kind of software systems they had used during their education (4.4). Just two out of ten of the interviewees had heard about Linux, and only one of them had actual seen a computer running Linux. All their training was conducted on computers running MS Windows.

We were told that the main motivation for studying computer science was to get a job in one of the larger computer companies in India (4.4), because a job in one of these companies could bring great prosperity in form of salaries from two or three times more the average income to more than forty times the average income. People suggested that educational system was educating people using MS systems because they think that this is the most commonly used system in the commercial companies, and that they are educating people for this companies (4.4). But the way we see it, an extended use of Linux in educations would not directly conflict with the educational institutions target of create scientists for the commercial companies. All the people we interview had good or average knowledge in Java (ibid.). Java and other platform independent programming languages can be thought on Linux systems. Platform dependent systems has to be thought on supporting platforms, but when people teaches skills that can be trained using Linux as well as MS Windows, we think the educational institution should choose Linux to support the governments' transition plan. This would also correspond to third requirement stated by Weerawarana and Weeratunga which emphasize that in education on a higher level it is important to educate developers in open source development techniques, because this will develop a culture for the use of open source systems (Weerawarana and Weeratunga, 2004).

If the lack of Linux and open source systems in educational institutions is a result of believes that these systems are second best over proprietary systems, and therefore educational training is best done on MS systems it can seem difficult to pursue the use of open source systems. During our project we observed an increased interest for software engineers that had knowledge in open source libraries (i.e. Hibernate) and systems (i.e. Linux and Apache Tomcat) (2.7). As we see it, if the demands for open source developers become a trend, then the request for open source engineers probably will influence the educational system to teach open source technology. And the increased knowledge among higher educated people will, according to Weerawarana and Weeratunga, in the end result in higher awareness of open source systems among the rest of the public (Weerawarana and Weeratunga, 2004).

The fourth and fifth requirement for use of open source listed by Weerawarana and Weeratunga is freedom of information and English-skilled developers. The freedom of information, they say, is related to the access of internet. If people are denied access to Internet open source development cannot succeed in that country (Weerawarana and Weeratunga, 2004). In India we did not experience any regulation of access to Internet. To us this reflects the governments wish to prepare and make it possible for open source development.

The importance of English-skilled developers is based on the programming languages and on the language used for exchange information thru Internet. India has 23 official languages, and among these is English (Wikipedia, undated), which is spoken more or less all over India. During our project we traveled around in different state in company with HISP India employees (4.3). The languages would change from state to state but also within states the language would change and several languages was spoken. When the Indians did not speak the same mother language English seemed to be preferred. We were also told that school starts early in teaching people English, not just in English classes, but English was used as the spoken language in all classes. In our opinion, the general English knowledge in India makes it easier for Indians to learn programming languages compared to countries where English knowledge is poor, because the *lingua franca* in programming languages is English. The same could be said about communication with other developers thru Internet as well (Weerawarana and Weeratunga, 2004).

We think that the issues connected to freedom of information and English skilled developers in India is of minor concern, because, according to our observation, the access to information on Internet is not restricted, and the average Indian master English well.

The last requirement listed by Weerawarana and Weeratunga is the need for a skilled or trainable developer pool. This can be related to the educational infrastructure where the skills of open source development are taught, and where the trainable developer pool is needed for participation in open source development (Weerawarana and Weeratunga, 2004). If the country does not have a skilled or trainable developer pool, then it is not feasible to participate in open source development (ibid.). Even though Weerawarana and Weeratunga talks about a trainable pool on a national scale, we see that need for a trainable pool also apply to the work of HISP, for instance, the development of DHIS2 where people with good knowledge of open source development is used to build up, and train other developers in the HISP network to participate in the open source development of DHIS2 (3.1).

