UiO : Department of Informatics
University of Oslo

Enabling web based mapping for public health in resource constrained environments: Bootstrapping implementations by leveraging platform generativity

Sushil Chitrakar

Master's Thesis Autumn 2015

Enabling web based mapping for public health in resource constrained environments: Bootstrapping implementations by leveraging platform generativity

Sushil Chitrakar

3rd August 2015

Abstract

Due to digitization of data and its access through Internet, the web based mapping applications are also increasing rapidly. Moreover, the thematic mapping adds more value to the mapping applications, which helps to express natural and socio-economic phenomena in the map. This could be further useful to provide support in planning, analysis, and decision-making of any organization. Some of the examples of mapping applications include ArcGIS, and Google Map. Web based mapping is also important in public health sector and some of the related applications are namely accessing health facility placement, early warning of emergency diseases, and respond to health emergency. Although, web mapping is a useful tool, it is time consuming to add and access data through Web in resource constrained environments. Moreover, resource constrained environments may lack experts and the people over there may not be able to use the system if it is technical. Taking a case of DHIS2, a health information management system, there are series of technical steps that has to be followed in order to configure the system for its mapping features. These processes can be very difficult to be followed in resource constrained environments. Considering the easiness for users to import mapping data into the system in resource constrained environments, an app (alpha and beta versions) has been developed by following the ADR (Action Design Research) methodology, and the concept of bootstrappig and platform generativity to make the platform more generative. The alpha version of the app integrates all the required tools into one solution and beta version improves the user interface, and functionality to provide more features to the users while importing coordinates into the system. In this thesis, the potential to integrate all the required steps to configure web thematic mapping has been identified by analyzing the DHIS2 platform. Moreover, a method to leverage platform itself to make it more generative has been identified by developing an app for the import functionality of a mapping application.

Preface

I express my sincere gratitude to my supervisor, Dr. Knut Staring for the continuous support and guidance throughout my master thesis with his valuable suggestions. With his deep knowledge about the subject matter helped me to move easily and complete my thesis. Without his guidance completing this thesis alone for me would have been very difficult.

My sincere thanks also goes to Bob Jolliffe, Mark Polak, Lars Helge Øverland and entire DHIS2 team who helped me to understand different aspect of developing an app using DHIS2 Web API platform and gave their valuable suggestions and helped me to test the app to make it better.

Thanks to my friend Adrian Manuel Arevalo Soria who helped me to understand the ADR concept and its use.

My final thought goes to my lovely and supportive wife Ambika Shrestha Chitrakar and my family.

Thanks all for your continues Support.

Sushil Chitrakar, 2015/08/03

Table of Contents

List of Figures

List of Tables

Abbreviations

ADR - Action Design Research
ANC - African National Congress
API - Application Programming Interface
app/App - Application
BCG - Bacille Calmette Guerin
BIE - Building, Intervention, and evaluation
CRS - Coordinate reference system
DHIS2 - District Health Information System 2
EPSG - European Petroleum Survey Group
ESRI - Environmental Systems Research Institute
GADM - GADM database of Global Administrative Areas
GIS - Geographic Information System
GIS - Geographic Information System
GML - Geography Markup language
GUI - Graphical User Interface
HIMS - Health Information Management Software
HIS - Health Information System
HISP - Health Information System Program
kbps/mbps - Kilobits Per Second / Megabits Per Second
OGC - Open Geospatial Consortium
OU - Organization Units
PROM - Programmable Read-Only Memory
QGIS - Quantum Geographic Information System (software)
REST - Representational State Transfer
UTM - Universal Transverse Mercator Coordinate System
WHO - World Health Organization

Chapter 1

Introduction

1.1 Topic

Due to digitization of data and its access through Internet provides tremendous opportunities to share knowledge among people all around the world. This is also valid for map related information. Many GIS (Geographic Information System) technology based applications such as ArcGIS [17] are already available in market to serve geography related solutions. Such applications help to visualize geographical locations into the map. Moreover, with the emergence of Web1.0 and Web2.0 technologies [28], it is possible to access map related applications through Internet and share open api between applications. Google Map is one of the examples of Web based GIS applications. Many mapping applications use thematic maps, which adds more value to the applications. It helps to express natural and socio-economic phenomena in the map [21], which can be very useful to provide support in planning, analysis, and decision-making level of any organization.

Public health sector is one of the application area where maps can be used to visualize the health related information based on geographical information. Winslow [76] defines public health as "the science and the art of preventing disease, prolonging life, and promoting physical health and efficiency through organized community efforts for the sanitation of the environment, the control of community infections, the education of the individual in principles of personal hygiene, the organization of medical and nursing service for the early diagnosis and preventive treatment of disease, and the development of the social machinery which will ensure to every individual in the community a standard of living adequate for the maintenance of health ". Some of the mapping applications within the public health includes accessing health facility placement [52], early warning system of communicable diseases [64] etc.

GIS is a useful tool for public health sector but it is difficult to be deployed in resource constrained environments. Mostly, least developed and developing countries suffer from resource constrained situations such as poor economy, lack of proper manpower, lack of data, poor Internet facility, no Internet in all the places of the country etc. Please refer to

the section 2.1.2 for more detail about resource constraints in developing countries to use GIS tools. There are different open source tools that can be used for mapping functionality that removes the cost of purchasing GIS tools, but the geographical data, which is the main element for mapping, are huge in volume. Importing all the data and accessing map through Web is a time consuming process and the time required to load the mapping solution largely depends on the amount of geographical data available to visualize the map. One way to speed-up the mapping feature in any application could be by reducing the number of geographical data. If we remove some data from the original data, we also remove certain level of information. The data should be minimized in a way that when map is drawn it should preserve at least the boundary of the geographical location.

One can think of adding a feature to import the simplified map into the health information management systems in case of resource constrained environment. We can either add this feature within the stand-alone software or develop an app for generative software platforms. The trend of stand-alone software is moving towards software platforms with generativity at its core. It is because, the stand-alone software cannot evolve highly even when it has modular structure [78]. Su et al. [29] defines modularity design as a special structure where all the parameters and the functionality within one module are interdependent and those ones are independent with other modules. According to Huynh and Cai [49], modularity in design is very important in software systems and it helps to determine the software quality in terms of evolveability, changeability, maintainability etc.. Modular software design is evolveable but adding or changing modules require good knowledge about the system, which further restricts external users to add modules. Platform generativity on other hand, allow platforms to provide functionality to users that the designers of the platform did not have in mind and helps them to evolve[51]. Kretzer and Maedche [51] explains that the generativity refers to the ability of the platform to be extended, and the ability of the platform to be integrated into another system in order to create new output. Google, Facebook, and Apple are some of the powerful examples of companies that have deliberately created platform to produce products and services that were not imagined by them at the beginning. Map services from the Google's Web API is one of such examples [78].

We can also see DHIS2 [13], a health information management system as an emerging platform [25]. DHIS2 selected the Java-based technology for its development, which was first deployed in a clinic of kerala, India in 2006 [73]. DHIS2 is basically an open source tool for collection, validation, analysis, and presentation of aggregate statistical data especially for health information management activities. As of October 2012, DHIS2 software was used in more than 47 countries in Africa, Asia, and Latin America. Some of the countries that have adopted DHIS2 as health information system (HIS) in their nation-wide level include Kenya, Tanzania, Uganda, Rwanda, Ghana, Liberia, and Bangladesh [13]. DHIS2 works as a software platform and provides Web API to interact with external applications [25]. This feature helps DHIS2 to add more features even from the external users.

During implementation, it is important to follow some strategy that can guide to get started and reach the critical mass. Bootstrapping is one of such strategies for the implementation of any problem. It suggests to start the implementation with a simple solution, add more complexities to the started task and iterate the process until the desired output is achieved [68].

This master thesis aims to explore a good solution to enable the web based mapping functionality in resource constrained environment. The implementation will use the bootstrapping strategy by leveraging the platform generativity.

1.2 Keywords

Thematic Mapping, GIS, Generativity, Bootstrapping, ADR, DHIS2, Platform, Web API

1.3 Problem description

We know that any web-based application helps people to connect quickly irrespective of their physical locations. In case of health sector, it is possible to access the health related content and update the system from any location, if the Internet facility is available. Despite various advantages of using web based mapping application, it can be challenging to use it in resource constrained environments.

Since in resource constrained environment, there can be lack of high-speed Internet facility, skilled staff, good economy etc., these constraints further restrict the web based applications to be bit selective towards the choice of tools and functionality. Below are some of the challenges that can come across while implementing web based mapping functionality for public health in resource constrained environments:

- *Data sources:*
  Geographic coordinates are the key elements to display different regions in the map. Different country have different organizational hierarchy and they can be displayed in the map if we have their geographical coordinates. Organization hierarchy with their coordinates are data for any mapping tool. Figure 1.1 shows an example of organizational hierarchy, which is a standard hierarchy of census geographic entities. The hierarchy can be followed with the help of straight lines (Nation - Regions - Divisions - States - Counties - Census Tracts - Block Groups - Census Blocks). The organizational hierarchy of all the countries may not be the same. For example, some countries do not have states and some countries have village development committees (has similar level to the municipality in a country) in the hierarchy.

  Finding a proper data source for maps is a challenging task. GADM [38] and government are some examples of data sources for the maps. GADM [38] is one of the spatial databases that provide spatial data of

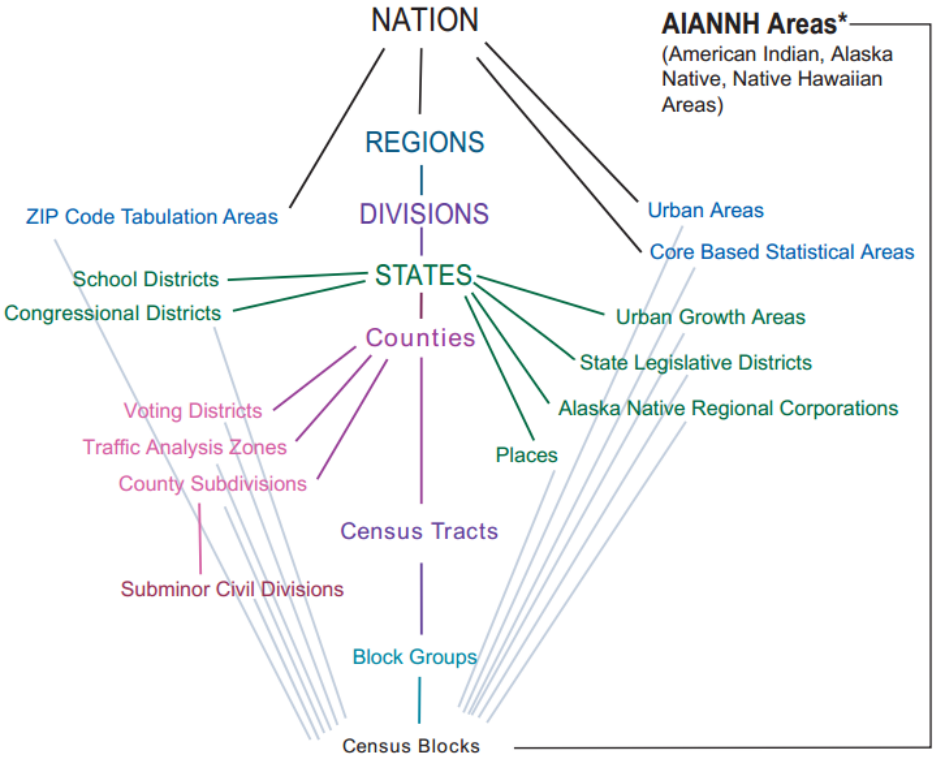


Figure 1.1: Standard hierarchy of Census Geographic Entities [70]

the location of the world's administrative boundaries that can be used in GIS or GIS like softwares. GADM refers to organization hierarchy of a country such as provinces, departments, and bibhag. It provides spatial data for organization hierarchy of a country in shapefile, ESRI [32] geodatabase, RData etc. Some of the government websites that provide data for maps are http://earthexplorer.usgs.gov/ for United States, and http://drm.moha.gov.np/ for Nepal. They also provide data in different file formats such as shapefile, GeoJSON, Excel, etc.

- *Automatize the manual processes:*
  There can be several processes involved while storing proper geographical data into the system. Some of the steps may include conversion of the coordinate projection, simplification of the coordinates, storing data into the system, and displaying the map based on these data. Coordinates for organizational hierarchy of countries that we get from the public data sources may not have the projection that one require. In such case, the projection of coordinates should be changed. Simplification of coordinates to reduce the size of data can be another step. After simplifying the coordinates of the organization hierarchy, one might need to store them properly in a health management system. Another step could be the display of map itself based on these simplified coordinates. Subsection 2.3.3 in chapter 2 provides a quick picture of how complex the enabling thematic map-

ping process could be.

People may not have expensive commercial GIS system to perform all these tasks in resource constrained environment. They might do most of the tasks manually and doing manual work for large data might include human errors. Moreover, these processes are not easy to be done by an unskilled staff; it requires series of steps to be performed to get the final simplified data that has to be stored properly into the system. It would be better if these manual processes could be automatized so that less human interaction would be required. This could avoid the chances of human error and at the same time could save the time of staffs.

- *Selection of open-source tools:*
  It is better to automatize the manual processes but it is necessary to find out the open-source tools that are available for these purposes. In addition, it is also necessary to test if these identified open-source tools can be integrated together in a common platform or not.

1.4 Motivation

Thematic mapping is useful, but hard to set up in resource constrained environment. The emergence of generic web platforms is a powerful trend Within the field of public health. DHIS2 is one of the emerging platforms in 47 countries [7]. There is a lot of literature on thematic mapping, also for health. There is also a growing literature on platforms, and quite a lot of research around DHIS2, even on bootstrapping [44]. However, there is little literature on leveraging platforms for thematic mapping in public health (but some on Google Maps etc.), and very little on bootstrapping platforms for public health.

DHIS2 is an open source generative software platform for managing the health related information. It provides web based mapping features and allow external apps to interact with it through Web API [13], which can be exploited to store the data related to the organization hierarchy and their coordinates into the system. This further helps in enabling the web based thematic mapping feature in DHIS2. Therefore, we want to explore leveraging the DHIS2 platform itself to get started with thematic mapping on the DHIS2 web platform, in other words, bootstrapping it.

It is important to empower the users with less technical knowledge for enabling the web based mapping in resource constrained environment. Therefore, we decided to achieve this goal by developing an app that can help to simplify the coordinates and can use the Web API provided by the DHIS2 to import the simplified data into the system. The app should be designed in a way that it can be easily adopted to any other web based system for simplifying the coordinates and then importing them into the system with only few specific changes. This way, we can explore bootstrapping of platform for thematic mapping.

1.5 Research questions

The main research objective of this master thesis is to explore the bootstrapping of a generic platform for thematic mapping, with a specific focus on the field of public health and resource constrained settings. This leads to the following research questions:

1. What can be the potential for integrating all necessary stages for configuring web thematic mapping?
   This research question ask to identify all the necessary stages for configuring web thematic mapping and then explore the potential to integrate them. We plan to solve this problem by considering DHIS2 platform.
2. How can we leverage platform itself to make it more generative?
   In order to answer this research question, we plan to develop an app to integrate all the necessary stages for configuring the web thematic mapping. This will be achieved by bootstrapping the DHIS2 platform to make it more generative.

1.6 Contributions

There is little literature regarding how to use bootstrapping concept to leverage the generic platform for thematic mapping. This project aims to contribute some work to the literature related to this area.

1.7 Thesis outline

This section provides an overview of contents presented in each chapter of this thesis. These chapters start with the background and related work, which is followed by the methodology, design and implementation, discussion, conclusion and future work chapters serially. Brief description about each chapters are provided below:

- Chapter 2 presents background and related work about GIS, the concept of bootstrapping and generativity, and DHIS2. First, it describes GIS and its importance in public health. It also provides brief information about the importance of GIS in developing countries. Secondly, it provides literature reviews about the concept of bootstrapping, generativity, and platform. Lastly, it provides background about the DHIS2 as a case for this thesis, evolution of thematic mapping in DHIS2, explains DHIS2 as a platform and explains all the stages that are required for its import functionality to configure the DHIS2 for mapping.

- Chapter 3 provides overall methodology of the thesis. First, it provides background information about ADR (Action Design Research) and explains why ADR has been chosen for the methodology

of this thesis. This chapter ends by providing information about how different data collection methods are used to collect data and requirements of this master project.

- Chapter 4 provides detail information about the design and implementation of alpha and beta version of the app that are developed as a solution to the research questions of this master thesis. The first section briefly explains different tools that were chosen to be used in the app, the second section explains the design implementation of alpha version, and the third section explains the design implementation of beta version. The last section provides detail about the used scripts, tools during the experiment.

- Chapter 5 provides analysis and discussion of research questions on the basis of feedbacks received from the users of DHIS2

- Chapter 6 concludes this master thesis by providing summary of the findings of this research work.

- Chapter 7 provides the list of works that can be performed in future in order to improve this thesis work.

Chapter 2

Background and Related work

2.1 GIS and public health

This section provides detail about geographic information system, thematic maps and explains their importance in public health sector. It also presents the difficulties in deploying maps in resource constrained environments and ends by providing information about DHIS2 regarding thematic mapping process as a case for this master thesis.

2.1.1 Geographic information system (GIS)

Some view Geographic Information System (GIS) as a tool for spatial research and policy analysis, while others believe it as a part of a larger emerging new science including geography, cartography, geodesy, and remote sensing [63]. GIS in a formal way can be defined as a computer based system that provides following four sets of capabilities to handle geo-referenced data: *i) data capture and preparation, ii) data management, including storage and maintenance, iii) data manipulation and analysis, iv) data presentation* [48]. It implies that the users of GIS tool can enter geo-referenced data, store and maintain these data, modify them as per their requirement for analysis, and get various options to visualize them.

GIS is widely used all over the world for different purposes in both government and private sectors. Some of them include identifying the high-risk areas of annual monsoon-related flooding by investigating rainfall patterns and terrain characteristics, health planning and monitoring the health situation in a country, identifying the health affected by spatial factors such as demographic, life style, environment etc.

There are lot of open source and commercial GIS engines in market. Some of them are ArcGIS [1], GeoTools[2], and SharpMap [3].

2.1.2 GIS in developing countries

GIS is considered to be one of the critical tool in developed countries for resource management, regional planning, and economic development [56]. The use of GIS is not only valid for developed countries but also for the developing countries. There are many potential GIS applications in

developing countries such as public health studies, irrigation and drainage, disaster avoidance, management of conservation areas and parks, etc. [80] provides five main constituents of GIS as people, hardware, software, methods, and data as in figure 2.1.

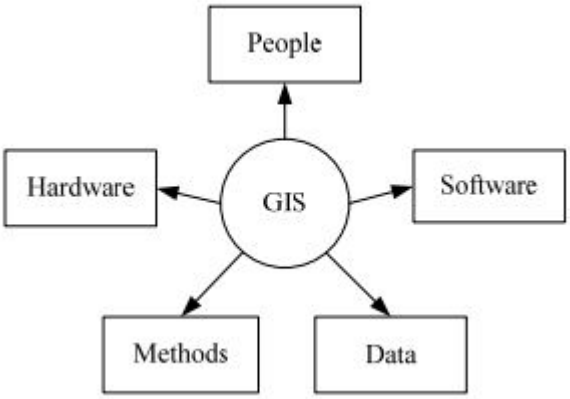


Figure 2.1: The main five constituents of GIS [80]

The requirements for GIS in both developing and developed countries are the same but developing countries suffer from some constraints because of which it is very difficult to use applications based on GIS. Some of these constraints summarized from [56, 67] are provided below:

- *Cost constraints*: GIS application require purchasing of proper software and hardware infrastructures. Moreover, it is necessary to hire some professionals to operate and manage the GIS application. Developing countries have limited financial resources, because of which it becomes expensive for them to have all the latest functionality of GIS in their applications.

- *Infrastructure constraints*: Infrastructures such as electricity and good Internet bandwidth that are required to use GIS applications may not be available in all parts of developing countries. Many places in a country may have limited access to such infrastructure. Load shedding and availability of few kbps/mbps Internet bandwidth to individuals are some of the existing problems. Moreover, due to geographical structure and opportunities, people prefer to live in big cities and facilities from government are more focused on main cities of the developing countries. This results in problem to provide commercial support services in remote places.

- *Educational constraints*: Developing countries also suffer from poor literacy rate. Computer education is becoming popular but people who have mastered computer skills are still few in numbers. Most of the organizations that use GIS applications select local staff and send them abroad to study GIS courses. Importing skilled workers from abroad is very expensive for developing countries but there

are lot of seminars or trainings related to GIS, which are available through Internet. Attending online training instead of importing skilled workers or sending local staff abroad to learn about GIS could be beneficial for the developing countries in terms of cost.

- *Data constraints*: Valid data is the core of GIS systems to make maps. The GIS applications cannot provide proper result if the data are not complete or faked. Moreover, making maps and updating them is an expensive and time-consuming operation. Therefore, it is better to inserted all the valid data at one time. It is useful more in developing countries. However, all data may not be available at once or may be made hidden, it might also be difficult to collect data which are related to socioeconomic phenomena such as population density, growth etc.

- *Political stability*: Political instability is one of the problem in most of the developing countries and it can affect GIS related projects from not accomplishing its full installment.

2.1.3 GIS and health care

Health care is one of the applied field of GIS and there is tremendous potential for GIS in health care organizations, although it has been recently realized. It is anticipated that GIS technologies have lot to offer in helping improve health related services, management, and research [79]. Some of the applications of GIS in health care could be analyzing the spatial distribution and trends in space and time of disease, the potential distribution of high-risk groups, risk factors stratified analysis, resource allocation assessment, disease surveillance planning, disease surveillance alarm, and et. [63]. [80] presents some of the application areas of health care using the functionality of GIS. They are provided below:

- *Disease surveillance and epidemiological studies*: GIS when used in disease surveillance, can display the real-time and dynamic changes of disease development [80]. We can take an example of an outbreak of dengue fever in a village in southern India, reported on 2001. Researchers used GIS for spatial analysis and demonstrated a centrifugal spread of cases from the most affected street until it involved the entire village. Spatial analysis revealed that cases occurred in clusters and that these could not have occurred by chance. The researchers were able to prevent the outbreak from spreading to an adjacent village by controlling the dult mosquitoes and larvae [74].

- *Environmental health research*: GIS can also help in environmental health research by doing space analysis. It can model estimates on situation of environmental pollution, and help to visualize it in the form of map. GIS can also connect the environmental factors to the health conditions, and help to understand the possible health effects posed by the environmental hazards [80].

- *Health services, utilization and decision research*: Health resources should be distributed according to the extent of health demand and priority. GIS integrate locations, coverage of health service and its utilization, population, and other information in order to help health decision-makers to conduct the management of health resources [80].

- *Respond to health emergencies*: GIS can also help in responding to the health emergencies. For example, if a patient is infected by a life threatening transferable disease, GIS can be used to prevent further spread of this epidermic. It can be done by performing statistical analysis on the suspected patient and his/her close contacts, and take decision (such as segregation, disinfection, quarantine, etc.) based on the specific cases [80].

One of the open-source health information system that uses GIS is WHO (World Health Organization). It realized the advances in technologies such as desktop GIS packages, proliferation of computers, connection to World Wide Web, and low cost mapping and remote sensing in 1990s. It also recognized the opportunities for public health such as integration of data from multiple sources, easy visualization of diseases and health information that supports decision-making and advocacy, spatial analysis for disease clustering, risk, accessibility etc. As a result, WHO developed *The HealthMapper* as an entry point to GIS. The healthmapper is an open-source GIS desktop package with ready-made database of core geographic data. It provides a data management and mapping applications designed especially for public health users. The key-features of healthmapper for mapping interface include thematic maps, easy to locate and select feature, measurement and proximity analysis, interactive graphs, my favorite maps, and icon driven layering of geographic data (vector and raster) [60].

2.1.4 Thematic maps

The thematic map is a prominent way of expressing natural and socio-economic phenomena and making specialized maps with the help of map contents [21]. It is different from general reference map; it does not just show lakes, cities, coastlines, and political boundaries but uses this information to understand the specific theme or physical phenomenon such as immunization coverage, population density, life expectancy etc [65].

According to [77], the traditional methods of making Thematic Maps are based on Geographical Information System (GIS) and applied to only one layer: map. Please refer to the section 2.1.1 for more details about GIS. This paper also states that the Thematic Maps can also show the patterns and trends that are not found easily.

In the process of developing cartography, the thematic maps of its information transmission, storage, conversion, and display has great advantages. It has become a part of research, analysis and evaluation,

forecasting, planning, and command and management of important tools and means [21].

Thematic mapping as a service on Internet is widely used in practice. Number of map applications on Internet has been increased since Google launched its Google Map on the Internet in 2005. Moreover, with the concept of Web2.0, sharing of open api, the coupling between data and application services, has strengthen the geographic visualization by adding the ability to present logical analysis. Some of the relevant analysis could be revenue analysis of people in a country, analysis of damage from earthquake, and so forth [50].

Web based thematic mapping is also important in public health to visualize and quickly understand the health related information. Some of the uses of thematic mapping in public health could be displaying the facilities in hospitals, and visualizing the indicators such as child mortality rate, BCG doses given to children under 1 year in a particular organization unit. It also helps in analysis of health related problems and can support in planning and decision making.

2.2 Bootstrapping and platform generativity

We want to bootstrap the DHIS2 platform for thematic mapping to make it more generative. Detail about the concept of bootstrapping and platform generativity is provided in following subsections.

2.2.1 Bootstrapping

According to Hanset and Aanestad [43], bootstrapping is a process of making tool by means of the tool itself. It can also be described as a process, which is capable of loading and starting itself. Starting a Windows system is a good example of a bootstrapping process. When a computer is turned on, hardware reads and executes a piece of program, which is stored in PROM. This software again loads and starts a piece of software at a certain address of the hard disk. This software then loads and runs MS-DOS, which in turn loads and starts a Windows system.

Bootstrapping can also be used as a concept for the implementation strategy. Skorve and Aanestad [68] provides a brief review of some selected literature from the vast area of implementation studies and explains that different implementation strategies have different benefits like cost and risks but do not provide any specific selection criteria for how to approach implementation, where and how to start, which users to begin with etc. They introduce the concept of bootstrapping in order to fulfill these issues. Below is a short explanation of the concept or algorithm of bootstrapping based on [16, 68]:

1. Start by designing the first, simplest, cheapest solution with few limited number of users

2. Use the technology and repeat it as long as possible by enrolling more users

3. If possible, explore, identify and adopt more beneficial ways of using the solution, go to 2

4. Use the solution in tasks that are more critical, go to 2

5. Use the solution in tasks that are more complex, go to 2

6. Improve the solution so that new tasks can be supported, go to 2

This algorithm suggests starting any implementation with a simple and cheapest solution as possible with limited number of required users. After successfully completing this task, more complexities and cases can be added to the started task. It is an iterative process of improving the task until the requirements of implementation are achieved.

2.2.2 Generativity

Tilson et al. [71] defines generativity as "the ability of a self-contained system to create, generate, or produce new content, structure, or behavior without additional help or input from the original creators". By referring to this definition, we can say that any technology that includes generativity strategy in its core will have capability to generate new contents without the intervention of the original creator of the technology. We can take an example of a software system that provides Web API (application programming interface) to interact with external apps. Any individual developer from outside of the system can develop such apps by following the standards provided by the Web API in order to interact with the system. The functionality of the app is not implemented within the system but with the help of Web API, the system is capable of generating new contents and share its content to external programs.

According to Zittrain [81], capacity for leverage, adaptability, ease of mastery, and accessibility are four crieteria of generativity functionality of any technology. These crieteria ensure that the generativity increases with the ability of users to generate new, valuable uses that are easy to use and are in turn sources of further innovation. The four properties of the generativity functionality by Zittrain [81] are shortly explained below:

- *Capacity for leverage*: It refers to the capacity of the technology to the extent it enables the valuable accomplishment across a range of tasks.

- *Adaptability*: It refers to both the use of technology without any change and the readiness of the technology to get modified with an objective to broaden its range of uses.

- *Ease of mastery*: This property of the generativity of technology reflects the easiness for a broad range of users to adopt the technology regardless of whether the technology was designed by considering

the tasks that user want to perform or not. Here, easiness to adopt the technology refers to the skill required by the users to make use of the technology.

- *Accessibility*: This property is directly related to the ease of mastery. More the technology is easier to be adopted, the more the accessibility of the technology.

According to Elaluf-Calderwood et al. [31], generativity is one of the important strategies in business models of software platform. We will talk more about platform in section 2.2.3.

2.2.3 Platform

According to Hanseth and Lyytinen [44], platform has heterogeneous and growing user base and their design context is not fixed due to constant generification of included IT capabilities. Some of the examples of platforms includes office software platforms (MS Office, officestar), operating system platforms (Unix, Linux), application development platforms (e.g. Service Oriented Architecture). Many platforms initially have limited IT capabilities and later obtain emergent features due to their open characteristics to the user community.

According to Tiwana et al. [72], platform is an element of platform-based software ecosystems. Firefox browser that has thousands of *add-on extensions*, Apple's iPhone operating system (iOS) that has millions of *apps* are few examples of platform-based software ecosystems. Figure 2.2 shows all the elements of platform-based software ecosystems.

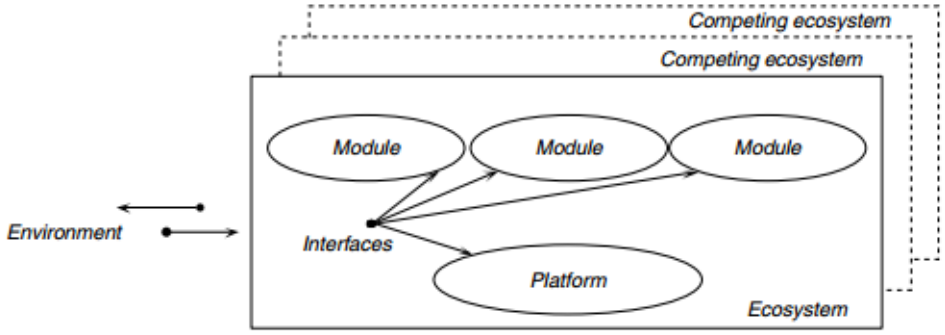


Figure 2.2: Elements of software-centric platform ecosystem from Tiwana et al. [72]

The figure 2.2 shows that the platform's ecosystem (also known as digital ecosystem [31]) is a collection of platform and modules specific to it and it creates a competitive barriers for rival ecosystems. The software-based platform is the core of the digital ecosystem and it provides core functionality to the modules that interoperate with it through the interfaces. The modules are the add-on software programs that connect and add functionality to the platform and the interfaces provide specifications

and design rules to exchange information between the platform and the modules [72].

DHIS2 can be perceived as a generative platform. Similar to the characteristics of a platform, DHIS2 also has heterogeneous and growing user base and its design context is generic. Its design evolve due to the iterative modification on local and global designs requirements. Moreover, DHIS2 can be seen as a software-based ecosystem, where DHIS2 performs as a platform and allow multiple modules or apps to connect and add functionality into the system through Web API. Here, Web API complies with the rules of REST architectural style [25] and plays the role of interface between different modules and DHIS2 platform.

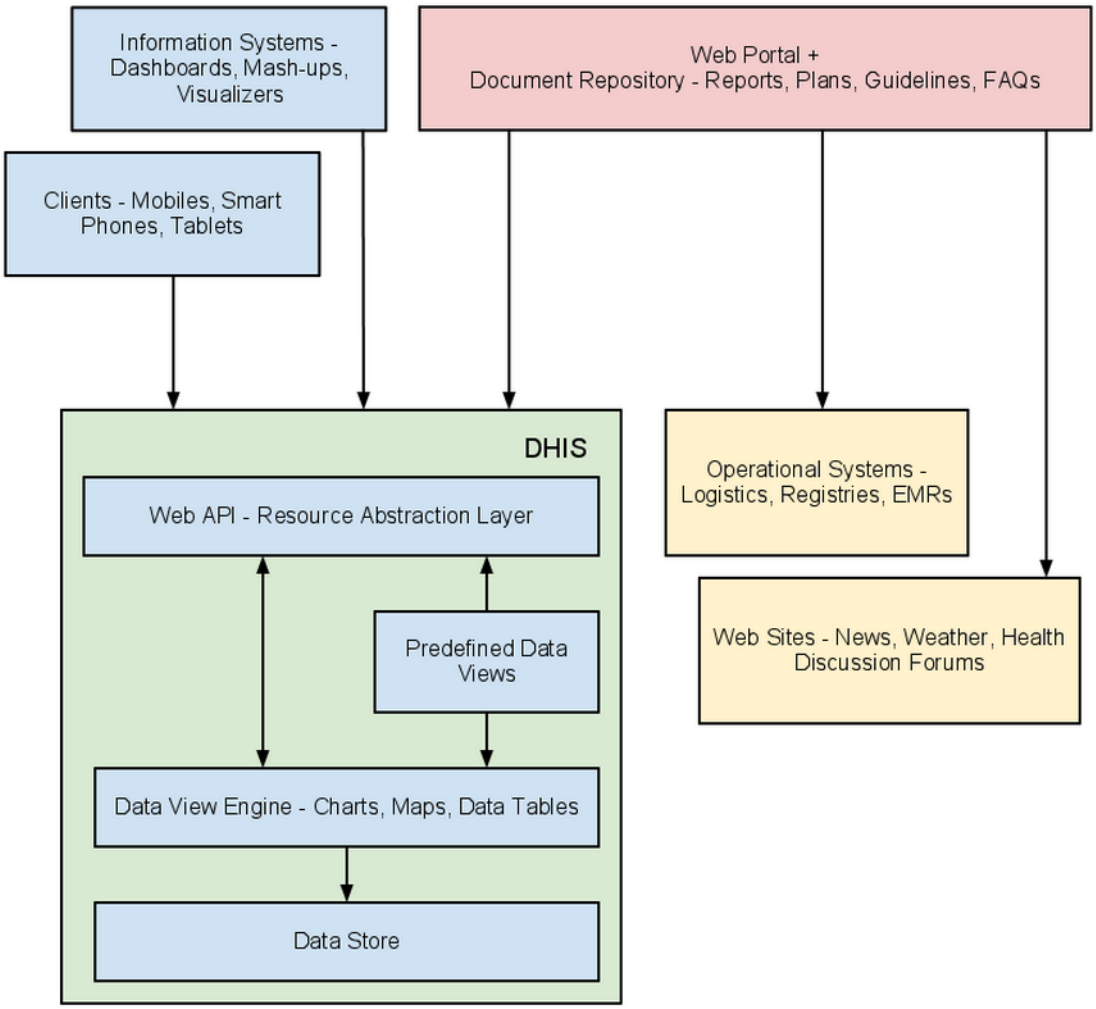


Figure 2.3: DHIS2 framework [25]

Figure 2.3 shows a framework of DHIS2. It clearly shows that DHIS has core functionality to store data and display charts, maps, and data tables. Other modules of the DHIS2 such as Web portal, clients, and information systems through Web API using URL can access these core functionalities. The DHIS2 functionality can be extended by allowing apps to interoperate

with DHIS, independent of the implementation of DHIS with the help of Web API. It fulfills all four criteria of generivity functionality. It has capacity to leverage in a range of different tasks. For example, DHIS2 can serve as a management system for domains such as logistics, labs, and finance [25]. It is also easily adoptable since users does not need to deal with the implementation part of DHIS2. Even with fewer skills, apps can easily interoperate to DHIS2 with the help of Web API. Since DHIS2 is an open source and easy to use, it is accessible to different user communities. These properties of DHIS2 ensure that the generativity of DHIS2 increases with the ability of users to generate new, valuable uses that are easy to use and are in turn sources of further innovation.

2.3 DHIS and thematic mapping

2.3.1 Evolution of thematic mapping in DHIS

"DHIS emphasises the use of information for action, improved health services, user participation, and 'live' (in real contexts), agile and rapid prototyping. The DHIS software development effort was organized within the HISP network, and has since its inception been embedded in a synergetic mixture of public health and participatory design perspectives."[73]

[42] provides a short history about DHIS, which are provided below:

- In the year 1994, a Health Information System Program (HISP) was started in South Africa. Initially HISP was based in two Cape Town universities and funded by Norwegian Agency for Development Corporation (NORAD). Taking part in the reconstruction and development program launched by the ANC to reconstruct the health service in South Africa.
- In 1997, open source DHIS was developed. After the success of pilot phase, the strategies, process and software developed in the pilot areas were adopted by the Department of Health in 1999 as National Standard.
- In 2002, the design team of DHIS came up with the five design parameters for the development. One of the design parameters was to make this system free (Open source). Earlier version of DHIS was developed based on commercial software making users to pay for the commercial tools used by the DHIS.

In the year 2005, [42] developed a module for GIS interface in DHIS that allowed the creation of health thematic map. The application was based on java libraries. They build an interface to display maps, which looks like in figure 2.4.

Because of strong using of Microsoft windows operating system in the world, the initial version of DHIS was developed for Microsoft windows, using the Access as database. With the growing users, new demand and countries, it became hard to to address those requirement with the

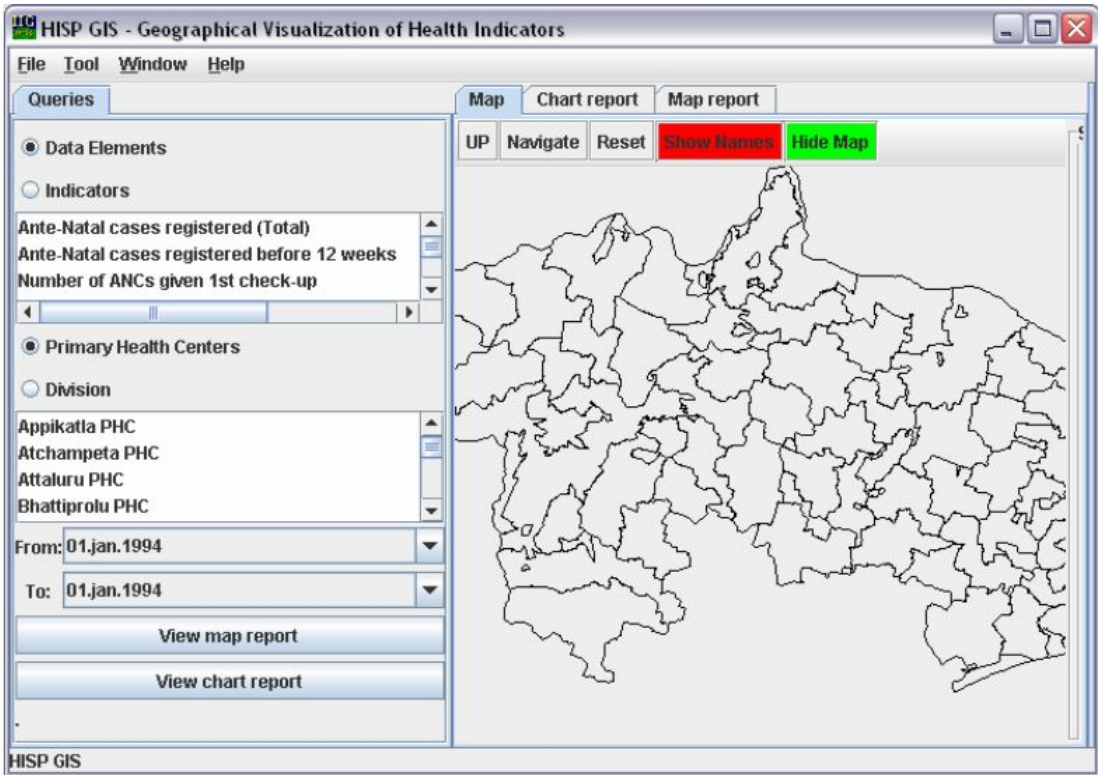


Figure 2.4: GIS interface example from [42]

current DHIS software. The implementation of DHIS 1.3 in Cuba in 2003 showed the limitation of this DHIS at that time. Though, DHIS generally being regarded as the open source and free, the dependency of DHIS with commercial Microsoft technologies required some expenditure to acquire these commercial technologies. These expenditures were unacceptable for the developing countries and resource constraint environments.

These issues prompted a requirement of developing DHIS, entirely using the free and open source framework. Therefore, in the spring of 2004, a decision was taken to start developing a new version of DHIS named DHIS2, in university of Oslo. The system would be platform independent and would run in any operating system and on most database management systems [75].

With the development of new DHIS2, the importance of integrated GIS module was becoming more prominent and DHIS tried to come up with several solutions for this purpose, but none of the module truly matched the requirement. The current module of GIS used in the DHIS2 is the result of research and development done by [75] as his master thesis in which, he had the requirement of developing a web based, open source and integrated GIS module for DHIS2. One of the interfaces from his thesis is provided in figure 2.5.

In current DHIS2, thematic mapping has been implemented to visualize the classified health information related data in four different vector layers [27]: i) Facility layer, ii) Boundary layer, iii) Thematic layer, and iv)

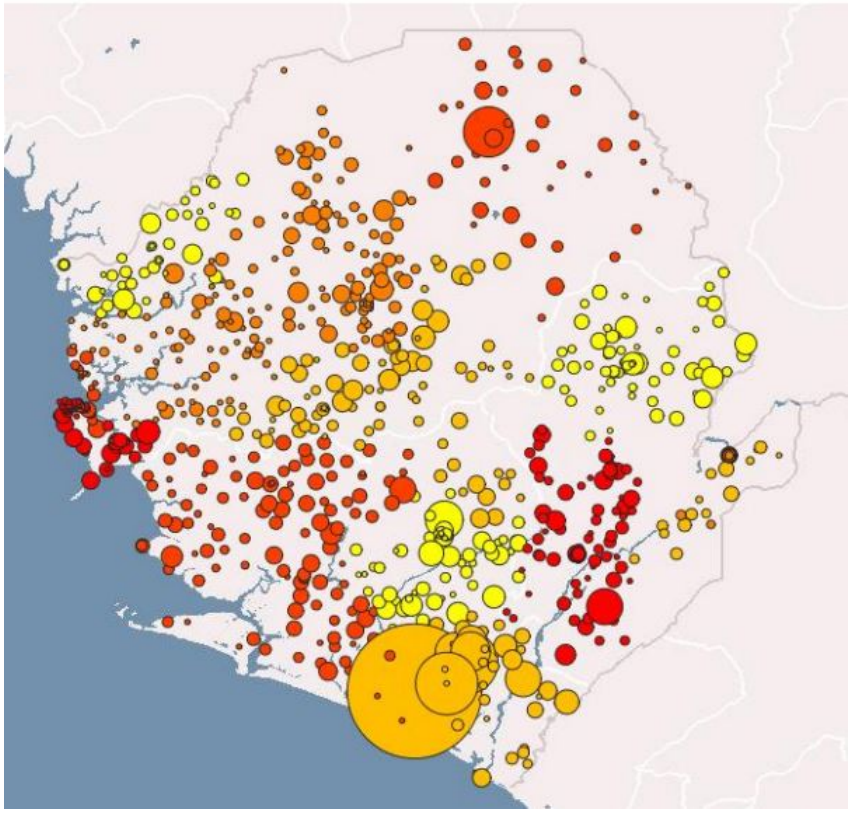


Figure 2.5: GIS interface example from [75]

Event layer. Facility layer displays icons indicating facilities based on the facility type (hospital, clinics, etc.). Boundary layer displays the boundaries/coordinates in the system based on the selected organizational unit. Thematic layer allows users to visualize data based on the combination of indicators, period and the map along with the selection of a particular organizational unit level for the boundaries. Event layer enables users to drill down from the aggregated data displayed in the thematic layers to the underlying individual events or cases.