During our project we encounter issues that can be related to the requirements of a skilled or trainable developer pool and the requirement of an educational infrastructure. The HISP GIS project required a lot of effort to gather information about different open source technology and to gain knowledge in how to incorporate such technology in to an application. By working in a team this effort could be shared among the team members. Gathered information could then be discusses within the team to agree upon the best solutions. The project benefited from the HISP India members knowledge of the Indian health system and DHIS, but the lack of Java knowledge among the team members was a big constraint for the project. The lack of Java skills did not only mean that they could not participate on discussing the technology, but we had to spend time in the limited project period to teach them Java (4.1). The fact that we, alone, had to do most of the work on HISP GIS, and that we would not be able to pass on our knowledge of the application to the Indian HISP team, because they did not understand common base of Java terms and programming concepts, became the biggest challenge in our project.

We think that, the lack of Java skills indicates that one of these two mention requirements (educational infrastructure and skilled pool) is not fulfilled. HISP India has a trainable pool in form of all the developers. But the limited knowledge in Java, which is the chosen language for the DHIS2, indicates that HISP do not have a good enough educational infrastructure to take the full advantage of open source.

6.3 Summary

In the discussion above we have looked at available geographical and statistical analysis tools and the map needs of Indian health workers. These functions have then been the source for a discussion of which tools to enable in HISP GIS and what the different options for implementation are. We have not pointed to a preferred solution, but tried to highlight the challenges and benefits of the different options. We have also discussed the challenges tied to multi-disciplinary use of GIS and how it relates to the Indian context.

We feel the discussion show that the implementation of a GIS application for DHIS integration is a difficult task. The developers will have to decide what data to include, what tools to implement, how to support web and desktop users and how to connect the system to other DHIS modules.

In the section about open source software we have made our discussion around six requirements for open source development in India. Each of these requirements we could apply to our project context, and it reviled that some of the issues we encounter during the project can be considered as open source issues that are critical for open source development.

7. Conclusion

In the previous chapter we have used our observations and experiences from the development process in an effort to review use of GIS and open source in India. We have tried to balance the discussion by looking at technical, practical and human aspects. We have attempted to underline complexity of GIS by providing a technical discussion combined with interesting observations from our visits in the field. For the open source discussion we have tried to make much use of our experiences from the development process. These discussions have enlightened our views on HISP, GIS and open source.

We feel that it is difficult to draw any general conclusions from the discussion, because our period of observation lasted for only two month. We do however think that some of issues reflected in the discussion can be of value for further work with the HISP GIS application, and open source GIS development in general.

Table of authorities

AFT Project, 2003
"Arab Fertility Transition Project"
Article retrieved from official web site of: Diego State University, Department of Geography.
[Visited 20.05.2005] Available at:
http://geography.sdsu.edu/Research/Projects/Aftweb/ProjectDescription/GIS.htm

Braa and Hedberg, 2002
Jørn Braa and Calle Hedberg
"The Struggle for District-Based Health Information Systems in South Africa"

Berg and Lewis, 2003
"Group report and system description – HISP India GIS Project"
[Visited 23.05.2005] Available at:
http://heim.ifi.uio.no/~morberg/public/P3_5230.pdf

Calla and Koett, 1997
Jason Call and Richard Koett
"GIS Implementation at the Squamish Nation"
[Visited 20.05.2005] Available at:
http://www.innovativegis.com/papers/sngis97/sngis97.html

CEN, undated
[Visited 20.05.2005] Available at:
http://www.centc251.org/Ginfo/Glossary/tcglosh.htm

Chung, Yang and Bell, 2004
Chung Kyusuk, Yang Duck-Hye and Bell Ralph
"Health and GIS: Toward Spatial Statistical Analyses"
Journal of Medical Systems, Vol. 28, No. 4, August 2004

CMC, undated
"Family Welfare, Health Information and Monitoring System"
Article retrieved from the official web site of: CMC Limited
[Visited 20.05.2005] Available at:
http://www.cmcltd.com/industry_practices/e-Governance/healthcare/fhims.htm