2.3.2 DHIS as generic platform

DHIS is a health information management system, which has been developed by the researchers of the University of Oslo, Norway. This system has been designed in such a way that it can be launched both locally and globally as a solution to the health information management. This system has been already installed either completely, partially, or in pilot stage in 47 countries including Bangladesh, India, Liberia, Sierra Leone etc. [7], and it aims to launch in several countries also. Systems that aim as a global solution should follow method design such as "generification". Generification is a key design method to spread the local innovations across a heterogeneous network [57].

In case of DHIS2, two processes: i) in-country design of the HIS with an

emphasis on the system rather than on the software, and ii) across-country design of a globally distributed toolbox of software and best practices for in-country HIS design, continuously feed into each other.As shown in figure 2.6, globally distributed solution grow out of local design and uses, and the global toolbox is utilized in the local design processes adapting the global standards [69].

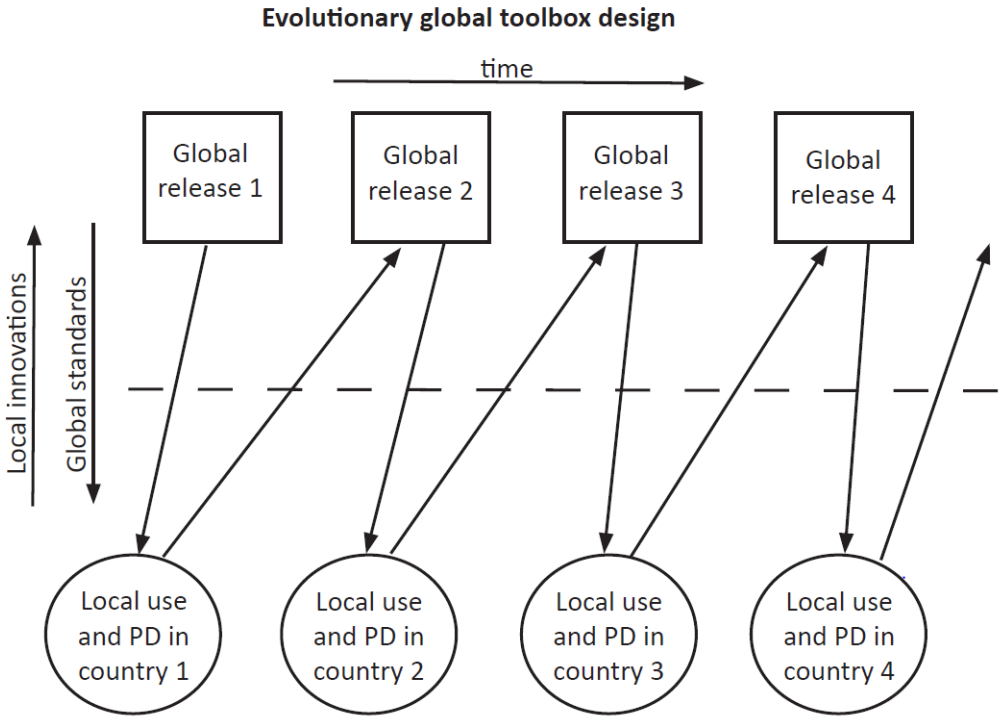


Figure 2.6: Evolutionary global toolbox design [69]

Due to generic nature of the DHIS2 system, it is necessary that any App within the DHIS2 system should also be generic. In DHIS2, importing the organizational units and their coordinates in order to enable the thematic mapping is a manual work. The most commonly used files for importing the organization units are shapefiles. These shapefiles may have different structures to store the organization units depending on the shapefiles providers of different countries. The objective of this master thesis is to come up with a generic solution to import the organization units, considering the variability of the structure of the shapefiles. The app that will be implemented as a solution for importing the organization units into the health information management system should follow the generification method design and should be suitable for both in-country and across-country processes with minimum number of specific changes by the required party.

2.3.3 Existing import process in DHIS2

Existing processes to import data and coordinates into the DHIS2 system can be divided into four sub processes: i) *simplification*, ii) *projection transformation*, iii) *GML file generation*, and iv) *Import*. These processes are handled separately, one after another. In this section, we will discuss in detail about these processes.

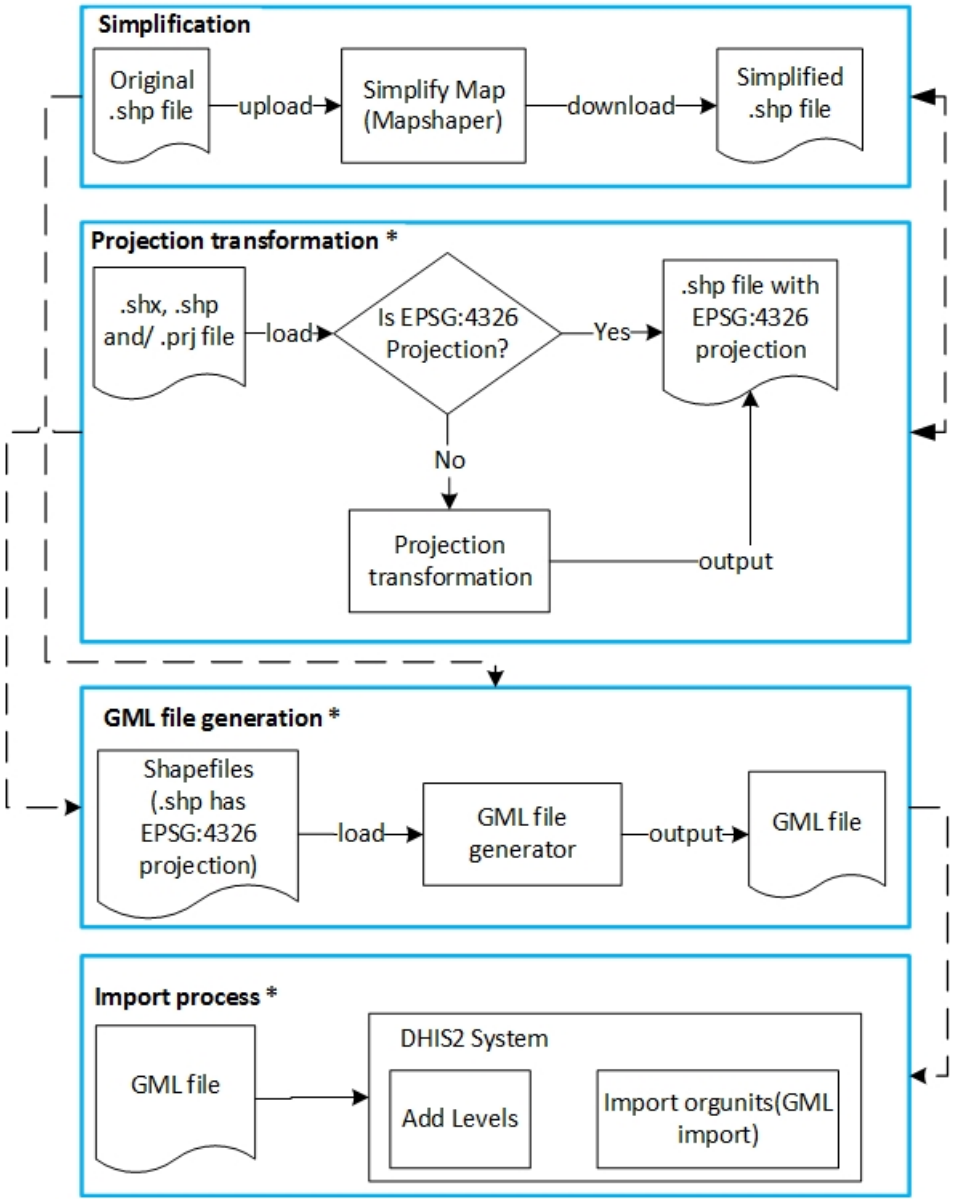


Figure 2.7: Current processes to import data and coordinates in DHIS2

Figure 2.7 shows a flow diagram for these processes. Here, the process of simplification of coordinates is optional. Simplification is required if user wants to reduce the number of coordinates for organization units. It is helpful especially in the case of constrained environment but at the

same time, this process also removes some detail information related to the coordinates of the organization units. Therefore, this process is optional. Other three processes are mandatory and are done one after another. Dashed arrows resemble the linkage between different processes, where one-way arrow represents the specific flow order and bidirectional arrow represents the possibility of any order to perform the processes. We can see that, a bidirectional arrow connects the simplification and projection transformation processes. It means that if an end user choose both the processes, he/she can do any of them at first independently. The one-directional dashed arrows on other hand restrict the flow of processes in order. For example, GML file generation process cannot be done before projection transformation process.

This section is sub-divided into four sub-sections based on the four sub-processes of existing import functionality of DHIS and details about them can be found in their specific sub-sections.

Simplification

Shapefiles 4.1.1 provide coordinates for organization units that can be imported into DHIS2 system to enable thematic mapping functionality. Since the targeted market of DHIS2 is mostly in developing countries, and thematic mapping is a heavy functionality in any application, it requires lot of resources to handle such functionality. As mentioned earlier, this process is optional but it is very important in resource constrained environments. Please refer to the section 2.1.2 for different possible resource constraints in developing countries, while using GIS applications. Through simplification process, it is possible to reduce the size of data which can further speed-up to display map in the application.

DHIS2 has proposed Mapshaper 4.1.4 tool to simplify the coordinates [14] which are stored in *.shp file. This file along with *.dbf file that stores corresponding organization units can be uploaded to the Mapshaper for simplification. End users can achieve the desired level of precision for the map by using slider at the top of this tool. Once the desired output is obtained, user can export output as shapefile, GeoJSON and TopoJSON. For DHIS2, how fast the map can be displayed without trading necessary details from the map is very important. These exported files are then used in other remaining processes.

Projection transformation

Another important part of importing organization units into DHIS2 is converting the coordinate reference system CRS[18] to the coordinate reference system that is supported by DHIS2. This is a mandatory process. This process can be done before simplification or after simplification. It is same, but if end users want to simplify the coordinates, then it is better to simplify the coordinates first and then transform their projection if it is not according to the specification of DHIS2.

There are different coordinate systems that have different projection types. DHIS2 supports only one coordinate projection system that is EPSG:4326 with geographic latitude and longitude [14]. Please check [58] for more detail about geographic coordinates. The projection coordinates in a source file from where we are going to import the data (e.g: shapefiles) might be different than EPSG:4326. Therefore, to make it work with DHIS2, it needs to be converted to the projection system that is supported by DHIS2 which is EPSG:4326. There are many tools available to convert the projection of shapefiles. Also, there are many software api that are possible to be integrated with the app. DHIS2 has recommended "ogr2ogr"tool for geographical format conversion [14]. This tool should be available for most Linux distributions sudo apt-get install gdal-bin and for Windows "FWTools"[4] can be used.

In case of shapefiles, coordinate information are stored in a shape file with ".shp"file extension. It is necessary to check the current projection of the shapefiles before transforming them to EPSG:4326 format. Projection information about coordinates are stored in a separate file called the projection file, which has file extension as ".prj". End users can check the projection of the shapefiles by using fwtools shell using following command [14]:

ogrinfo -al -so filename.shp

After the projections of the shape files are known, the second part is to transform projection from source projection to EPSG:4326, if the current projection is different. This can be achieved by again using the fwtools shell command as below. In this command, "ogr2ogr"is a tool that can transform the projection of coordinates. It is followed by several command options where "-s_srs"indicates the source projection type, "EPSG:proj_src"indicates the current projection type of the *.shp file, "-t_srs"indicates the transformation projection source, "EPSG:4326"indicates the required projection transformation format, "reproj_fname.shp"is the reprojected filename, and "src_fname.shp"is the filename of the current source file that has to be transformed. Note that the projection transformation is always performed on *.shp file in case of shapefiles.

ogr2ogr -s_srs EPSG:proj_src -t_srs EPSG:4326 reproj_fname.shp src_fname.shp

For example, if the current coordinate projection is EPSG:27700, source shape file is srcfile.shp, required projection transformation is EPSG:4326, and reprojfile.shp is the reprojected filename, then projection transformation command looks like below:

ogr2ogr -s_srs EPSG:27700 -t_srs EPSG:4326 reprojfile.shp srcfile.shp

GML file generation

The next process after coordinate projection transformation is the generation of Geographical Markup Language (GML) [47] file. GML is a XML

based file format that is flexible, portable for graphs. GML is also easy to implement. DHIS2 is one of the example applications where this format is used to store the organization units and their (simplified) coordinates for each organization units before importing them into the system. Importing from GML format is one of many ways to import organization unit into DHIS2. Following command helps to generate GML file from the shape file [14]:

ogr2ogr -f GML filename.gml filename.shp

Here, "filename.gml"in above command is a new file in GML format and "filename.shp"is the file that has to be converted to GML file. While generating a GML file, organization units from *.dbf file that are connected with the coordinates in given filename.shp file are also included as data inside the GML file. After successfully running the command, GML file is saved in the same folder where shapefile is located.

The generated GML file as in figure 2.8 is still not ready to be imported into the DHIS2. In the file, organization unit is represented as <gml:featureMember>. Inside feature members, there are lot of attributes depending on the contents of *.shp and *.dbf files of a organization hierarchy level. Coordinate information about an organization unit is stored within <ogr:geometryProperty> tag inside <gml:featureMember>. End users should identify the attributes where organization unit name and other related information are stored and rename them according to the specifications of the importer of the DHIS2. The importer of DHIS2 looks for "ogr:Name"to access the name of an organization unit from GML file. If there is already information about organization unit in database and we just want to import coordinates for that organization unit then we need to identify the correct attribute where the organization unit name is stored and rename it to "ogr:Name". Moreover, the name should be exactly the name as in DHIS2 database. After doing all the required modification, the GML file will be ready for the import.

Import

In this process, user adds new levels as required. Some examples may be adding levels as country, province, district etc. After that, user prepares organizational level hierarchy in .csv or .xml format and uses import feature in DHIS2 to import organizational units. The user need to make sure that it has all the mandatory fields that are mentioned in DHIS2 user guide chapter 20.1.2, table 20.2 [15]. This process can be done independently if coordinates are not required. To import organization units along with their coordinates, or to only import the coordinates of the organization unit, GML file should be used. This process becomes the last process for this purpose.

```
<?xml version="1.0" encoding="utf-8" ?>
<ogr:FeatureCollection
     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
     xsi:schemaLocation="http://ogr.maptools.org/ SLE_adm0.xsd"
     xmlns:ogr="http://ogr.maptools.org/"
     xmlns:gml="http://www.opengis.net/gml">
  <gml:boundedBy>
    <gml:Box>
      <gml:coord><gml:X>-13.30350589752197</gml:X>
      <gml:Y>6.91761589050293</gml:Y></gml:coord>
      <gml:coord><gml:X>-10.26575279235834</gml:X>
      <gml:Y>10.00043106079102</gml:Y></gml:coord>
    </gml:Box>
  </gml:boundedBy>
  <gml:featureMember>
    <ogr:SLE_adm0 fid="F0">
      <ogr:geometryProperty><gml:MultiPolygon>
      <gml:polygonMember><gml:Polygon><gml:outerBoundaryIs>
      <gml:LinearRing><gml:coordinates>
      -11.534582138061353,6.944028854370401 -11.534582138061353,...
      </gml:coordinates></gml:LinearRing>
      </gml:outerBoundaryIs></gml:Polygon></gml:polygonMember>
      <gml:polygonMember><gml:Polygon><gml:outerBoundaryIs>
      <gml:LinearRing><gml:coordinates>
      -11.534582138061353,6.944028854370401 -11.534582138061353,...
      </gml:coordinates></gml:LinearRing>
      </gml:outerBoundaryIs></gml:Polygon></gml:polygonMember>
      </gml:MultiPolygon></ogr:geometryProperty>
      <ogr:ID_0>202</ogr:ID_0>
      <ogr:ISO>SLE</ogr:ISO>
      <ogr:NAME_ENGLI>Sierra Leone</ogr:NAME_ENGLI>
      <ogr:NAME_ISO>SIERRA LEONE</ogr:NAME_ISO>
      <ogr:EUmember>0.00000000000</ogr:EUmember>
    </ogr:SLE_adm0>
  </gml:featureMember>
</ogr:FeatureCollection>
```

Figure 2.8: GML file example generated after running the ogr2ogr command

Chapter 3

Methodology

This chapter is divided into two main sections: choice of research method and choice of data collection methods. The first section discusses about the ADR approach or research methodology used in this master thesis project and second section provides a number of data collection methods used to get information about existing systems and the requirements.

3.1 Action design research (ADR)

This section includes three subsections. The first section shows why ADR method is suitable for this project, the second subsection explains different stages of ADR method, and the third section explains how the project has adhered all the stages of ADR.

3.1.1 Why ADR methodology?

A design research (DR) approach seems to be suitable to address the research question of this project. However, according to Sein et al. [66], existing DR approach fails to recognize the artifacts that can emerge from the interaction of the organization. It considers the organizational intervention to be secondary. March and Smith [55] define DR as an approach "build and then evaluate"cycle i.e. evaluation is followed in sequence after development of the artifacts. Another approach that considers organizational intervention is action reserarch (AR). Davison and Martinsons [22] define AR as an iterative process based on working hypotheses refined over repeated cycles of inquiry. Sein et al. [66] believes that organizational intervention is necessary to recognize the new emerging artifacts and argue that the evaluation cannot follow the development artifact in a strict sequence. Hence, he propose Action Design Research (ADR) approach as a combination of DR and AR methods. Sein et al. [66] also defines ADR as "method needed to conduct DR that recognizes that the artifact emerges from interaction with the organizational context even when its initial design is guided by the researchers".

In order to solve the research questions of this project, We decided to develop an app that can help to import organization units and their

simplified coordinates into the HIMS so that thematic mapping can be enabled in the system even in resource constrained environments. The app should be generalized enough to import the organization units and their coordinates without interfering much into the core system. We consider the case of DHIS2 while developing the app. DHIS2 has been deployed as a solution to the health information management in different organizations of various countries. It is necessary to get regular feedback from the real users of the app (in our case, organizations where DHIS2 is deployed and the people working in core DHIS2 development team) and update the app accordingly for its successful integration in DHIS2. Therefore, we consider ADR approach from [66] as a more suitable methodology for this master thesis project.

3.1.2 Stages and principles of ADR

ADR method contains four stages and principles that guide the research processes. Figure 3.1 shows all the stages and principles of ADR and detail about them are provided in following subsections [66]:

Problem formulation

Problem formulation is the first stage of ADR and it helps to determine the initial scope, deciding the roles and scope for participants, and formulate the initial research questions on the basis of information from practitioners, end-users, the researchers, existing technologies, and/or review of prior research [66]. According to Hevner et al. [46], this stage identifies and conceptualizes a research opportunity based on existing theories and technologies. This stage is driven by two principles: *practice-inspired research*, and *theory-ingrained artifact*. The practice-inspired research principle emphasizes research inspired by practical problems. The action design researcher, instead of seeing problems as a software engineer or consultant, should generate knowledge that can be applied to the class of problems needed to be solved. The theory-ingrained artifact principle on the other hand emphasizes that the artifacts are created and evaluated through ADR and are informed by theories. According to Gregor [40], abstraction and generalization are the core of a theory and he says that the level of generality or scope of the theory includes specifying the boundaries. For example, the boundary could be *only* within specific problem domain if the scope of the theory is limited, or the boundary could be for *all* systems for a very general theory.

Building, intervention, and evaluation (BIE)

This second stage of ADR is an iterative stage of *building* the IT artifacts based on stage one, *intervention* in the organization, and *evaluation* of the artifacts. ADR tries to ensure relevance, novelty, and usefulness of a proposed artifact by considering three principles of BIE stage: *reciprocal shaping*, *mutually influential roles*, and *authentic and concurrent evaluation* [19].

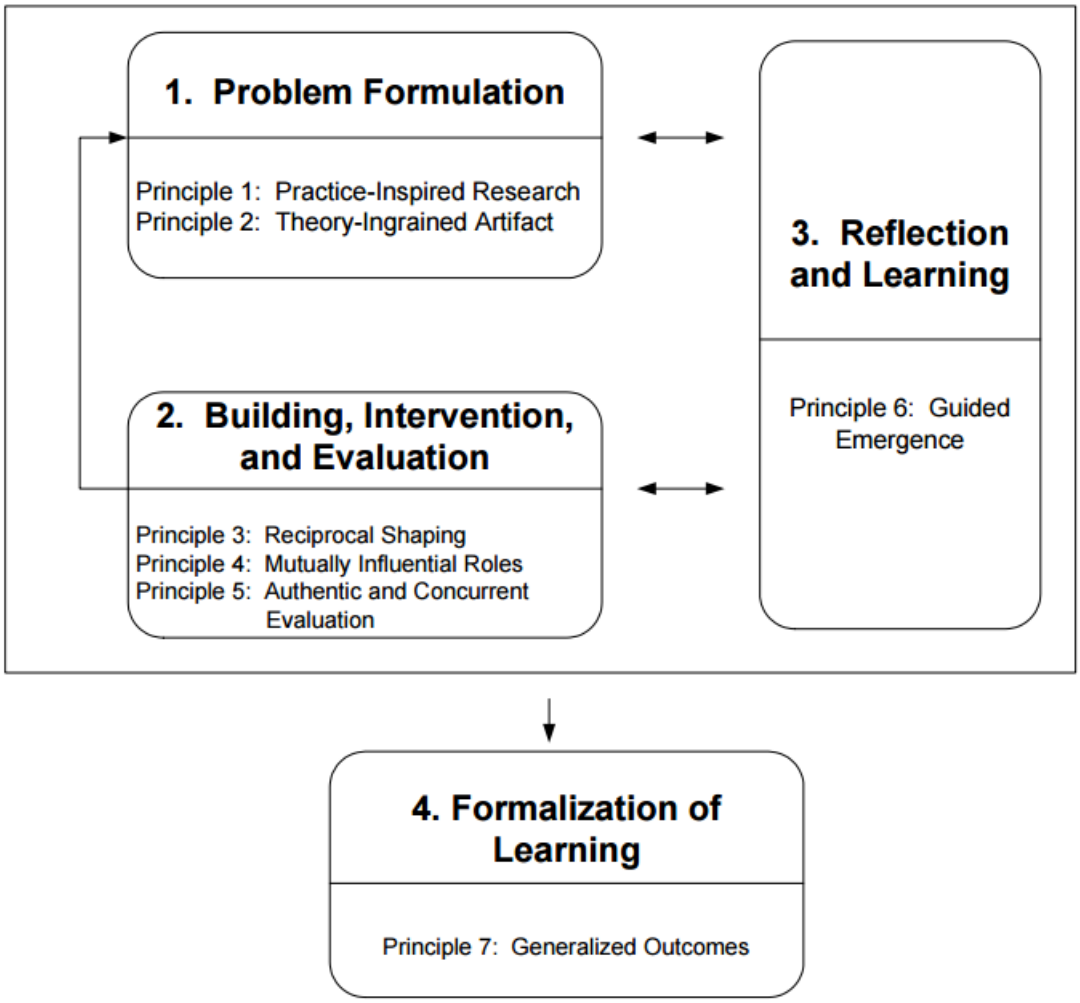


Figure 3.1: ADR method: stages and principles [66]

The reciprocal shaping principle emphasizes on inseparable influences between IT artifacts and the organization context domains, the mutually influential roles principle points out the importance of mutual learning among different participants of the project (for example, researchers and practitioners), and the last principle i.e., authentic and concurrent evaluation principle emphasizes that evaluation is not a separate stage of the research process that follows building [66].

Reflection and learning

This reflection and learning is a continuous stage of ADR that works in parallel with the first two stages i.e., problem formulation and BIE. This stage helps to reflect on design and redesign of the artifacts as a process of continuous learning. This stage is driven by *guided emergence* principle. This principle emphasizes that the ensemble artifact will reflect on both initial design and ongoing shaping by organization, participants,

and concurrent evaluation. It consequently provides an opportunity for the participants to generate and evolve design principles through the process [66].

Formalization of learning

This is the fourth stage of ADR method and it asks to develop the project in a more general way so that it can be suitable for a class of field problems. This stage is derived by *generalized outcomes* principle. This principle emphasizes on moving from the specific-and-unique solution to generic-and-abstract solution [66]. This conceptual move can be done in three levels: "i) generalization of the problem instance, ii) generalization of the solution instance, and iii) derivation of design principles from the design research outcomes"[66].

3.1.3 ADR for the project

This master thesis project adheres all the stages of ADR except that the communication for organization intervention is done through emails rather than meeting them individually. This subsection shows how each stages of ADR has influenced the development of this project.

Problem formulation

In order to formulate research questions, I consulted to my supervisor of this master thesis. We had meeting for several time to discuss about the issues in DHIS2. Finally, we formulated research questions, which are related to enabling thematic mapping for HIMS in resource constrained environment. Please refer to 1.5 for the research questions. We also formulated the ADR team with me as a researcher, my coordinator, implementers, GIS expert, and mailing list users. My coordinator himself is a researcher in DHIS2, implementers are responsible for installing DHIS2 in different countries, GIS expert is a person who is responsible to handle GIS related work in DHIS2, and the mailing list users are the users who are involved to DHIS2 from all over the world. Both practice-inspired principle and theory-ingrained artifacts drive our research. We consider generalization as a theory to our research problem and our research questions should come up with a generic solution that can address mass number similar applications with few domain specific changes.

Building, intervention, and evaluation (BIE)

This research work has also followed the iterative process between building the artifacts, organizational intervention, and the evaluation of the artifacts. All the feedbacks from the end users and organization are collected via email. After analyzing the current problems for importing organization hierarchy with their coordinates in DHIS2, an Alpha version of the app was developed that used to allow users to upload the shapefiles 4.1.1, simplify

the coordinates and then automatically insert the data into the DHIS2. This app was sent to Srilanka through email for evaluation. They had some problem with this app at first and after analysis, we found that the problem was due to duplicate data in the shapefiles. We asked them to do some modifications on their shapefiles manually and use the app for import. This time, they could successfully import the data through the app. Manual update was a quick solution to Srilanka, but we knew that we need to perform duplicate checking on organization units before importing them to the database. The same app was sent to Madagascar, Nepal, and East Timor but we did not get any feedback from them.

We also sent this app to the mailing list users. This time, they gave us different feedback; they wanted to import some of the selected organization hierarchy only. I also observed some functionality that has to be added in the app. For example, the DHIS2 system allows only 4 digits after decimal point for coordinates in order to store into its database because of which it was necessary to convert coordinates to this format. It was also necessary to allow users to check the correct parent organization unit; otherwise, there was possibility of importing wrong levels of organization units into the system. The idea of allowing end users to modify their parent records would also be useful, since end users are familiar with the data and they know what the parent should be for a particular record.

All these error checking and functionality were added in next version of the app (Beta version) and it was sent for evaluation to a core developer of DHIS2, and mailing list users.

Reflection and learning

Through BIE stage, we got opportunity to learn and evolve our app to effectively import the organization units with their coordinates. At the same time, practitioners also got opportunity to import shapefiles in a easier and quicker manner. The app was redesigned continuously to meet the requirement of different practitioners (implementers, mailing list users, and core developer of DHIS2). BIE cycle was iterated for several time by updating the app to reflect the requirements of the practioners.

Formalization of learning

The format for the shapefiles (especially *.dbf) can vary from one file to another. The app developed is generic to handle any format of shapefiles. This app also allows users to map the columns from *.dbf file to columns (fields) of the database of a particular application in order to generate the format necessary to import data into the system. The second part could be specific to the application, since the data from shapefiles should be mapped with the database fields of specific application. Therefore, we can say that the app is designed in a generic way to import shapefiles of any format but it can be fully used by making few changes for mapping the data specific to the applications.

3.2 Data collection

During the research process, data collection has been very crucial part. In our case, observations, document studies, key informants, tests, and case study are the main data collection methods. More on how these methods are used for data collection in this thesis is discussed below:

3.2.1 Observations

According to Frechtling [37], observational method helps to gather data on a wide range of behaviors, to capture a great variety of interactions, and to openly explore the evaluation topic. These approaches also allow the observer to learn about issues the participants is not aware of or unable to discuss candidly in the interviews. During this research project, I followed the observation method while understanding the context of web thematic mapping for public health in resource constrained environment and while exploring the workflow of import functionality of DHIS2 system. I collected firsthand data, understood the process, and came to know about different issues that the current users of DHIS2 are facing with the import functionality. These issues become more critical in resource constrained environment and I found that there is a need of a better way to import organization units even in resource constrained environment.

3.2.2 Document studies

Study of the documents related to DHIS2 helped me a lot to gain insight about the DHIS2 processes. Lincoln and Guba [41] defined a document as any written or recorded material that are not prepared for the purposes of the evaluation or at the request of the inquirer. In our case, the documents that are provided for the DHIS2 [13] were studied to understand the workflow of import functionality in DHIS2. It also helped me to get information about different available tools and how they can be used in different stages of import functionality. It further helped me to test the tools for our app.

3.2.3 Key informants

According to Frechtling [37], "key informant is a person who has unique skill and professional background related to the issue/intervention being evaluated, is knowledgeable about the project participants, or has access to other information of interest to the evaluator". Frechtling [37] also adds that the members of such group may be specifically selected or invited to participate because of their skills or professional backgrounds. In my thesis, DHIS2 user and developer mailing lists were considered as key informants. I used to get emails for the issues related to the import functionality of DHIS2 from these mailing lists. DHIS2 user mailing list contains users who have experience in using DHIS2 platform for their applications and DHIS2 developer mailing list contains experts

who are involved in the development of DHIS2. Information from these groups were really helpful to understand the issues in DHIS2 for import functionality, which further helped me understand the issue in general. I also knew about their expectations from the new app that I was going to implement.

3.2.4 Case study

According to Frechtling [37], case study provides very engaging, rich explorations of a project or application. Since my thesis is related to the web thematic mapping for public health in resource constrained environment, I found DHIS2 as a suitable case to be explored. During the case study, I came to know about different stages of importing organization units and their coordinates in the system, how difficult and tedious some processes are, and at the same time, I also some possible solutions for them.

3.2.5 Tests

According to Frechtling [37], tests methods provide a way to assess subject's knowledge and capacity to apply this knowledge to new situations. They also provide information that is measured against a variety of standards. Since I considered DHIS2 as a case for my research project, I had to install DHIS2 in my local computer and check all the processes and functionality related to the import feature. I also tested real data from different users to check the real issues. In addition, I also tested some tools such as Mapshaper, and proj4js to check if they can be integrated into one web app or not in order to import organization units and their coordinated into the DHIS2 system. During this testing method, I came to know about different functionalities, advantages, and disadvantages of these tools, which further helped me to develop the app.

Chapter 4

Design and Implementation

This chapter gives an overall idea about the implementation of the app for DHIS2 that can be used as a solution to solve the research problems of this master thesis. The implementation of the app is based on the data and feedback collected from different group of people (implementers, DHIS2 mailing list users, DHIS2 developers), and analysis of existing import functionality of DHIS2 system. With the problem description and requirement for the app from various sources in hand, we decided to implement the app in two phases namely as Alpha and Beta versions. Alpha version is the first implementation and Beta version is the second implementation. First section of this chapter provides details about all the tools that are used in different processes of import feature of the app. Details about Alpha version of the app is provided in the second section and Beta version in the third section. The last section of this chapter provides information about the tools and programming languages that are used for the implementation.

4.1 Choice of tools

This section provides the methods, concepts and tools that will be used for the implementation of an app to import organization unit hierarchy from layered shapefiles.

4.1.1 Shapefiles

A popular file format to store the vector data format for geographic information system (GIS) is Shapefiles[12][6]. Data contains the nontopological geometry and attribute information for the spatial features. The geometry for a feature is stored as a shape comprising a set of vector coordinates.

It supports the point, line and area feature. In this format area are represented as closed loop, double-digitized polygons. Attributes are held in a dBASE format file. Each attribute record has a one-to-one relationship with the associated shape record. The shapefiles contains the group of files that stores the geometry and attributes of geographically referenced features in three or more files with specific file extensions that should be stored in

the same project workspace. The individual file with different extension should have the same prefix. List of files are:

Mandatory files:

- filename.shp files: This is the main file that stores the feature geometry
- filename.shx: The index file that stores the index of the feature geometry
- filename.dbf: The dBASE table that stores the attribute information of features. The attribute values can be name and other details of the shape.

There is a one-to-one relationship between geometry and attributes, which is based on record number. Attribute records in the dBASE file must be in the same order as records in the main file.

Optional files:

- filename.prj: The file that stores the projection information of coordinates
- filename.sbn and filename.sbx: The files that store the spatial index of the features.

There are more optional files, which can be viewed in [11].

We can use shapefiles as a source of data for maps but the way of storing information into the files may differ with the providers of the shapefiles. For example, information about organization units such as name, parent organization unit, and ID are stored in *.dbf file of the shapefiles and column names that are used to refer these information may be different and the number of columns may vary with *.dbf files. Another option to shapefiles could be tools like PostGIS [61], which is a spatial database extender for PostgreSQL object-relational database. This has more complex structure than shapefiles.

Shapefiles are used by the users for importing the organization units into DHIS2 by converting the data from shapefiles into GML files manually. Please refer to [14] for checking the process of importing organization units using GML files in DHIS2. Since the shapefiles are already well known and used by the users, we decided to use the shapefiles to make an app for automate this process of making GML files from shapefiles and importing organization units and their coordinates into DHIS2 system. This is helpful for users for importing the data by reducing the time required for learning the new tools and process.

4.1.2 GADM shapefile

GADM[38] is a spatial database of the location of the world's administrative areas (or administrative boundaries) for use in GIS and similar software.

Administrative areas in this database are countries and lower level subdivisions such as provinces, departments, bibhag, bundeslander, daerah istimewa, fivondronana, krong, counties, and thana. GADM describes where these administrative areas are (the "spatial features"), and for each area it provides some attributes, such as the name and variant names.

4.1.3 Converter

The only coordinate system that DHIS2 consumes is EPSG:4326 format. One another coordinate system other than EPSG is Universal Transverse Mercator Coordinate System (UTM) [30][53]. It is a coordinate system that was developed on year 1984's by the Corps of Engineers, U.S. Army. The UTM projection defines horizontal positions world-wide by dividing the surface of the Earth into 6 degree zones If the shapefile does not contains coordinates in EPSG:4326 format then they have to be converted into the EPSG:4326 first, in order to import them into the DHIS2 system. EPSG:4326 contains geographic latitude and longitude as data. In order to convert coordinate system, we need to know Coordinate Reference System (CRS)[18] and projection [36][53] of shapefile which is defined in *.prj file. This file is made available while downloading original shapefiles for a map [14]. Projection of map is a method by which the curved surface of the earth is portrayed on a flat surface. This generally requires a systematic mathematical transformation of the earth's graticule of lines of longitude and latitude onto a plane. [36][53].

In order to convert the coordinate system to EPSG:4326 format, it is important to understand about the tools that are already available for conversion. List below provides some of such tools:

- http://trac.osgeo.org/proj4js/
- http://leafletjs.com/

It is possible that sometimes .prj file being missing. In such situation, it becomes difficult to find out coordinate reference system of the shapefile. Systems that can help in finding coordinate reference system of shapefile in the absence of .prj file are mentioned below:

- http://georepository.com/crs_32645/WGS-84-UTM-zone-45N.html
- http://wiki.openstreetmap.org/wiki/OpenLayers_Simple_Example

Coordinates are stored in .shp file and information about level and hierarchy of organizational units are maintained in .DBF file. Couple of links to check the link between .DBF and .shp are provided below:

- https://github.com/abstractvector/node-dbf
- https://github.com/wavded/js-shapefile-to-geojson

4.1.4 Mapshaper

Mapshaper is a tool that helps to simplify the complex and bulky map into simple and less bulky map that can be loaded fast when the user does not need very detail map and level of acceptance is high. Using the slider provided user can choose the level of detail that is required. For this thesis, using this app we are importing the coordinates and the organization levels into the dhis2 system.

Simplification of coordinates

One of the objectives of this thesis is to allow the users to simplify the map while importing data and the coordinates into DHIS2 system. For this purpose, we plan to integrate the Mapshaper tool as an app for DHIS2 system. Mapshaper is an online tool that can be used to generalize the geographical data that we find in the shapefiles. This tool can be accessed in [54]. Any original shapefiles contains geographical data in detail. In many cases, detail geographical data may not be necessary and it is especially very hard to load the map where the internet connection is very slow. Therefore, in order to reduce the size of the file with geographical coordinates, we will integrate the Mapshaper. It helps users to get the desired level of detail data to generate the map. Figure 4.1 provides general system architecture of Mapshaper tool.

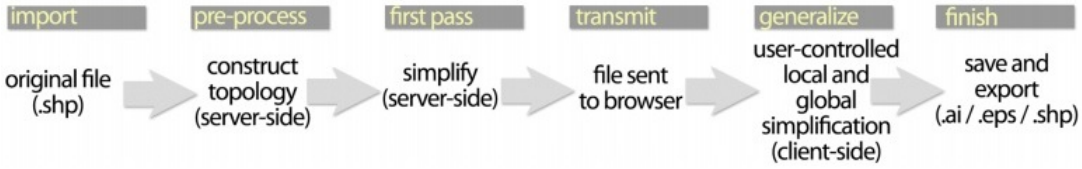


Figure 4.1: General system architecture of MapShaper tool [45]

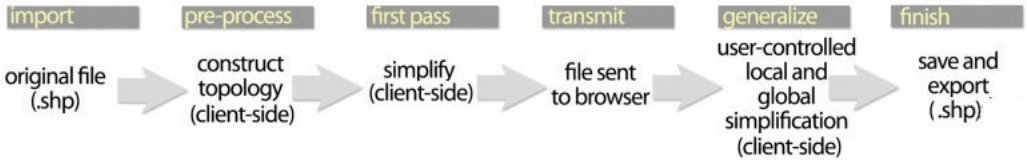


Figure 4.2: General system architecture of MapShaper tool [45]

According to the Figure 4.2, user can import original .shp file into the system and system performs pre-processing and first pass simplification automatically. Then it sends file to the browser. In client-side, user can control local and global simplification and then export final simplified result into .shp format.

Figure4.3 shows an example of using Mapshaper v0.1.13 tool. It shows a map generated after importing original shapefile. This tool provides a slider just above the map and it is named as 'Simplify'. This slider ranges

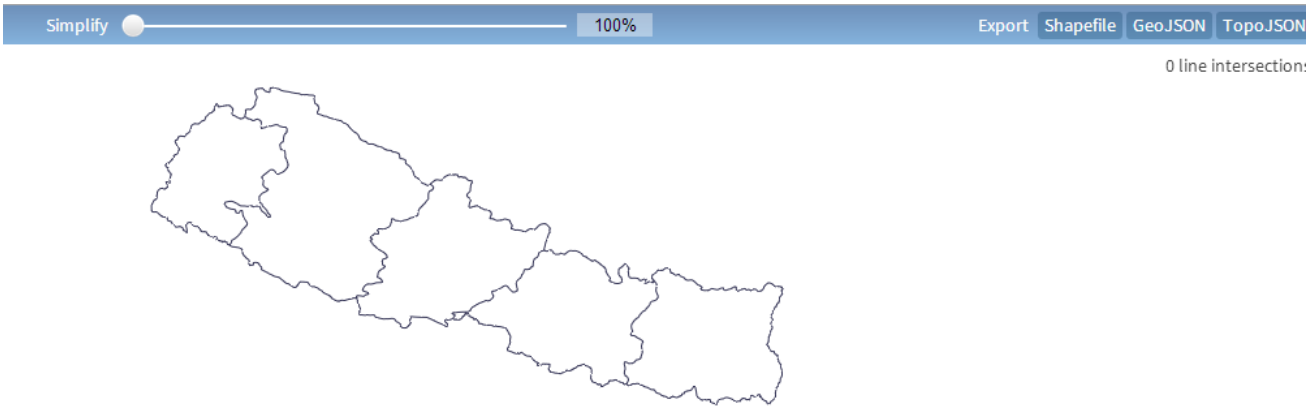


Figure 4.3: Example of using MapShaper tool

from 0% to 100% and it is on full control of users. The more percentage of simplification, the more it removes geolocations and becomes more simple. Once desired level of simplification is obtained, user can export the new generated results in three different formats i.e., ShapeFile, GeoJSON and TopoJSON file format. With the updated version, users can now install Mapshaper in the local system as well.

4.1.5 Geography Markup Languagee (GML) file format

Geography Markup Language (GML) [9] [8] is a XML grammar that is written in XML Schema, for expressing, storing and transporting the geographic information. It serves as the language for modeling geographic systems. GML is open interchange format for geographical transaction on the internet.

In DHIS2, while importing the coordinates into DHIS2 system, the data from shapefiles are converted to GML file that will contain all the organisation units with their coordinates. Users are using this file format to import the coordinates into DHIS2. With the view that it will be easy to use the same file format and making it possible to generate the data from shapefiles in the GML format automatically, we can remove the process of manual work of converting the shapefiles to GML in Mapshaper.

4.2 App Design: Alpha version

In order to come up with our first alpha design for the app, we did some requirement analysis and designs. This section is divided into three subsections to explain different stages of design to reach the Alpha version of the app.

4.2.1 Identifying the main processes

In order to define our objectives for the design and implementation of our Alpha version of the app, we collected some specific requirements based

on the analysis of existing import functionality of DHIS2 system. They are listed as below:

- Change the projection of shapefiles (DHIS2 uses projection EPSG:4326).
- Upload the shapefiles to Mapshaper to simplify the coordinates as per the user needs.
- Import the coordinates into the DHIS2 system with organization hierarchy and coordinates. Allow users to specify the child parent relation.

Based on these requirements, we identified three main processes to import the organization hierarchy from shapefiles into the DHIS2: *projection check and transformation*, *coordinate simplification*, and *importing data*. These processes are depicted in version A design of figure 4.4.

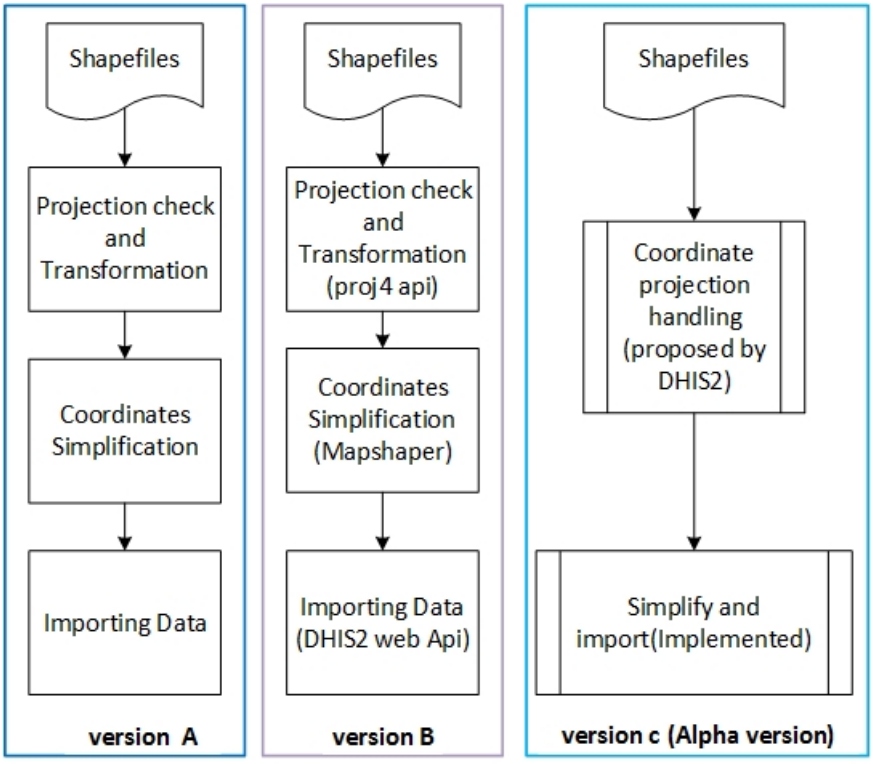


Figure 4.4: Chain of design versions for Alpha version of the app

The first process needs to check the projection of the shapefiles and transform them if they are not according to the specification of DHIS2. Another process is to simplify the coordinates we get from the shapefiles, and the last process is to import the data from shapefiles along with their simplified coordinates into the DHIS2 system.