CNET Asia, 2003
"Trying step by step to eliminate Microsoft"
Article retrieved from official web site of: silicon.com.
[Visited 20.05.2005] Available at:
http://www.silicon.com/management/government/0,39024677,39116719,00.htm

Cohn, Ghahramani and Jordan, 1996
Davis A. Cohn, Zoubin Ghahramain and Michael I. Jordan
"Active Learning with Statistical Models"
Article retrieved from official web site of: Carnegie Mellon School of computer science
[Visited: 20.05.2005] Available at:
http://www-2.cs.cmu.edu/afs/cs/project/jair/pub/volume4/cohn96a-html/statmodels.html

Cromley and McLafferty, 2002
Ellen K. Cromley and Sara L. McLafferty
"GIS and Public Health"
The Guilford Press ISBN:1-57230-707-2

CUNY, undated
"Computer Crime Mapping: A Collaborative Project"
Spatial autocorrelation and spatial regression explained
Article retrieved from the official web site of: The Graduate Centre, CUNY
[Visited 20.05.2005] Available at:
http://web.gc.cuny.edu/Cur/Projects/Nij/Spatial2.htm

Deitel and Deitel 2001
Paul J. Deitel and Harvey M. Deitel
"Advanced Java 2 Platform: How to Program"
Prentice Hall, ISBN: 0130895601

ESRI, 1998
"ESRI Shapefile Technical Description - An ESRI White Paper"
Article retrieved from official web site of: Environmental Systems Research Institute Inc.
[Visited 20.05.2005] Available at:
www.esri.com/library/whitepapers/pdfs/shapefile.pdf.

Foote et al., 2000
MANGLER FORFATTERE
"Geographic Information Systems as an Integrating Technology: Context, Concepts, and Definitions"
Article retrieved from official web site of: University of Colorado at Boulder.
[Visited 20.05.2005] Available at:
http://www.colorado.edu/geography/gcraft/notes/intro/intro.html

GPL, 1991
"GNU General Public License"
[Visited 20.05.2005] Available at:
http://www.linux.org/info/gnu.html

HISO India, undated
Article retrieved from the official web site of: AP State AIDS Control Society
[Visited 20.05.2005] Available at:

http://www.apsacs.aphealth.org/index_files/Page416.htm

HISP, 2005
"DHIS 1.3"
[Visited 20.05.2005] Available at:
http://www.hisp.info/confluence/display/DHIS1/DHIS+1.3

HISP, 2005b
"DHIS-2 Modules"
[Visited 20.05.2005] Available at:
http://www.hisp.info/confluence/display/DHIS2/DHIS+2.0+modules

HISP India, 2004
"Software Enhancement"
[Visited 20.05.2005] Available at:
http://healthgis.org/hispindia/index.php?option=com_content&task=view&id=48&Itemid=1

Karlsson, undated
Anders Karlsson (sales engineer, MySQL AB)
"GIS and Spatial Extensions with MySQL"
[Visited 20.05.2005] Available at:
http://dev.mysql.com/tech-resources/articles/4.1/gis-with-mysql.htm

Lippeveld and Sauerborn ,2000
"A framework for designing health information systems" in Lippeveld, T. et al.: "Design and implemetation of health information systems"

Maguire, 1991
Maguire, D.J.
"An Overview and Definition of GIS. In: D.J. Maguire", M.F. Goodchild and D.W. Rhind (Editors), *Geographical Information Systems*. John Wiley & Sons, Inc, New York, pp. 9-20.