Based on the requirements we gathered from the analysis of existing import functionality of DHIS2, we defined our objectives in order to get

the final Alpha version for our app. The objectives for Alpha version are listed below:

- Check different tools and api that we found from our own observation and also from the web mailing list that are related to import the shapefiles into the HIMS (DHIS2 in our case).
- Check possibility of integrating the selected tools and api into the app.
- Build an app that can change the projection of the coordinates, simplify them and import those coordinates along with organizational hierarchy into the fresh installed DHIS2 with no organizational hierarchy.

4.2.2 Identifying the tools and api

There were many open source and commercial tools available for coordinate transformation and simplification of the coordinates. Some of them are provided in table 4.1.

Since we were planning to give a solution to help users in resource constraint environment to enable the thematic mapping, our obvious choice was to choose and test the open source solutions so that the users do not need to pay anything for using the services and tools that are provided by different private companies. Moreover, we also planned to select tools and api that provide javascript libraries, since it is easy to integrate with DHIS2 Web API. We choose some tools and api on the basis of our observation and the practice that was followed by the DHIS2. These selected tools and api are listed in table 4.2. Selection of api and tools for each main processes of the app are shown in version B of figure 4.4.

Detail about these tools while implementation of alpha version are provided in following subsections:

Proj4js for transforming the projection

For Proj4js [62], I was able to get well-documented api and it worked well to transform coordinates for already known projection of shapefiles. In case of shapefiles, projection related information are stored in *.prj file. But upon closer analysis and observation, I found that *.prj files are not common for all the shapefile providers. Without knowing the projection of shapefiles, it was difficult to transform the coordinate projection into desired projection correctly. ArcGIS on other hand can transform into the desired projection of coordinates without knowing the current projection type of shapefiles in advance [35], but ArcGIS is a commercial tool which is out of our scope of selecting tools and api. Therefore, for the projection transformation, we decided to stick with the older manual process that was followed by the users and proposed by DHIS2. Please refer to the section 2.3.3 to check how users can do the coordinate transformation using the method proposed by DHIS2. After facing this issue, we decided to implement proj4js later, or

To convert the coordinates projection		
SN#	Tools/api	Description
1	Proj4js [62]	Provides an open source javascript library that can be used to transform the projection of the coordinates.
2	Fwtools [4]	Open Source GIS binaries for Windows (win32) and Linux (x86 32bit) systems. It contains lot of packages such as OpenEV, GDAL, MapServer, PROJ.4, OGDI etc.
3	OpenLayers [59]	By default, OpenLayers helps to transform coordinates between geographic (EPSG:4326) and web or spherical mercator (EPSG:900913 et al.) coordinate reference systems. It uses proj4js library to allow transforms between arbitrary coordinate reference systems.
3	ArcGIS [33]	A commercial GIS engine that can also be used to identify the coordinate projection type and also to transform the coordinate projection.
To simplify the coordinates		
SN#	Tools	Description
1	Mapshaper [39]	An open source tool where shapefiles can be uploaded and coordinates can be simplified.
2	HealthMapper	A desktop based application which provides preinstalled coordinates for all the countries.
3	ArcGIS [34]	Commercial tool that also includes coordinate simplification feature.

Table 4.1: Tools and api to convert the coordinates projections and simplify them

SN#	Tools/api	Description
1	Proj4js api [62]	For coordinate projection transformation.
2	Mapshaper tool [54]	To upload the shapefiles and simplify the coordinates.
3	DHIS2 web api [26] [33]	api provided by DHIS2 that has the functionality of accepting the GML file to import the organizational hierarchy and coordinates into the DHIS2 system.

Table 4.2: Selected tools and api to convert the coordinates projections, simplify them, and import data into DHIS2

find some other solutions that can help users with projection identification and also transform coordinates, or find some other tools that can identify the projection of the shapefiles with or without the *.prj files as well.

Mapshaper to simplify the coordinates

For Mapshaper, we did not find any documentation regarding its integration into the app. It was a tool in itself. Therefore, we had to check the coding of the tool and find out a way to integrate this tool for uploading and simplifying the shapefiles. Since Mapshaper is an open source tool, I was able to get all the files and codes that the Mapshaper had used. Based on my previous knowledge of software development, I used some debugging tools like firebug, notepad++, and checked the coding done for the Mapshaper to answer the following questions:

- What are the mandatory files of shapefiles to display the map in Mapshaper?
  I found that *.shp file is enough to display map on Mapshaper tool, but it is better to upload *.shp and *.dbf files together if someone wants to export the simplified coordinates and their corresponding organization hierarchy at the same time. It is possible, since the Mapshaper tool provides a feature to export the updated files at same time.

- How does Mapshaper handle the uploaded files feature and how it reads the files?
  After user upload *.shp and *.dbf file for a level of organization unit, Mapshaper reads the values from these files. A map is displayed on the screen based on the coordinates from the *.shp. Files canbe uploaded by clicking the "select"button provided by the tool itself or we can just drag the files into the page. (please refer to the section 8.1 in Appendix for more detail)

- How does Mapshaper store data from the *.dbf and *.shp files?
  Mapshaper stores the data from uploaded *.dbf and *.shp files into a file object with the help of *MapShaper.exportContent(layers, arcData, opts)*. For example, in a code snippet of figure 4.5, a file object "file"is created to store the data. This operation can be found within the "callFunction"function of "mapshaper.js".

- How the coordinates of an organization unit and details about organization hierarchy are linked in Mapshaper?
  It is easy to work with JSON object than with simple file object. Therefore, the data stored in file object variable "file"can be converted to JSON object "objJSON"variable as in figure 4.6, and this JSON object variable can be used to perform tasks related to coordinates and organization hierarchy.

  Each organization units in the *.dbf file are linked with their coordinates in *.shp file by common row order starting from 0.

```
function callFunction(format, done, functionName) {

    var opts = {
        output_format: format,
        output_extension: MapShaper.getDefaultFileExtension(format)
      },
      files, file;// files and file contains the data from dbf file and shp files.
    Utils.extend(opts, options);
    files = MapShaper.exportContent(layers, arcData, opts);
```

Figure 4.5: Code snippet to store uploaded data in a file object

```
file = files[0];
var jsonString = file.content;
var objJSON = eval("(" + jsonString + ")");
```

Figure 4.6: Code snippet to convert file object variable to JSON object variable

It means the coordinates at row number 5 of *.shp file are the coordinates of organization unit at row number 5 of *.dbf file. Since these data can be stored in a JSON object "objJSON", they can be accessed using the code snippets given in figure 4.7 and figure 4.8. In both code snippets, "RowID"denotes the order of the row in *.dbf file that also link the corresponding coordinates in the *.shp file. The code snippet in figure 4.7 helps to get the data of a particular column (key) of a organization unit (RowID), and the code snippet in figure 4.8 helps to get all the coordinates of the same organization unit (RowID).

- How can I use and display the data stored by Mapshaper from *.dbf file to the users?
  After identifying the relevant objects and the properties for storing the data in the Mapshaper, a simple loop can be used for the code snippets in figure 4.7 and 4.8, to display the data stored by the Mapshaper tool.

DHIS2 Web API to import data into DHIS2

Since we had taken DHIS2 system as a case for our research, it was necessary to identify how we can store data related to organization units and their coordinates into the database of DHIS2. I found that DHIS2 has developed its Web API [26], allowing users to develop applications on top of the Web API and interact with the data stored. In order to take advantage of DHIS2 Web API, I had to study how I can integrate it in our app. For this, I studied the Web API documentation of DHIS2 [26] and found out the key methods that are useful to connect our app and Web API and to understand the format that the Web API accepts to store the data into the DHIS2. These methods are provided below:

```
objJSON.features[RowID].properties[key]
```

Figure 4.7: Code snippet to access the properties of a organization unit

```
objJSON.features[RowID].geometry.coordinates
```

Figure 4.8: Code snippet to access coordinates of a organization unit

- DHIS2 Web API can be enabled by including "dhis2.util.js"library in the app [23].
- DHIS2 Web API accepts GML file format whose structure is similar to the structure shown in figure 4.9.

```
<metaData xmlns='http://dhis2.org/schema/dxf/2.0'>
    <organisationUnits>
        <organisationUnit
            shortName = 'SLE'
            code = 'Sierra Leone'
            name = 'Sierra Leone'
            level='1'
            featureType='MultiPolygon'
            coordinates= '[[[[-12.6276,7.6415],[-12.5693,7.5476],
                          [-12.5837,7.4546],[-12.9140,7.5646],
                          [-12.9057,7.6060],[-12.6276,7.6415]]]]'>

            <parent id=''/>
            <active>true</active>
        </organisationUnit>
    </organisationUnits>
</metaData>
```

Figure 4.9: GML file format for DHIS2 Web API

The code snippet 4.9 provides a basic structure of the GML file format to import organization units and their coordinates into the DHIS2 system with the help of DHIS2 Web API. Variables such as "shortName", "code", "name", "featureType", "coordinates", "featuretypes", and "coordinates"in 4.9, are the database field names of DHIS2 to store the corresponding values from the shapefiles.

- It is possible to use AJAX script to use DHIS2 Web API to import the data (organization units and their coordinates) into the DHIS2 system. For that, We need to specify the URL as "<SERVER_LOCATION>/api/metadata", use "xml"content types, and use "POST"method. "<SERVER_LOCATION>"is the server location where DHIS2 is hosted. A small code snippet for this purpose using AJAX script to send the data to the Web API of DHIS2 which

is again responsible for importing the data into the DHIS2 system, could be as in figure 4.10.

```
function postBulkDataValues(data){

    if(!data){
        data = E("valutxt").value;
    }

    $.ajax({
        url: '../../api/metaData',
        headers: {
            'Content-Type': 'application/xml',
            'Accept' : 'application/xml'
        },
        type: "POST",
        cache: false,
        crossDomain: true,
            data: data,
            xhrFields: {
                withCredentials: true
        },
        success: function(data) {
            alert('Data Updated Sucessfully!');
            //hide the loader
            canvasLoader.hide();
        }
    });
}
```

Figure 4.10: AJAX script to link DHIS2 Web API and import the data into DHIS2

4.2.3 Implementation of Alpha version of the app

Building a first level app (Alpha version) was one of our objective derived from the requirement analysis of existing import functionality in DHIS2 system. Since we decided to keep the projection transformation process of DHIS2 as it is, we wanted to implement coordinate simplification and import of shapefiles data into DHIS2 in Alpha version (version c (Alpha version) in figure 4.4) of our app. This app should provide users an interface from where they can simplify the coordinates of organization hierarchy as well as can choose the parent and child relation between different level of shapefiles, while importing the organization hierarchy with their coordinates into DHIS2. Figure 4.11 shows detail implementation design for the Alpha version of the app.

Figure 4.11 shows three main processes: *projection handling*, *simplify*,

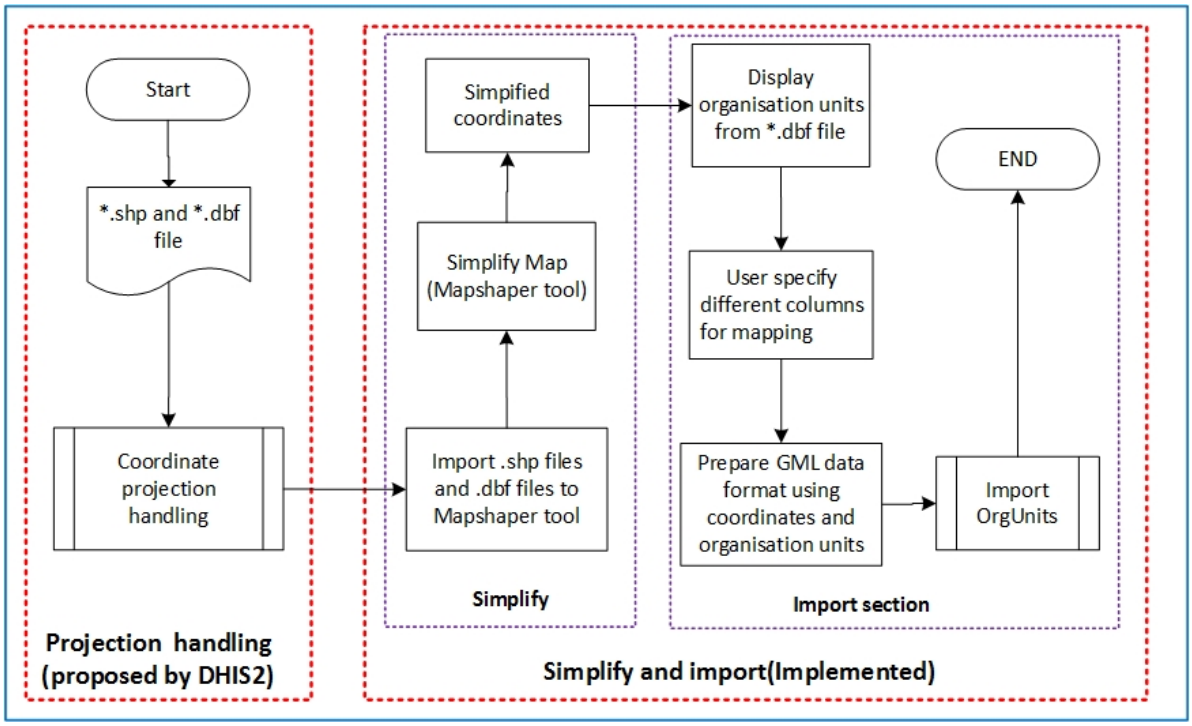


Figure 4.11: Implementation design for Alpha version of the app

and *import*. Here, projection-handling process works in a same way as in existing import functionality of DHIS2. Simplfy and import functionality has been implemented in Alpha version. This section is sub-divided into two subsections based on implemented functionality in Alpha version of the app:

Simplify the coordinates

We selected Mapshaper tool to simplify the coordinates of the organization hierarchy. By now, we know that Mapshaper is a standalone tool that helps to import files like *.dbf and *.shp and shows a map in an interface. As shown in simplify process of figure 4.11, end users after obtaining proper projection for the shapefiles can upload *.dbf and *.shp files into the app where Mapshaper tool is integrated. With the help of Mapshaper tool from this app, users can simplify the coordinates. Mapshaper provides functionality to export the simplified coordinates along with their corresponding organization units into various formats such as Shapefile, GeoJSON, and TopoJSON. Users can download updated shapefiles in different format by using the property of Mapshaper. Importing these updated information into DHIS2 system automatically has been implemented in the app. We will discuss about it in next subsection. Please refer to the section 4.1.4 for more details about the Mapshaper tool and section 8.1 for more information about how to upload shapefiles and simplify the coordinates using the app.

Import shapefiles data into fresh DHIS2

We first wanted to import shapefiles content related to organization units and their coordinates into fresh DHIS2 system (DHIS2 system without any organization hierarchy). This app helps to display all the organization units from the uploaded *.dbf file and then allow users to map the columns of this file with the database fields of DHIS2. After correctly mapping the file information with DHIS2 system, our app helps to prepare a GML file format that includes all the mapped organization units from *.dbf file and their corresponding simplified coordinates from *.shp file. The app then imports the GML data into the DHIS2 with the help of it Web API. Figure 4.12 shows detail about this "import OrgUnits"process.

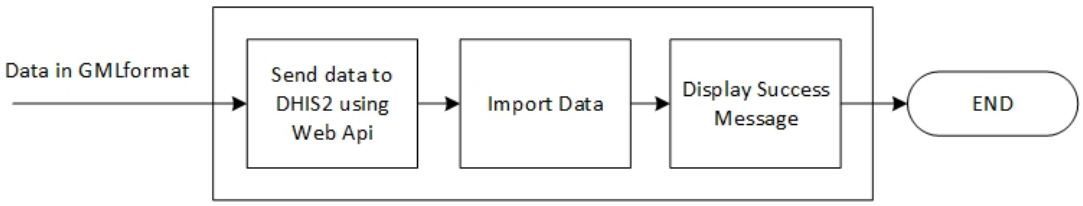


Figure 4.12: Import organization units into DHIS2

When DHIS2 Web API into DHIS2 successfully imports GML data, a success message is displayed in the app and the process stops. Please refer to the section 8.1 for more detail about displaying the organization units, mapping *.dbf file content with DHIS2 database fields, preparing GML file format, and sending GML file to DHIS2 Web API.

4.3 App Design: Beta version

From the implementation of Alpha version of the app, we came to know that it is possible to integrate Mapshaper tool to simplify the coordinates within the app, which again can be connected to DHIS2 system with the help of its Web API in order to import organization hierarchy related data in the form of GML data format. However, the Alpha version was implemented to import data for the DHIS2 database that did not have any organization units, which may not be the valid case for all. The Alpha version was sent for evaluation to different implementers, and mailing list users. With the help of their feedback and our own observation, we prepared a list of requirements/objectives/functionality to be added in the new Beta version of the app.

This section is divided into three subsections. The first subsection explores different possible conditions that has to be checked while mapping shapefiles and DHIS2 database. The second subsection explains the reasons for adding each functionality in the Beta version and the third subsection provides a detailed implementation design of the Beta version.

4.3.1 Possible matching conditions

In existing DHIS2 system, GML data is imported into the system by matching on the name of the organization unit [15]. It means, different possible conditions should be checked related to the name of the organization units before importing the shapefiles into the DHIS2 system, otherwise we might provide wrong data to the import procedure. From our observation, we found three possible matching conditions: *unique match*, *no match*, and *duplicate match*. They are explained below:

- *Unique match*: This is the condition that has to be checked when information about an existing organization unit has to be changed, or it has to be inserted and linked with the proper parent organization unit, while importing the shapefile contents. In first case, the organization unit should be uniquely identified and in later case, the parent organization unit for that particular organization unit has to be identified. In Beta version, these unique matching are performed by uniquely comparing the combination of organization unit name and code. There can occur conflicts in both case if multiple occurrences are found. Note that, organization unit at root level does not have any parent and there is no need to find the unique parent.

- *No match*: No match condition can occur either for a new organization unit that does not exist in the DHIS2 or for already existed record but with different name. There is no problem to import new organization units but the other case should be handled properly. This case is difficult to handle automatically and may result in wrong data insertion if it is done automatically. Therefore, we allow users to check for the existing records before importing into the system. This reduces the risk of importing wrong information or multiple records for the same organization unit.

- *Duplicate match*: Existence of multiple organization units with same name results in duplication of record. This condition should be avoided, since it can update all the duplicate records even when the update is meant for only one organization unit. There can be duplicate organization unit name in same hierarchy level or same parent organization unit name in immediate parent level. These conditions should be checked both in DHIS2 database as well as in *.dbf file before the import process.

4.3.2 New import functionality

Based on the observation about possible matching conditions from subsection 4.3.1, and with a vision of providing end users an interactive user interface to address the possible issues during import process of the organization units into the DHIS2, we formulate some functionalities for the Beta version. Details about these objectives are provided below:

- Allow users to update the coordinates
  After getting the feedback from the users, I found that lot of users have already installed the DHIS2 and have their existing database with the organization units. End users of this category usually try to update the current coordinates of the organization units into the database. Earlier Alpha version was helpful for adding the new organization units and their coordinates only in the freshly installed database of the DHIS2. These requirements resulted in the decision to add the feature that can allow users to update the existing coordinates into the database.

  In order to add this feature in the Beta version, I had to check what DHIS2 Web API has to offer. It was mandatory that we match the name of the organization unit exactly [15]. If there were more than one organization units with the same name, it was possible that all organization units would be updated with the same coordinates.

  After fair analysis and testing of different GML file, it is necessary to match the organization units name from *.dbf file with the organization units name in the DHIS2 database, since coordinates are linked with each organization units. While matching the organization units name, Beta version provides two options to update the coordinates. The app displays the matched organization unit to the end user if the organization unit's names are matched with the organization units in DHIS2 database. Otherwise, the app list all the organization units of the same level of hierarchy from the database into a select box and allow users to choose the proper matching organization unit. The code and id of the selected organization unit is then used to update the coordinates of that particular organization unit in the process of importing the GML file format. If the user does not specify any organization unit in case of no match, the app inserts that organization unit along with its coordinate as a new record into the DHIS2 database.

- Allow users to choose particular organization units from the list
  Another feedback was to be allowed to choose few organization units for import instead of importing all of them that are available in the shapefiles. This case might occur when end users miss some organization units into their DHIS2 system. In order to provide this functionality to the users, HTML check boxes are added in the app for each organization units from the *.dbf file. The app updates/inserts organization units whose check boxes are checked for import. Again, before importing the selected organization units, duplicate records can be checked and the options to select the proper organization units can be provided.

- Error and success message handling
  A proper message handling process is important for end users to understand the process or the effect of their action in any application. The Beta version of the app handles two types of messages: error

messages and success messages. Error messages are displayed when the app finds error such as duplicate records, parent not selected for hierarchy level more than 1, and parent not found in the DHIS2 database etc., whereas success messages are displayed when the app successfully performs some actions such as successfully importing the data. In order to clearly distinguish these two types of messages, error messages are displayed with red color and success messages are displayed with green color.

- Ask users to provide the level of hierarchy before importing the shapefiles
  Mapping correct parent and organization unit for each organization units in the shapefiles into the DHIS2 system can be very time consuming. The mapping process time increases with the increasing number of organization units available in the DHIS2 database. Therefore, it is important to narrow down the number of organization units to be mapped in order to import the data quickly into the system. By allowing users to provide the level of hierarchy, we can limit the scope of mapping around organization units that have the same level. If we have to find a particular organization unit from the database and the level of hierarchy is given, we can map the organization unit within that level only. This way, we can skip other organization units from mapping and the mapping process becomes faster. The same is also valid for identifying the parent of a particular organization unit. It is faster to search immediate parent of an organization unit, since the parent organization unit contains one level less than the child organization unit does.

- Allow users to specify the correct parent
  Allowing users to specify the correct parent before importing the data into the system is also an important step. It is possible that the app will not be able to find parents of all the organization units from the database. Moreover, it is also possible that the app will find wrong parents for the organization units. This can happen due to spelling mistakes on name field of the organization units either in *.dbf file or database itself, or by storing wrong code for the organization units. Therefore, before importing the data, it is better if we allow users to check for the correct parents and allow them to choose the correct one from the list of possible parents. The list of possible parents for an organization unit can be fetched from the database with the help of level of hierarchy provided by the users.

4.3.3 Implementation of Beta version

Figure 4.13 shows an implementation design of beta version of the app. There are three main processes as in alpha version but we add new functionalities explained in subsection 4.3.2 in the import process. Projection transformation process is similar as in the existing DHIS2 system

and the process can be referred in section 2.3.3. The simplifying coordinates process is similar as in alpha version and it can be referred in section 4.2.3.

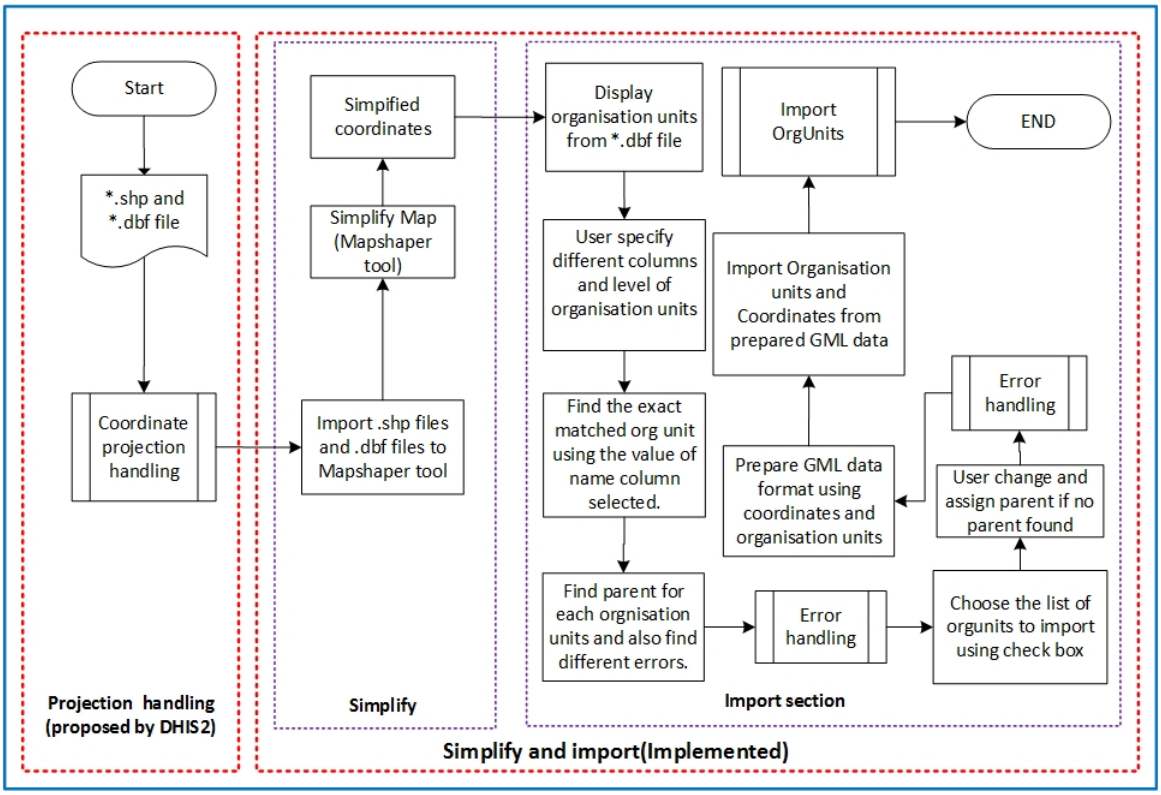


Figure 4.13: Implementation Design of Beta version of the app

In import process, all the organization units that are uploaded from shapefiles are displayed. These organization units are internally connected with their corresponding (simplified) coordinates. Before importing organization units and their coordinates into the DHIS2 system, users provide hierarchy level of the uploaded shapefiles and map the correct database field name with the column field of the *.dbf file. Based on the user's defined inputs, organization units and their parents are fetched from the database. Users are provided with a option to check and change parent organization units before importing the data. If user choose this option, all the fetched parent organization units for different organization units are listed. They can choose the correct one if the app fetches wrong parent. This process can be repeated for several time until the user gets the correct data. After properly checking all the data, all the information should be converted into GML file format. The app prepares GML file format that includes all the mapped organization units information along with their (simplified) coordinates. The app then connects to the Web API of the DHIS2 system and posts the GML data into it. The Web API on other hand inserts or updates data into the DHIS2 system.

4.4 Tools used in the experiment

Some of the tools are used while developing and debugging the app (both Alpha and Beta version) and they are provided in table 4.3.

SN#	Tools	Purpose
1	Tomcat server from Xampp	To run DHIS2
2	Notepad++	As an editor
3	Chrome/Firefox/ Internet Explorer (version greater than 8) browsers	To test the app and its integration in DHIS2 system
4	JavaScript	To develop the app
5	Firefox firebug	To debug the errors of the scripts

Table 4.3: Tools and scripts used for the implementation and debugging of the app

Tomcat server from Xampp was used to run the DHIS2 system on this computer, Notepad++ was used as an text editor, JavaScript for developing the app, firefox firebug to debug the errors of the JavaScript, and the app was tested through browsers such as Chrome, Firefox, and Internet Explorer (version greater than 8).

Chapter 5

Discussion

This master thesis focuses on solving two research questions: identifying the potential for integrating all the necessary stages of configuring web thematic mapping for public health in resource constrained environment, and identifying the processes to leverage such platforms to make them more generative. We decided to solve these research questions by exploring the DHIS2 platform as DHIS2 can give us the idea about the existing issues that users are facing in importing the coordinates and different solutions users are using to solve them.

In order to solve our first research question, we identified all the necessary steps to configure web thematic mapping for DHIS2. These steps mainly include simplification of coordinates, coordinate projection transformations, GML file generation for organization hierarchy and their coordinates and importing GML file into the DHIS2 platform. Please refer to subsection 2.3.3 for more details about these different steps required for the import functionality of the DHIS2 platform. The current import functionality of DHIS2 is very tedious since all these steps should be followed separately and manually and if users do something wrong in any one of the steps then they might not get proper data and this leads them to repeat the process from where mistake was occurred. These stages of the import process become more difficult when the number of organization units and coordinates to be imported is very large. In addition, very often people need help from the expert level users. These issues are realized more in resource constraint environment where there are no or very less technical people from whom they can get the help. In order to identify the potential for integrating all these manual steps, it was necessary for us to understand different challenges that the DHIS2 users are facing and how they want to resolve these challenges.

DHIS2 has several mailing lists such as user mailing list, developer mailing list, and documenter mailing list [24]. Among these three mailing lists, we had communication between user and developer mailing list users. From their email messages, we also found that the process of importing coordinates for organization units using the GML file has been very difficult and users consider it as a technical problem.

A GIS expert says,“HISP-SA would strongly support dropping the

GML step - or at least to short-cut it with an option to import shapefiles directly. GML adds nothing to the mix; it just makes the process more cumbersome and techie". His comment shows how tedious this process is. Even the users with expert level of knowledge find it difficult to use the current method for importing the coordinates into the DHIS2.

Another DHIS2 expert says, "As for the app, I did not try it and agree with the <first expert> that it is much easier for people to import a shape file directly. However, we do not always (but usually do) have a shapefile to import, so I would not be in favor of removing GML support at all". Here, he supports the idea of using the shapefiles for importing the coordinates into the DHIS2, but does not support dropping the GML feature from the DHIS2. There are cases where the GML solution to import the coordinates is useful. Like when the user wants to move the existing organization hierarchy and the coordinates from some legacy system to DHIS2. In these cases, the app that only supports the shapefiles for importing the organizational hierarchy and coordinates can help a very little. With the help of GML feature provided by DHIS2, the users can simply convert the data from the legacy data to GML format and then import them into the DHIS2. This expert also insists to use open source standards. He says, "From an open source point of view we should be using open standards based approaches, like GML and other standards recommended by OGC, instead of locking people into importing geometries with only a single proprietary format."According to him, it is better to follow different standard data format for spatial data that is recommended by OGC standards (http://www.opengeospatial.org/standards/is). That way, users will be able to use multiple formats that they are comfortable with for importing the data into the DHIS2, which I think is a good thing from the user's point of view.

Similar views are shared by a DHIS2 developer. He says, "I'm not really seeing any reason to drop GML support altogether, but the prospect of letting users directly import shapefiles is an interesting one for sure. I'd be very interested to have a look". His suggestion is to directly use the shapefile to import the organizational hierarchy into DHIS2. However, he does not suggest dropping GML support from the DHIS2.

Since the users are finding the import process very difficult, they are also proposing many different alternatives that may be used to make the process easier in DHIS2. According to the DHIS2 developer "GML is fairly complex and I'm not really convinced moving the complexity from established GIS suites like GDAL and QGIS and into DHIS2 itself is a clever move". Here, the DHIS2 developer is saying that it is not good to add all the functionality that are provided by the software like QGIS in the core DHIS2 system. These software has lots of complex features and adding all these functionality can make DHIS2 more complicated. It is better to choose some necessary features and add them into the system. Apart from that, QGIS is a desktop application and it cannot be integrated with the web application like DHIS2.

A DHIS2 user says "Usually the hardest part of this is to properly match the parents, so that the OU tree displays correctly, in your data file, do you

also have the parent OU name/code or something that can be matched on?"According to him, the process of matching the individual organization units (OU) with its parent is the hardest part and wants something easier that can help to do so.

After reviewing different comments to check how the things were done, I did my own observations on DHIS2 especially for the import functionality. I came to know about different aspects of enabling thematic mapping for HIMS in resource constrained environments, with DHIS2 as a case. I understood the importance, issues, and tools that can help to resolve our research problems. Together with some of the tools and file formats proposed by DHIS2 to import organization hierarchy and coordinates [14], I also came to know about different open source and commercial tools that can be used for simplifying the coordinates and convert the projection of shapefiles. For example, Arcgis [17], which is a commercial tool for different GIS related purposes and proj4js [60], which is an open source for transforming the projections of the coordinates. In addition, I also observed some issues in DHIS2 import functionality such as difficulty in converting the coordinate projection, converting shapefiles to GML file, matching the parent child relation for organization hierarchy, managing the duplicate organization units etc.

After configuring fresh DHIS2, the first thing the user is required to enter is the organizational hierarchy. If user intends to use the GIS module that is integrated with the DHIS2, then user needs both organizational hierarchy and coordinates of all the organization units and levels of hierarchy. Since users are finding it difficult to use different tools and create GML file to import organizational hierarchy proposed by DHIS2 and to import their coordinates for jump-starting the DHIS2, we saw a need for a method that will help users to import the coordinates into DHIS2 without any help from any expert level users in resource constraint environment. After analyzing the comments and suggestions from the users about the import functionality of DHIS2 and from the analysis about different tools used in different steps of import process, we saw the potential to integrate all these steps into one solution but it was necessary to study in detail if we can integrate different tools into one solution or not.

Our initial decision for the solution was to add all the basic functions that are required to import the organization units and their coordinates into the DHIS2 system. For this, we would be making the app that can change the projections of coordinate system, simplify the coordinates, link each organization unit with the correct parent, prepare the GML format, and import the organization units and coordinates using the Web API provided by the DHIS2. Some features of the app like transformation of coordinates from one projection to another, finding the projection system of the shapefiles were very complex to be implemented in a short period. Therefore, the next best solution was to use the already available open source solutions, meaning that they are freely available. The next step was to test the tools for the coordinate projections transformation and simplification of coordinates. The test included the tools that were already being used by the DHIS2 users for these purposes. For simplification of

coordinates, the DHIS2 system users were using the Mapshaper. We found that this tool can be integrated in the app and it can be used for uploading and processing the shapefiles. Moreover, the Mapshaper can be used for other purposes like removing trimming the coordinates, checking different irregularity in the data like removing the unsupported characters, making the GML format for importing the data etc. The FWtools software, which is proposed by DHIS2 for coordinate transformation is a desktop-based software. Therefore, it cannot be integrated into the web app that we were planning to develop. Therefore, other alternatives were required for this purpose. We found couple of tools that can be integrated with the web app for the coordinate projection transformations such as ArcGIS, OpenLayers, and Pro4JS. We decided to go ahead with pro4JS, since OpenLayers used the Pro4JS for the transformation and ArcGIS was not an open source tool. Also, the Pro4JS provided the Web API that can be integrated with the tool that we were planning to develop to convert the coordinate projection.

Our second research question ask to leverage the platform itself to make it more generative. To solve this, we thought of using a generative platform which allows users to produce new content without doing any change in the core system. Since DHIS2 is a health information management system and also a generative platform, this is where DHIS2 comes into place. DHIS2 platform has its own Web API which allows users to develop new apps and integrate them together without changing its core system. This enables users to be more creative and add more things that are more customized according to his/her needs. Therefore, using the Web API provided by DHIS2, we decided to develop an app and make its import functionality more simpler than the existing one. We used bootstrapping approach for the development of the app, since this concept guides us to start development of the app with small requirements and then move ahead after getting the feedback and add more complex functionality. We also started by making a simple solution first and then by allowing users to use the solution and getting feedback from them. Then we gradually kept on adding the complex features. It would have been very difficult for us if we had to do everything at once without getting feedbacks from the users and without following the concept of bootstrapping. Actually, there were two ways to develop this tool: either we could modify the core system or we could use the Web API provided by the DHIS2 platform. Since the second option helps developer to add new features without modifying the core system, we chose the second option i.e. to use the DHIS2 as a platform (please check subsection 2.2.3). This option was flexible enough and easier to develop the tool that we intend to.

We first started work for alpha version of the app to integrated all the tools that were necessary to import the organization units and their coordinates into the DHIS2 system. This is the first version of the app, developed to test and check how all the tools can be integrated. It also helps to perform all the steps necessary to be taken before user can import the data into the DHIS2. We added necessary controls on top of user interface provided by the Mapshaper itself to help the users to import the organization units and coordinates into the DHIS2 system. A link between

shape file and dbf file was also identified, which is required to make the GML format while importing the organization units. After finding this linkage, the alpha version could read data from Mapshaper and prepare the GML format. The next task was to send these data prepared to the DHIS2 system for import and this has been done by using the Web API of DHIS2. Successful integration of DHIS2 Web API in our app helped to send data from our app to the DHIS2 system and now the data from the shapefiles can be directly imported into the freshly installed DHIS2 system.

Another tool to be integrated with the app was Pro4JS. There were two ways to integrate it. The first was to convert the coordinate projection of all the coordinates in the shapefile to EPSG:4326. Another method was to allow user to simplify the coordinates and then convert them to EPSG:4326. The second option is better because there will be less coordinates after simplification and this would decrease the workload. While working with Pro4JS api, we found that it is not able to find the coordinate projection of the source file. Therefore, users have to find the projection of the source and give input to the api. This complicated our work and due to time constraint, after some exploration and testing, we decided to hold the integration of Pro4JS into the app for now. Instead, we decided to convert the coordinate projection of shapefile to EPSG:4326 and then import them to the app to be simplified and imported into the DHIS2. For this, users can follow the same process that is described in the DHIS2 user guide using FWtools [14].

After some modification and testing with different shapefiles, the alpha version was ready to be tested with some real data. We got some feedback from the people who used the alpha version to import the organization units into DHIS2. According to a DHIS2 expert, the alpha version did not work the way he wanted. He says, "nothing wrong at all with importing the shape files however, but the last time I tried that client Knut, it did not work for me (or at least the Work flow which I needed)". According to him, what he needed was a process by which he can update the coordinates of already existing organization units. This feature was yet not available in the alpha version.

A DHIS2 user in Srilanka says, "I already have some org units setup on server (upto district and few deeper). Will this app clash with existing org units I installed a new instance of DHIS2 to just test the app. once I upload the .shp file and .dbf the resulting table shows only upto 10 rows is it normal. and selecting id and parent id from provinces .shp is a bit confusing". In this case, one thing that user is complaining is the user interface not being clear. Another issue that he raised was, if the app that I was developing would be able to import the new data into the DHIS2 system that already has existing organization units. This feedback helped me to know the necessity of having more user-friendly interface for the app and a need to allow users to update the existing organization units in the DHIS2 system.

These feedbacks and my observations became the basis for the development of beta version of this app. For the beta version, it was important to handle different cases of importing data. These cases include allowing the users to change or update the existing organization unit and coordin-

ates, allowing users to choose the organization that they wanted to import, change different information using the user interface and then import, allow users to change the spelling as required in the user interface before import. Alpha version was able to import the organization units for the freshly installed DHIS2 only. Beta version needed to update the existing organization units as well. Therefore, this version needed to map the existing organization units with the new organization units from the shapefiles. In the alpha version, parent matching was done by using the code provided in the organization units. However, there are some cases where it can be more complicated. For example, the code that is stored in the DHIS2 system may not match the code that is available in the shapefiles. In this case, either user has to change all the code in the DHIS2 database or in the shapefile. The other solution is to allow user to choose the parent from the list of parent provided by the app. Therefore, another important feature that beta version needed was to allow users to check the parent automatically linked by the app. If there is no matching parent or wrong parent matching shown in the interface of the app, it should allow users to link the organization units with the correct parent organization unit from the list available. This feature would help the users to link the organization units with the correct parent organization unit. This way, it can help in creating the correct organization hierarchy in the DHIS2.

After modifying the alpha to beta version, the app was tested by a core developer of DHIS2. He also gave some valuable suggestions. Some suggestions that he made are regarding the user interface and the understanding. One issue and the alternative suggestion he mentioned was "What does compare actually do? Perhaps 'Map fields' is a better name?". Another suggestion that he made was "Progress indicator aka spinner would be nice to show the user the ". Sometimes it took little more time to complete the task and there was no way to tell if the app was still working or not. These suggestions were quite good and these feedbacks were also updated in the beta version. His feedbacks helped me to make the app more user friendly and responsive than before. One another issue that he found was about not allowing users to add more than one root element in the DHIS2 from the app. The issue he mentioned was "My import of Norway was successful but it should have been created as a new organization unit but I cannot find it back. I did not set a parent". As soon as he added one root element, it created some conflict with the already existing root element if the root element is different. This problem is not solved yet and it can be considered in the future work.

Both alpha and beta version of the app helped DHIS2 platform to become more generative. According to Zittrain [81], Ease of mastery is one of the important criteria for any system to be generative. Ease of mastery for a system is the easiness to use the system. Before these apps, users had to do all the different tasks manually in order to import the organization hierarchy and their coordinates. Based on my own observation and the experience of the DHIS2 users, import functionality of DHIS2 is very tedious and technical. This process requires the knowledge of experts, which may not be available in resource constrained environments. The

alpha version of the app removed the difficulties of using different tools for simplification of coordinates and for the GML file generation. It worked great for freshly installed DHIS2 system, though some users faced some issues for already existing DHIS2 system. These issues have been considered in the beta version and we have got good response from the users too. One of the user of the beta version says "I did substract the unneeded characters and till now it works great ". DHIS2 is a platform and it is generative in itself since it provides Web API through which different programs can be linked with DHIS2 without touching its core. But our app has made it more generative for the import functionality because this process was not that easy before but now user feel easy to use it and it can be used easily also in resource constrained environment where there may not be any experts to handle the case.

Chapter 6

Conclusion

GIS is a very useful tool in various fields where public health is one of them. Some of its applications in public health includes disease surveillance, environmental health research, and respond to health emergencies. GIS is a critical tool, which can help in quick analysis, decision making, and planning but due to some constraints in developing countries such as limitation of cost, infrastructure, education, and political instability, it could be very difficult to employ GIS. The concept of open source applications is very useful in such environments and the open source should be good enough so that people in resource constrained environment can use them easily without much support from the expert level. Therefore, this thesis focuses on two research questions that can help to configure mapping application for public health even in resource constrained environment. The first research question ask to identify the potential for integrating all the necessary stages to configure web mapping in resource constrained environment and the second research question ask to use a platform itself to make it more generative for the purpose of our first research question.

In order to achieve our goals, we studied a case of DHIS2, which is a health information management system and it has been employed in lot of developing countries as a solution for health information management system. Organization hierarchy and their coordinates are required to jump-start the DHIS2 and there are some series of stages that has to be followed by the users in order to import these data into the system. Since these stages of import functionality are manual, and users need to use different tools, the process is tedious and technical. There is also some level of possibility for human errors. Since there can be lack of experts in resource constrained environment, the import process could be difficult for the users in such environments. Therefore, the potential for the integration of these stages into one solution was identified by studying and testing different tools that are necessary for different stages of import functionality. After knowing that these tools can be integrated into one web solution, we decided to develop an app that can perform all these stages with the help of minimum user interactions.

We chose the bootstrapping concept and started developing the app with a simple design, tested the results by limited number of users, got the

feedback from them and gradually added the more complex functionality. We first developed alpha version of the app where all the necessary tools were integrated to support the import the organization units and their coordinates. This app was developed to configure web mapping for the newly installed DHIS2 system. The basic functions included loading and simplification of coordinates using Mapshaper, mapping the parent organization units using the code provided in the shapefiles, preparing the GML format of data for importing, and importing data into DHIS2 using its Web API. We got some feedbacks for the alpha version and included some of the feedbacks as updates in the beta version of the app. The updated features in the beta version include allowing users to link the parents with the child organization units, and allowing users to update the coordinates of the already existing organization units. The user interface has been also modified to make it easier to be followed by the users. The users were quite satisfied with the results of the apps though there is still some space for future work. The app has helped to make the platform more generative than before. Although DHIS2 is a platform and supports generativity, import functionality was not that easy for the general users. Our app integrated most of the stages in one solution and helped users to easily configure the web mapping. It is useful more in resource constrained environments. Therefore, we can say that our app used a generative platform and made it even more generative.