Mark et al. ,1997
"The GIS History Project"
David M. Mark, Nicholas Chrisman, Andrew U. Frank, Patric H. McHaffie and John Pickles
Article retrieved from the official web site of: University at Buffalo The State University of New York, Department of Geography
[Visited 20.05.2005] Available at:
http://www.geog.buffalo.edu/geography/

Nagaraj, 2002
Sudha Nagaraj
"Open IT: Govt to rewrite source code in Linux"
Article retrieved from the official web site of: "The Economic Times"
[Visited 20.05.2005] Available at:
http://economictimes.indiatimes.com/cms.dll/articleshow?artid=24598339

Opengeospatial.org, undated

"History of OGC (brief version)"
Article retrieved from the official web site of: Open Geospatial Consortium.
[Visited 20.05.2005] Available at:
http://www.opengeospatial.org/about/?page=history

Openknowledge.org, undated
"A Brief History of Free/Open Source Software Movement"
[Visited 20.05.2005] Available at:
http://www.openknowledge.org/writing/open-source/scb/brief-open-source-history.html

Pohlmann, 2005
Frank Pohlmann
"Find your way around open source GIS"
Article retrieved from the official web site of: IBM
[Visited 20.05.05.2005] Available at:
http://www-128.ibm.com/developerworks/opensource/library/os-gis/

Puri, 2002a
Satish.K.Puri
"Building networks to support GIS for land management in India: Past learnings and future challenges"
Published in proceedings of the international federation of information processing(ifip) WG9.4 Working conference on ICTs and socio-economic development: balancing global and local priorities, S Krishna and S. Madon (eds.) Bangalore, India, May 2002, pp 402-413

Puri, 2002b
Satish K.Puri
"Participation in GI for land management in India: problems and prospects, published in Electronic Proceedings of the International System Research Seminar in Scandinavia (IRIS' 26), S. Laukkanen and S Sarpola (eds.), Haikko Manor, Finland, August 9-12, 2003.

Puri, 2003b
Puri K. Satish
"The Challenges of Participation and Knowledge in GIS Implementation for Land Management: case studies from India", University of Oslo, Institute of Informatics, 2003

Puri and Sahay, 2003a
Satish K. Puri and Sundeep Sahay
"Participation through communicative action: a case study of GIS for addressing land/water development in India"
Accepted for publication in the Journal of Information Technology for Development in October 2003.

RHINO, 2002
"The Potomac Statement and Agenda for Investment on the Collection and Use of Routine Health Information in Developing Countries", page 2

Richards et al.,1999
Thomas B. Richards, Charles M. Croner, Gerard Rushton, Carol K. Brown and Littleton Fowler
"Geographic Information Systems and Public Health: Mapping the Future"
Public Health Reports 1999;114:359-373
Article retrieved from official web site of: National Cancer Institute (Long Island)
[Visited 20.05.2005] Available at:
http://www.healthgis-li.com/library/phr/maintext.jsp

Sæbø og Titlestad, undated
Johan Sæbø and Ola Titlestad
"Evaluation of a Bottom-up Action Research Approach in a Centralised Setting: HISP in Cuba"

Wagner, 2005
"Find your way around open source GIS"
Article retrieved from official website of: FreeGis.org
[Visited 20.05.05] Available at:
http://freegis.org/database/viewobj?obj=578

Walsham, 2000
Geoff Walsham
"Making a world of difference: IT in a global context"
John Wiley & sons, LTD, ISBN 0-471-87724-7