In conclusion, this thesis has solved both of its research questions and the solution of this thesis could be very much useful to configure web mapping in public health even in resource constrained environments. Moreover, this thesis also shows how the concept of bootstrapping can be used and how a generative platform can be leveraged itself to make it more generative.

Chapter 7

Future work

During this project, I faced different challenges while solving the research problems. Some of the challenges include project time duration, technical issues, communication issues with the users, and limitation of real test cases. I was developing an app for the DHIS2 as a methodology to explore how the technological advancement in software development and GIS help the thematic mapping in public health in developing countries with resource constrained environment. Below are some of the tasks that I could not complete during my master thesis. These also include things that I realized while developing the app. These remaining works and some of my experiences during my master thesis can be considered in future.

- Improving coding approach: While writing different functions for linking Mapshaper, formatting the data, preparing the GML format, linking parents and user interface related functions, I have written all of them in JavaScript functions from the start of the coding. In the market there are many different JavaScript libraries like Angular JS, Knockout JS etc. which I could have used. If I had used these libraries and followed the coding pattern, defined by these libraries, the coding pattern would have been better. However, at the later stage of the coding due to time constrains, I could not convert all these functions to follow any of these approaches. Therefore, converting the current codes to follow any one of these libraries is one thing that can be done in future.

  I have changed and added codes of Mapshaper to get organization units data and use it for importing into DHIS2. Due to this, it is bit hard for general users to use the latest version of Mapshaper easily. If I had not mixed the codes done in Mapshaper with the coding done for different user interface functions of the app, then it would have been very easy for the users to use the latest version of Mapshaper for the same purpose. In future, it is better to separate these codes to make them compatible even with new versions of Mapshaper.

- Need of more testing on the app: Like I have mentioned above, I was not able to check the issues the users face at firsthand, making me not exposed to full extent of issues and problems that users face.

That means there might be some cases which are either solved by the implementer by their own way like correcting the spelling of organization units or issues that are not found yet. So in future, it is good to experience the issues faced by users at firsthand to know the issues that user face and try to handle that problem by using the app. This way, we could immediately find out whether app could handle such issues or not, and modify the app if it could not handle it.

In addition, regarding the functions that I have created, more users with different level of expertise should test it. Testing of this app by different level of expertise is important because the main goal of my thesis is to find a way so that users in resource constrained environment also can import the organization and GIS data into the system easily. More testing with different level of users can help us to find different issues related with user interface and make it more understandable and easy to use.

- Usability and ease of use: The UI design and procedure of the app seems to be little bit difficult to follow. One of the reasons might be because of less user guide to handle the situation. There are few things to understand and I believe it can be fixed with each new feedback and suggestion from the user and our own observations. The main purpose of this thesis is to understand and give a solution to users to import the organization units and their coordinates into the health management system, mainly in the developing countries where there is issue of resource constraints. The solution should be able to import into the system without or with very less guidance and jump-start the system. In my study, I took the DHIS2 as my case study. So better user interface is the key to make user comfortable and to make them feel easy to import the data into DHIS2.

- Coordinate Transformation: There are many coordinate projection systems available. All the GIS enabled system might be using different coordinate projection. In my case i.e. DHIS2, it only supports EPSG:4326. The coordinate projection of shapefiles depends on the provider, which can vary and may not match what DHIS2 requirement. So, such coordinates other than EPSG:4326 projection needs to be converted to EPSG:4326 before importing them into the DHIS2. DHIS2 has suggested to use FWTools to convert the transformation from any projection to EPSG:4326. My original plan was to integrate an open source transformation tool in the app that will help users to transform the coordinate projection to EPSG:4326. I decided to use an api provided by Proj4JS for this purpose. But since this api cannot identify the coordinate projection of shapefiles, I decided to halt the idea of integrating this tool for now. In future, I can see two ways of integrating the coordinate transformation in the app. One way could be to find a new Web API that can identify the projection in the shapefiles and transform it to EPSG:4326

and another could be to use the Proj4JS and ask users to identify the projection and provide the source coordinate projection in the shapefiles. That way the api provided by Proj4js can transform it to EPSG:4326.

- Topojson format: TopoJSON is a format that stores coordinates geometries like in shapefiles and GeoJson. It is basically an extension of GeoJSON [5]. For any web mapping applications, the file size and the complexity of geometry is very important. Bigger the size and complexity, greater the lag will be while downloading and displaying the maps in the web browser. Simplification of geometry is important to ease the file transfer and to get the interactive maps. GeoJSON is simple and convenient but is not compact enough. GeoJSON also lacks topological information needed for application such as geometry simplification, colouring and cartograms. Therefore, TopoJSON format was proposed to solve the issue of both the size of the file and simplification [20].The TopoJSON format geometries are stitched together from a shared line segments called arcs, and does not represent geometries discretely. TopoJSON files are said to be approximately 80% smaller or more than GeoJSON equivalents when fixed-precision encoding for coordinates are combined [20].

  In case of DHIS2, shapefiles are often used to import organization units and coordinates but shapefiles of countries can be very large. They contain coordinates for detail map, which is not required in web based application and needs to be simplified to reduce the size of the file before using it. In TopoJSON however the size of the files for the same organization hierarchy is very small. After following the steps listed in http://zevross.com/blog/2014/04/22/spatial-data-on-a-diet-tips-for-file-size-reduction-using-topojson/ URL to convert shapefile into TopoJSON, I found that the size of TopoJSON was greatly reduced even without doing any kind of simplification. Since it is necessary to simplify map in DHIS2 to be suitable for resource constrained environment, TopoJSON format is worth testing further in the future for DHIS2. The current version of DHIS2 does not support the TopoJSON format. Since this is entirely new format than the shapefiles, simple changes in app is not enough to support TopoJSON currently. It requires modifying the core of DHIS2 model and api first.

Bibliography

[1] http://www.esri.com/software/arcgis/arcgisengine. [Accessed: march, 2015].

[2] http://www.geotools.org/. [Accessed: march, 2015].

[3] http://sharpmap.codeplex.com/. [Accessed: march, 2015].

[4] http://fwtools.maptools.org/. [Accessed: march, 2015].

[5] https://github.com/mbostock/topojson/wiki/Introduction. [Accessed: july, 2015].

[6] Arcgis shapefile desc. http://help.arcgis.com/en/arcgisdesktop/10.0/help/index.html#/What_is_ [Accessed: March 2015].

[7] Dhis deployments. https://www.dhis2.org/deployments. [Accessed: March 2015].

[8] Geography markup language (gml) - extended schemas and encoding rules. http://www.opengis.net/spec/GML/3.3. [Accessed: March 2015].

[9] Gml open geospatial. http://www.opengeospatial.org/standards/gml. [Accessed: March 2015].

[10] Install apps in dhis 2. https://www.dhis2.org/doc/snapshot/en/developer/html/ch02s03.html. [Accessed: March 2015].

[11] Shapefile extensions. http://help.arcgis.com/en/arcgisdesktop/10.0/help/index.html#/Shapefile_ [Accessed: March 2015].

[12] Esri shapefile technical description. http://www.esri.com/library/whitepapers/pdfs/shapefile.pdf July 1998. [Accessed: March 2015].

[13] DHIS 2. Chapter 1. what is dhis 2? http://www.dhis2.org/doc/snapshot/en/user/html/ch01.html, 2013. [Accessed November, 2013].

[14] DHIS 2. Chapter 18.2. importing coordinates. http://www.dhis2.org/doc/snapshot/en/user/html/ch18s02.html, 2013. [Accessed November, 2013].

[15] DHIS 2. Chapter 20. import and export. https://www.dhis2.org/doc/snapshot/en/user/html/ch20.html, 2014. [Accessed January, 2014].

[16] Margunn Aanestad and Tina Blegind Jensen. Building nation-wide information infrastructures in healthcare through modular implementation strategies. *Journal of Strategic Information Systems*, page 161?76, 2011.

[17] ArcGIS. http://www.esri.com/software/arcgis.

[18] Geoscience Australia. Coordinate reference systems in quantum gis. http://www.ga.gov.au/webtemp/image_cache/GA20953.pdf, 2014. [Accessed January, 2014].

[19] M. Beer, M.C. Meier, B. Mosig, and F. Probst. A prototype for information-dense it project risk reporting: An action design research approach. In *System Sciences (HICSS), 2014 47th Hawaii International Conference on*, pages 3657–3666, Jan 2014.

[20] Michael Bostock and Jason Davies. Code as cartography. *Cartographic Journal, The*, pages 129–135, May 2013.

[21] Christina Yip Chung, Raymond Lieu, Jinhui Liu, Alpha Luk, Jianchang Mao, and Prabhakar Raghavan. Thematic mapping - from unstructured documents to taxonomies. In *Proceedings of the Eleventh International Conference on Information and Knowledge Management*, pages 608–610, New York, NY, USA, 2002. ACM.

[22] Robert M. Davison, Maris G. Martinsons, and Ned Kock. Principles of canonical action research. *Information Systems Journal*, 14:65–86, 2004.

[23] DHIS2. https://www.dhis2.org/doc/snapshot/en/developer/dhis2_developer_manual. [Accessed: march, 2015].

[24] DHIS2. https://www.dhis2.org/contact.

[25] DHIS2. Chapter 18. dhis as a platform. https://www.dhis2.org/doc/snapshot/en/implementer/html/ch18.html. [Accessed: march, 2015].

[26] DHIS2. Chapter 30. web api. https://www.dhis2.org/doc/snapshot/en/user/html/ch30.html. [Accessed: march, 2015].

[27] DHIS2. Thematic mapping. https://www.dhis2.org/doc/snapshot/en/user/html/ch17s02.html.

[28] Li Dong and Jin Shangjie. Website construction based on web2.0 technology. In *Computer Science and Society (ISCCS), 2011 International Symposium on*, pages 3–6, July 2011.

[29] Molang Dong, Weiji Su, Lin Zhang, and Xiuming Shan. The concept of modularity design in software radio. In *Wireless Communications, Networking and Mobile Computing, 2008. WiCOM '08. 4th International Conference on*, pages 1–3, Oct 2008.

[30] Joseph F. Dracup. Noaa history, geodetic surveying. http://www.history.noaa.gov/stories_tales/geod1.html, 1940. [Accessed January, 2015].

[31] S.M. Elaluf-Calderwood, B.D. Eaton, C. Sorensen, and Y. Yoo. Control as a strategy for the development of generativity in business models for mobile platforms. In *Intelligence in Next Generation Networks (ICIN), 2011 15th International Conference on*, pages 271–276, Oct 2011.

[32] ESRI. http://www.esri.com/. [Accessed: November, 2013].

[33] ESRI. http://support.esri.com/en/knowledgebase/techarticles/detail/29129. [Accessed: march, 2015].

[34] ESRI. http://webhelp.esri.com/arcgisdesktop/9.3/index.cfm?id=1658&pid=1651&topicname=Sim [Accessed: march, 2015].

[35] ESRI. http://support.esri.com/es/knowledgebase/techarticles/detail/24893. [Accessed: march, 2015].

[36] ESRI. Esri map projection. http://support.esri.com/en/knowledgebase/GISDictionary/term/projection, 1998. [Accessed January, 2015].

[37] Joy Frechtling. *The 2002 User-Friendly Handbook for Project Evaluation*. National Science Foundation, Directorate for Education and Human Resources, 2002.

[38] GADM. Global administrative areas. http://gadm.org/, 2013. [Accessed November, 2013].

[39] GitHub. https://github.com/mbloch/mapshaper. [Accessed: march, 2015].

[40] Shirley Gregor. The nature of theory in information systems. *MIS Q.*, 30(3):611–642, September 2006.

[41] Egon G Guba and Yvonna S Lincoln. *Effective evaluation*. San Francisco : Jossey-Bass Publishers, 1st ed edition, 1981.

[42] Lars Gunnar and Trond Andresen. Application of open source gis in district health information system. Master's thesis, University of Oslo: Department of Informatics, 2005.

[43] O. Hanseth and M. Aanestad. Design as bootstrapping. on the evolution of ict networks in health care. *Methods of Information in Medicine*, 42(4):385–391, May 2003.

[44] Ole Hanseth and Kalle Lyytinen. Design theory for dynamic complexity in information infrastructures: the case of building internet. *JIT*, 25(1):1–19, 2010.

[45] M. Harrower and M. Bloch. Mapshaper.org: a map generalization web service. *Computer Graphics and Applications, IEEE*, 26(4):22–27, 2006.

[46] Alan R. Hevner, Salvatore T. March, Jinsoo Park, and Sudha Ram. Design science in information systems research. *MIS Q.*, 28(1):75–105, March 2004.

[47] Michael Himsolt. Gml: A portable graph file format. http://www.fim.uni-passau.de/index.php?id=17297&L=1. [Accessed January, 2014].

[48] Otto Huisman and Rolf A. de By, editors. *Principles of Geographic Information Systems: an Introductory Textbook*. The International Institute for Geo-Information Science and Earth Observation (ITC), 2009.

[49] S. Huynh and Yuanfang Cai. An evolutionary approach to software modularity analysis. In *Assessment of Contemporary Modularization Techniques, 2007. ICSE Workshops ACoM '07. First International Workshop on*, pages 6–6, May 2007.

[50] Lili Jiang, Qingwen Qi, and An Zhang. The thematic mapping system on internet. In *Geoinformatics, 2010 18th International Conference on*, pages 1–4, June 2010.

[51] Martin Kretzer and Alexander Maedche. Generativity of business intelligence platforms : A research agenda guided by lessons from shadow it. *MKWI 2014 Proceedings*, 2014.

[52] Naresh Kumar. Changing geographic access to and locational efficiency of health services in two indian districts between 1981 and 1996. *Social Science & Medicine*, 58(10):2045 – 2067, 2004.

[53] D. H. Maling. Coordinate systems and map projections for gis. http://www.spatial.maine.edu/~beard/Lectures510/Projections%2004.pdf, 2004. [Accessed January, 2015].

[54] mapshaper. mapshaper. http://mapshaper.org/, 2013. [Accessed November, 2013].

[55] Salvatore T. March and Gerald F. Smith. Design and natural science research on information technology. *Decis. Support Syst.*, 15(4):251–266, December 1995.

[56] Brain E. Mennecke and Jr. Lawrence A. West. Geographic information systems in developing countries: Issues in data collection, implementation and management. *Journal of Global Information Management*, 9(4), 2001.

[57] Luciana D'Adderio Neil Pollock, Robin Williams. Global software and its provenance: Generification work in the production of organizational software packages. *Social Studies of Science*, 32(2):254–280, 2007.

[58] International Association of Oils & Gas Producers. Epsg geodetic parameter registry. http://www.epsg-registry.org/. [Accessed: march, 2015].

[59] OpenLayers. http://dev.openlayers.org/docs/files/OpenLayers/Projection-js.html. [Accessed: march, 2015].

[60] World Health Organization. Introduction to the healthmapper. http://www.who.int/hac/techguidance/training/predeployment/Health%20Mapper.pdf. [Accessed: march 2015].

[61] PostGIS. http://postgis.net/. [Accessed: March 2015].

[62] proj4js. http://proj4js.org/. [Accessed: march, 2015].

[63] Mullner RM, Chung K, Croke KG, and Mensah EK. Geographic information systems in public health and medicine. *Journal of medical systems*, 28(3):215–21, june 2004.

[64] David J. Rogers and Sarah E. Randolph. Studying the global distribution of infectious diseases using gis and rs. *Nature Reviews Microbiology*, 1(3):231–237, 2003.

[65] Bjørn Sandvik. Using kml for thematic mapping. http://thematicmapping.org/downloads/Using_KML_for_Thematic_Mapping.pdf, August 2008.

[66] Maung K. Sein, Ola Henfridsson, Sandeep Purao, Matti Rossi, and Rikard Lindgren. Action design research. *MIS Q.*, 35(1):37–56, March 2011.

[67] Neil G Sipe and Pat Dale. Challenges in using geographic information systems (gis) to understand and control malaria in indonesia. *Malaria*, 2(36), 2003.

[68] Espen Skorve and Margunn Aanestad. Bootstrapping revisited: Opening the black box of organizational implementation. *First Scandinavian Conference on Information Systems, SCIS 2010*, LNBIP 60:111?26, 2010.

[69] Knut Staring. *Organizational open source in the Global South: Scaffolding implementation based distributed development*. PhD thesis, University of Oslo, 2011.

[70] United states census bureau. 2013 tiger/line shapefiles. http://www.census.gov/geo/maps-data/data/pdfs/tiger/tgrshp2013/TGRSHP2013_TechDoc_Ch3.pdf, 2013. [Accessed December, 2013].

[71] David Tilson, Kalle Lyytinen, and Carsten Sørensen. Research commentary—digital infrastructures: The missing is research agenda. *Info. Sys. Research*, 21(4):748–759, December 2010.

[72] Amrit Tiwana, Benn Konsynski, and Ashley A. Bush. Platform evolution: Coevolution of platform architecture, governance, and environmental dynamics. *Information Systems Research*, 21(4):675 – 687, 2010.

[73] UiO. The process of developing the dhis. https://www.mn.uio.no/ifi/english/research/networks/hisp/hisp-history.html. [Accessed: march, 2015].

[74] Nisha V, Gad SS, Selvapandian D, Suganya V, Rajagopal V, Suganti P, Balraj V, and Devasundaram J. Geographical information system (gis) in investigation of an outbreak. *The journal of communicable diseases*, 37(1):39–43, Mar 2005.

[75] Jan Henrik Øverland. An open source approach to improving gis implementations in developing countries. Master's thesis, University of Oslo: Department of Informatics, 2010.

[76] Charles-Edward Amory Winslow. The untilled fields of public health. *Science*, 51(1306):23–33, 1920.

[77] Liutong Xu, Xiaoting Fu, Hao Lin, Yupeng Xiao, and Juan Yang. Thematic maps service for gis based urban pipe network visualization system. In *Information Technology and Applications, 2009. IFITA '09. International Forum on*, volume 1, pages 131–134, May 2009.

[78] Youngjin Yoo. The tables have turned: How can the information systems field contribute to technology and innovation management research? *J. AIS*, 14(5), 2013.

[79] F.B. Zhan, Yongmei Lu, A. Giordano, and E.J. Hanford. Geographic information system (gis) as a tool for disease surveillance and environmental health research. In *Services Systems and Services Management, 2005. Proceedings of ICSSSM '05. 2005 International Conference on*, volume 2, pages 1465–1470 Vol. 2, June 2005.

[80] Zhang Zhen, Jin Jing-min, and Liu Fang. The application of geographic information system (gis) in the field of public health. In *Geoscience and Remote Sensing (IITA-GRS), 2010 Second IITA International Conference on*, volume 2, pages 442–445, Aug 2010.

[81] Jonathan Zittrain. The generative internet. *Harvard Law Review (119)*, pages 1974 – 2040, 2006.

Chapter 8

Appendix

8.1 Implementation and User Manual for the app

This section explains the implementation of an app, specifically designed for the DHIS2 system, in order to import the organizational units for a particular level of a country with their simplified coordinates.The app uses Mapshaper tool and adds two features into it so that it becomes useful for the DHIS2. The two features added in the Mapshaper are: i) listing the organizational units for a particular level and associating them to the database structure of DHIS2, and ii) importing the simplified coordinates and organisation units of that level and storing them in the DHIS2. With simplified coordinates, maps can be loaded quickly even in the resource constraint environment.

This app provides these two features by implementing two graphical user interfaces (GUI) within "Org List" and "Show Map" menus in the toolbar of the Mapshaper tool.This section is further divided into four subsections: first subsection explains how the organizational units related files can be uploaded in the Mapshaper tool, second subsection explains how the menus can be added in the Mapshaper toolbar, third subsection provides detail about the implementation and functionalities of the GUI for "Org List" and the third subsection explains about the "Show Map" functionality.

8.1.1 Upload files for organizational units and their coordinates

This section describes how the *.dbf and *.shp files can be uploaded in the Mapshaper. There are two ways to upload files in Mapshaper: one by dragging and moving the file (figure 8.1) and another by clicking the select button (figure 8.2).The Mapshaper tool already provides these functionalities.

8.1.2 Adding menus in Mapshaper toolbar

This app adds two new menus "Org List" and "Show Map" which are highlighted with yellow color at top right side of the Mapshaper toolbar in

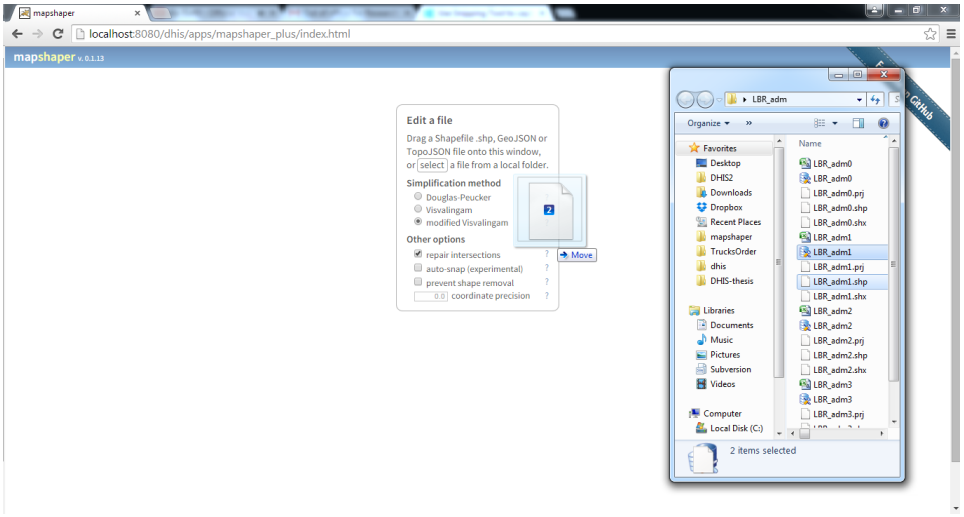


Figure 8.1: Select and Drag upload

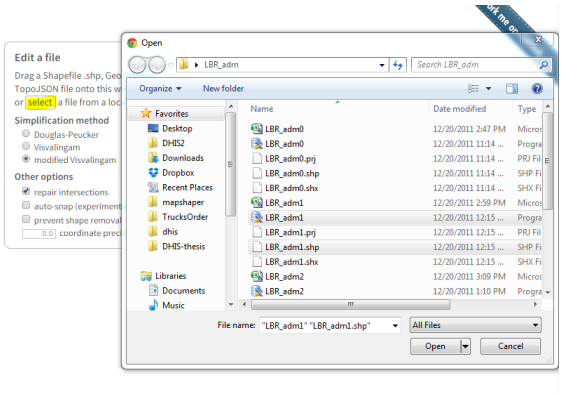


Figure 8.2: Select upload button upload

the figure 8.3. Figure 8.4 shows a Mapshaper toolbar before adding these two menus.Mapshaper shows this toolbar only after uploading the files.