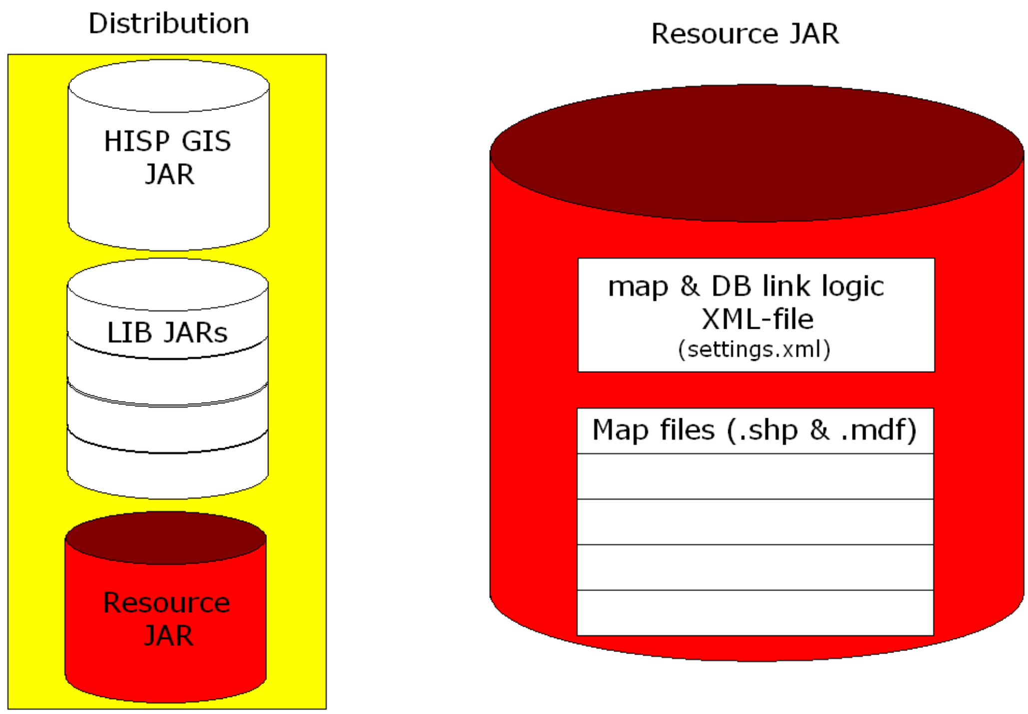
Weber, undated
"Open Source Software in Developing Economies"
Article retrieved from official web site of: Social Science Research Council.
[Visited 20.05.2005] Available at:
http://www.ssrc.org/programs/itic/publications/ITST_materials/webernote2.pdf

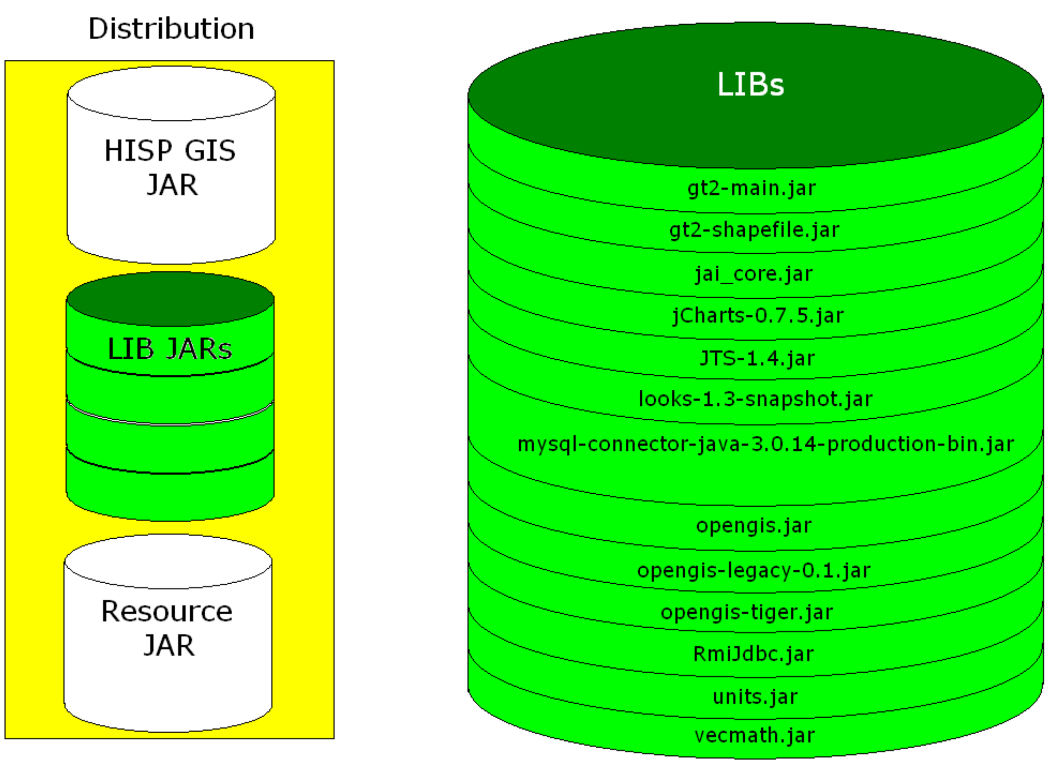
Weerawarana and Weeratunga, 2004
Sanjiva Weerawarana and Jivaka Weeratunga
"Open Source in Developing Countries"

Wheeler, 2005
David A. Wheeler
"Why Open Source Software / Free software (OSS/FS, FLOSS, or FOSS)? Look at the Numbers!"
Article retrieved from personal home page of: David A. Wheeler.
[Visited 20.05.2005] Available at:
http://www.dwheeler.com/oss_fs_why.htm

Wikipedia, undated
"India"
[Visited 20.05.2005] Available at:
http://en.wikipedia.org/wiki/India

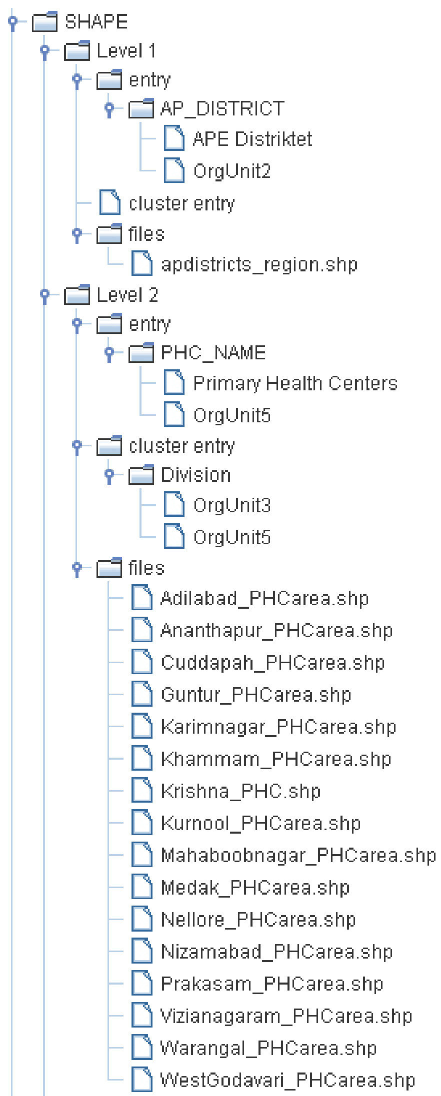
Appendix A:


Distribution
HISP GIS JAR
LIB JARs
Resource JAR
Resource JAR
map & DB link logic XML-file
(settings.xml)
Map files (.shp & .mdf)

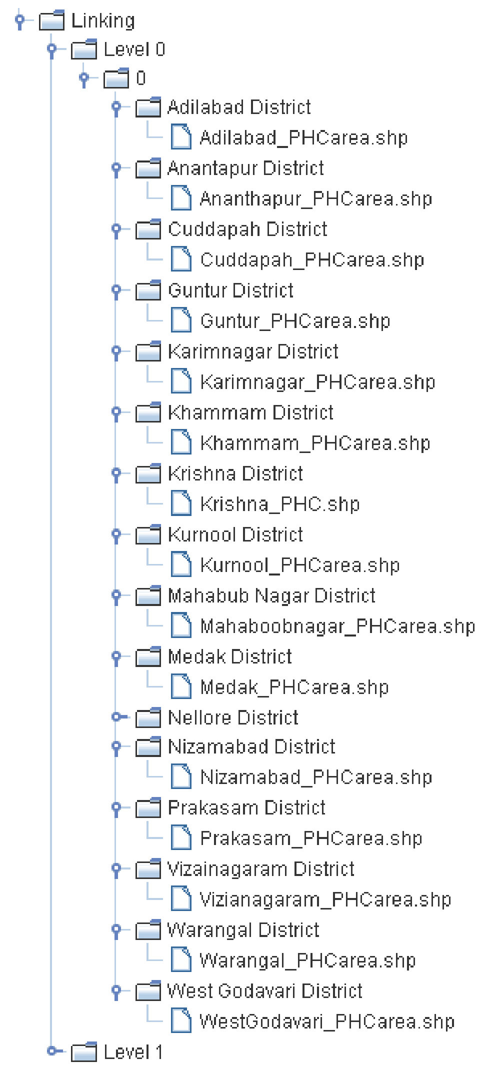

Distribution
HISP GIS JAR
LIB JARs
Resource JAR
LIBs
gt2-main.jar
gt2-shapefile.jar
jai_core.jar
jCharts-0.7.5.jar
JTS-1.4.jar
looks-1.3-snapshot.jar
mysql-connector-java-3.0.14-production-bin.jar
opengis.jar
opengis-legacy-0.1.jar
opengis-tiger.jar
RmiJdbc.jar
units.jar
vecmath.jar

Appendix B

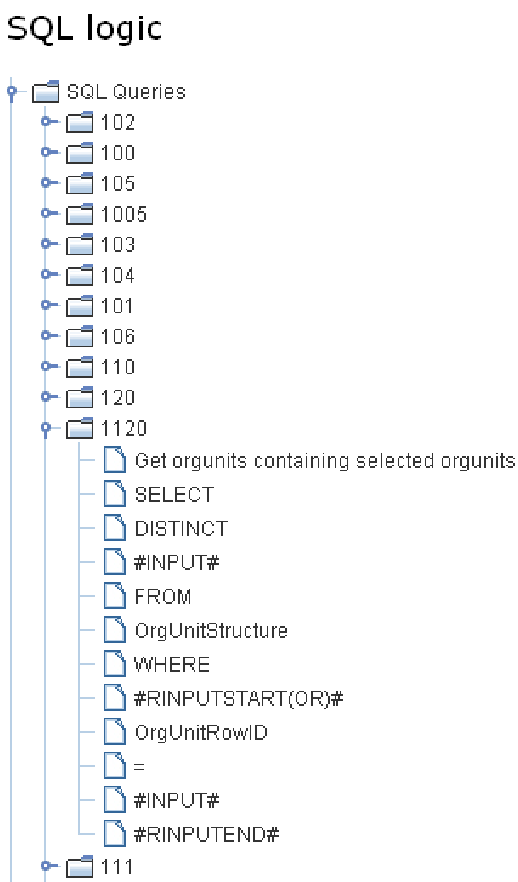
Linking maps with DB

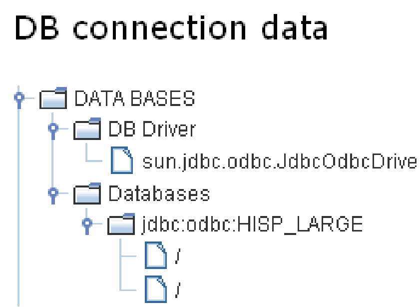


Linking maps with maps



Appendix C


SQL logic
SQL Queries
102
100
105
1005
103
104
101
106
110
120
1120
Get orgunits containing selected orgunits
SELECT
DISTINCT
#INPUT#
FROM
OrgUnitStructure
WHERE
#RINPUTSTART(OR)#
OrgUnitRowID
=
#INPUT#
#RINPUTEND#
111


DB connection data
DATA BASES
DB Driver
sun.jdbc.odbc.JdbcOdbcDriver
Databases
jdbc:odbc:HISP_LARGE
/
/