How to add menus in Mapshaper toolbar

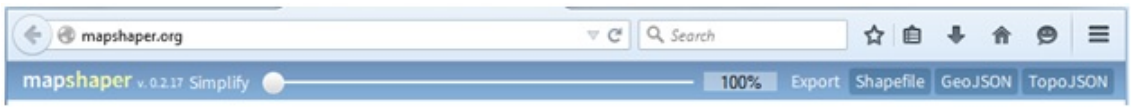


Figure 8.3: Mapshaper tool Original

The "Org List" menu has been added with a purpose to allow users to list the imported organizational units from the *.dbf file and associate them to the organizational unit database table in the DHIS2 so that the new data can be stored in a structured way in DHIS2 system.

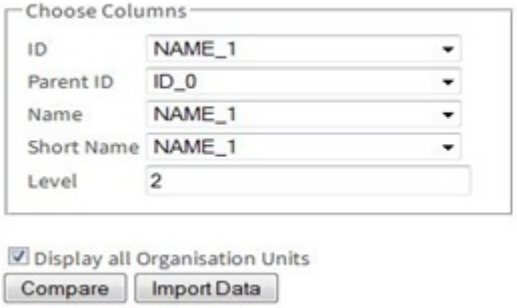
After user click "Org List" menu, the map displayed in the page is

Figure 8.4: Modified Mapshaper tool

hidden

8.1.3 Functionalities of Org List menu

This menu provides a simple GUI as shown in figure for listing the imported organizational units and associating them to the corresponding database fields of the DHIS2 system. We divide this subsection into three smaller parts based on the number of functionalities of this GUI.



Columns	ID_0	ISO	NAME_0	ID_1	NAME_1	NL_NAME_1	VARNAME_1	TYPE_1	ENGTYPE_1
1	127	LBR	Liberia	12	Nimba			County	County
2	127	LBR	Liberia	13	River Cess		Rivercess	County	County
3	127	LBR	Liberia	14	River Gee			County	County
4	127	LBR	Liberia	15	Sinoe			County	County
5	127	LBR	Liberia	1	Bomi			County	County

Figure 8.5: Functionalities of Org List Menu

List the organizational units

One of the feature of this GUI is to list the number of organizational units in a similar way as it appears in the uploaded *.dbf file.The tabular structure in the figure8.5 shows the list of organizational units data for a particular level once the *.dbf file is uploaded. The column names are the same as in *.dbf file and maximum 10 rows from the file are listed at first (if there are organizational units more than 10) in order to show the users what data and columns are there in the file.

Associate imported data with DHIS2 database

This GUI also helps to link the column fields of *.dbf file with the database fields of DHIS2 so that the selected organizational units can be imported into the DHIS2 system in a proper way. This feature is provided by the "Choose Columns" box in figure 8.5. DHIS2 stores organizational unit data in "organisationunit" table and levels in . Figure 8.6 shows the structure of this table.

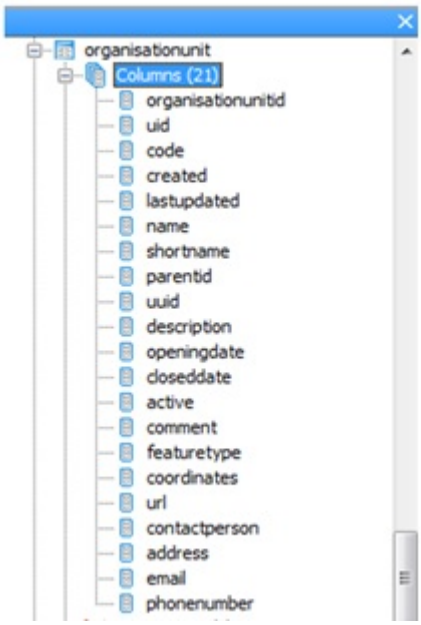


Figure 8.6: Columns Organisation Unit table

Labels such as ID, Parent ID, Name, Short Name, Level on left side of each list box within "Choose column" box of the GUI represents the code, code, name,shortname,.., fields of the database table "organisatoinunit". Each list box lists all the headings available in each column of the *.dbf file. If the file to be imported is for the first level, only two fields (ID and Name) are mandatory otherwise Parent ID field is also mandatory. User who uses this app should be aware of the structure of the *.dbf file and how the data can be linked to the database of the DHIS2 system.The user should select the proper values from the list box based on his/her knowledge; otherwise, there could be mistakes while importing the data into the DHIS2 system. Explanation of each label inside the "Choose Column" box and how the values should be selected for these labels are provided below: 1. ID: This label helps to link the unique values of the organization unit from the *.dbf file into the DHIS2. A care should be taken while selecting a particular column heading, since the selected column values should be unique even with the previous levels of the organization unit. For example in figure 8.7, the first file "LBR_adm0.dbf" contains the organizational unit for level 1, "LBR_adm1.dbf" file for level 2, "LBR_adm2.dbf" file level 3, and "LBR_adm3.dbf" file for level 4.

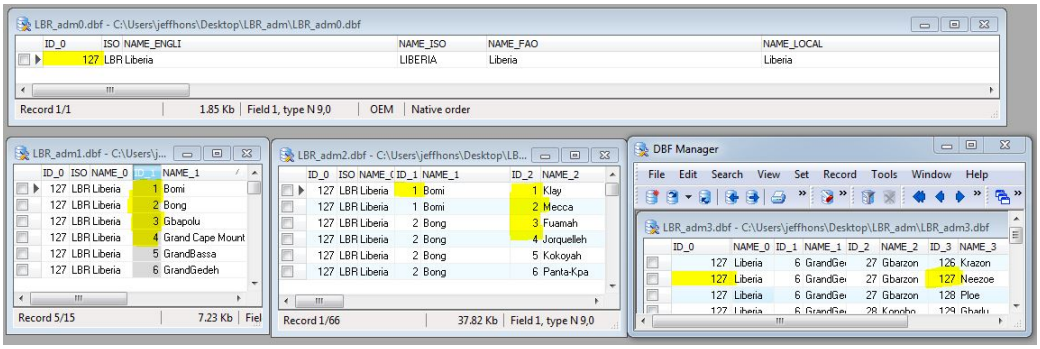


Figure 8.7: Matching ID in dbf files

If we want to import all the organizational units from level 1 to level 4 into the DHIS2 system, we first start with uploading and populating "LBR_adm0.dbf" which can be viewed as in figurefig:MatchingIDindbffiles. Now, we need to select the column with unique values from the populated content to map with the ID field. We can easily see that there is only one organization unit and all the fields are unique. Let's say we selected ID_0 column as its ID. While importing the organization units for level 2 from the "LBR_adm1.dbf" file, we again follow the same procedure and select the appropriate column from the populated content whose values should be unique to the values of the previously mapped column with ID field of the previous levels (in our case ID_0 for level 1). Let's say we want to select ID_1 to link with ID field for level 2. The value of ID_0 (127) is not available in ID_1 column. It means, if we map ID_1 with ID, the values are still unique. Therefore, mapping ID_1 with ID field for second level of organization units could be a valid choice. Now, if we want to import organization units for level 3 into the DHIS2 system, we upload the "LBR_adm2.dbf" file, populate them and select the appropriate unique column to map the ID. If we decide to choose ID_2, the values are not unique with the values of ID_1. ID_2 contains values like 1, 2, 3, 4 which are already available in ID_1. Therefore, we might select NAME_2 in this case, which is unique from the values of NAME_0 and NAME_1 columns.Similarly, if we want to import the organizational units of level 4, we upload the "LBR_adm3.dbf" file and we again need to select the proper column from the file to link with the ID. In this case, ID_3 column values are not unique. It contains 127 value, which is already available in ID_0 column of level 1. Therefore, we may choose NAME_3 column to map the ID in this case. The value mapped with ID field is stored in "code" field of "organisationunit" table of the DHIS2 database.This field has type VARCHAR, which means we can store both number and strings into it. Now, after saving this value in the code field, app can use this value to link a particular organization unit in *.dbf file with its respective organization unit in DHIS2. This "code" field can also be used to find out the parent organization unit's UID and name from the DHIS2 database.

- Parent ID:This label helps to associate the parent organization unit to

all the organizational units of a *.dbf file which will be further used to find a particular parent UID in the DHIS2 database. The selection of equivalent parent id from the populated content is quite straight forward; the column name, which was selected for ID in previous level, will be the Parent ID
for this level. For example, by considering the example explained for mapping ID to the column of populated organization hierarchy, the user should select the Parent ID for level 2 as ID_0, ID_1 for level 3, and NAME_2 for level 4. Mapping the Parent ID is optional, if the uploaded organization unit is of level 1.

- Name: This helps to link the name for the organization units from the *.dbf for the imported organization unit. This is a mandatory field but not necessarily unique. There are cases where parent organization unit might have the same name as of the child. However, it is of course a good practice to make this unique while importing the data into DHIS2. This helps to remove the confusion.

- Short Name:This helps to link the short name of the organization unit, if this field is available in the *.dbf file. This field is not mandatory since all *.dbf file may not have this column field.

- Level: This is a field to be added by the user and it denotes the level of the current organization unit in the organization hierarchy. This field value is useful while comparing and importing the value into the DHIS2 system.

Display all Organisation Unit

This is a feature provided by a checkbox outside the "Choose Column" box with the label "Display all Organization Units". This checkbox when checked helps to list all the values from *.dbf file in the GUI at once.

Compare the organization units

Importing organization unit's hierarchy is a difficult part. One may not get the desired output or hierarchy, if the organizational units are imported without properly linking with their parents. Once the link between DHIS2 database-table-fields and their corresponding column names from the *.dbf file are established, it is important to compare if there is still any conflict between them or not before importing into the system. This feature has been added with the objective to resolve the conflicts between organization units that are already available in the DHIS2 system and the new organization units that have to be imported. It allows users to change the mapping in case of conflict. This feature compares both the organization units (from DHIS2 system and that from the *.dbf file) and display a table to do modification, based on the mapping done from "Choose Columns" box. For example in figure 8.7, ID is mapped with NAME_2, Parent ID with NAME_1, Name with NAME_2, Short Name

with Name_2 and Level with 3 inside the "Choose Columns" box. After clicking the compare button outside the "Choose Columns" box, a table is displayed with the organization units values mapped with the existing organization units from the DHIS2. Note that the columns are displayed according to the selected column names in "Choose Columns" box.

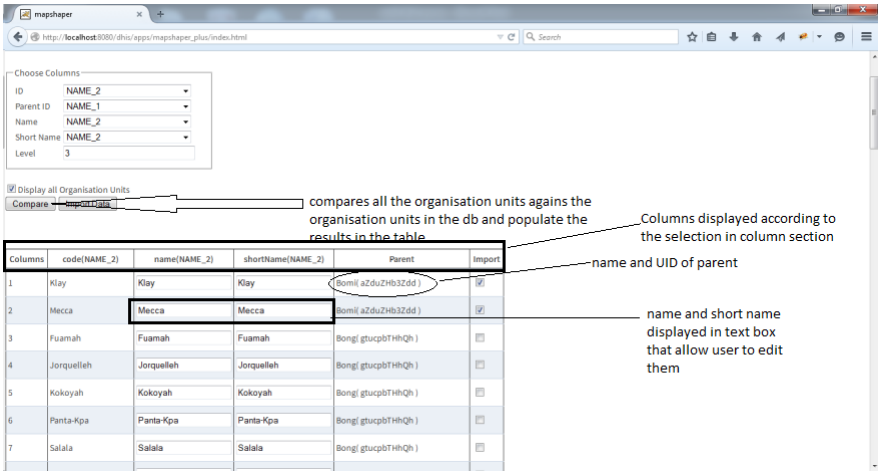


Figure 8.8: User Interface when compare button pressed

There could be conflict on selected column name for mapping the Parent Id from the previous level. If there is no conflict on mapping the Parent Id, the app finds the corresponding parent organization unit from the DHIS2 system and display it (name with the combination of parent name and UID) in the Parent column of the table. This parent name also has a link so that it can be modified if it is not correct. In case of conflict, an error message is displayed stating "select parent" with link to modify it. There could also be conflict on duplicate names for the organization units in parent level or same levels. The names are displayed within the textbox. If user wants to remove the duplicate names, he/she can modify them.

Import Data into the system

This function will prepare the data from the table in the GML format for importing them into the DHIS2 system. If the user has chosen only few organization units (possible from the compare functionality with the help of checkboxes in import column) to import, then only those selected organization units are imported into the database of the DHIS2 system.

8.1.4 Functionalities of Show Map menu

When Org List button is clicked, the map is made hidden. If user needs to review the map, user can use the show map button at the top. This will show the map and hide the tables and other UI fields. Now user can work with the map and check other details here. The coordinates related to organization units can be simplified and updated from here.

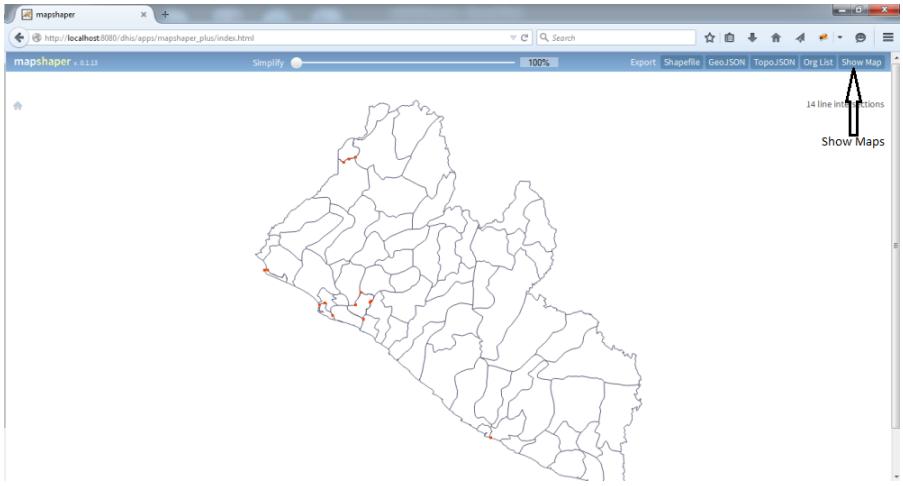


Figure 8.9: Show Map button to display the map

8.2 Setting the app in DHIS2

Installing the app in the DHIS2 system can be explained stepwise as below. For more detail please refer to [10].

i. Open the "App Management" page by clicking the "Apps" link from the top right corner of the DHIS2 system. Figure 8.10 displays this option.

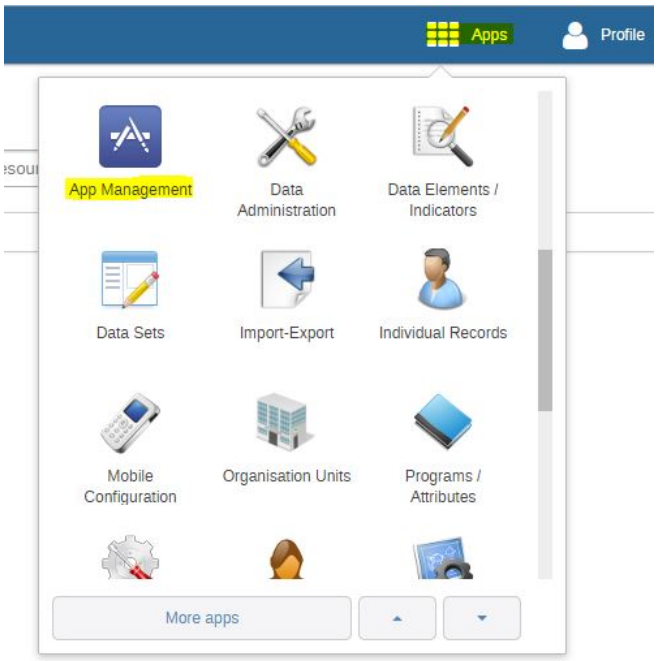


Figure 8.10: Open App Management Page

ii.The user needs to configure two options for the app before installing the app into the DHIS2 system. These two options are setting the "App installation folder" where the uploaded app will be stored, and another

is the "App base URL" which represents a location of where the app can be found on the web.These two options can be changed by clicking the "Settings" link at right hand side of the "App Management" page. Figure 8.11 shows this option.

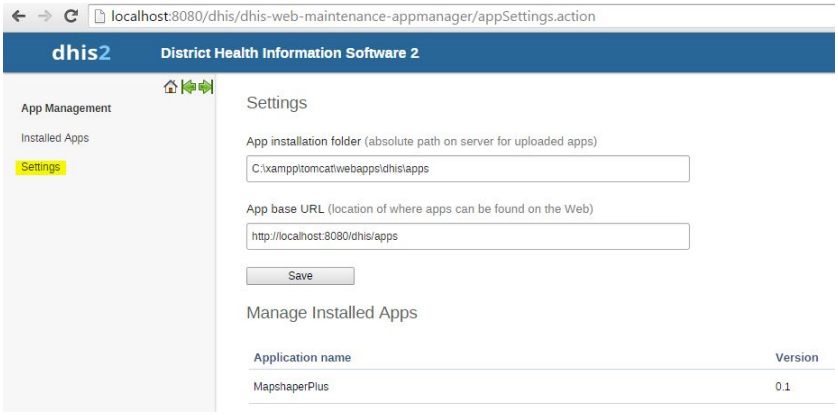


Figure 8.11: Setting path for App installation folder and App base URL

iii. Now the app can be installed by opening the "Installed Apps" link at left hand side of the "App Management" page and uploading the zip file for the app (in our case, the app is provided in a zip file format).File can be uploaded by clicking the "choose file" button on the page as shown in figure 8.12

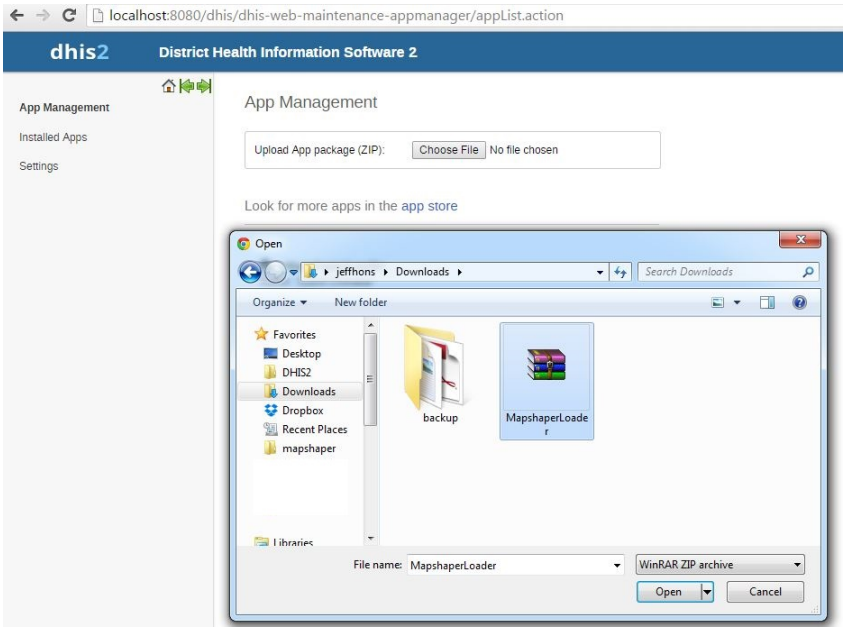


Figure 8.12: Select the zip file of app for installing app in DHIS2

iv. After uploading the correct zip file, the app gets installed and a notification about the installation is provided as in figure 8.13.

v. After few seconds of displaying the successful installation message,

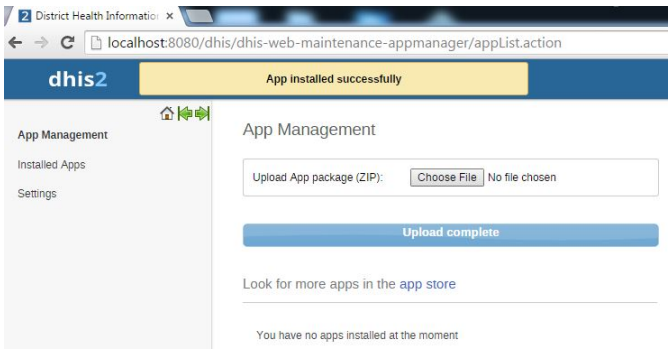


Figure 8.13: Confirmation after app is installed

the user should be able to see the uploaded app "MapshaperLoader" in our case in the same screen 8.14

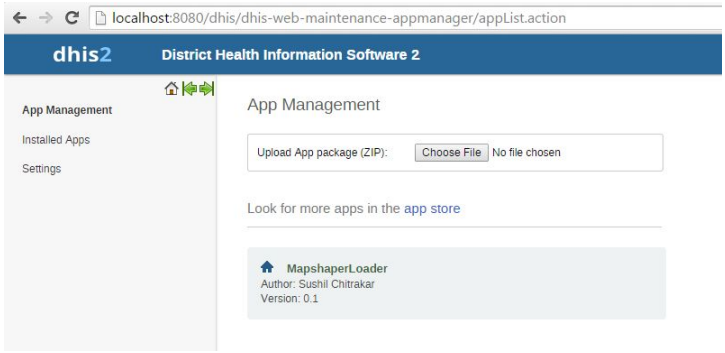


Figure 8.14: Installed app is displayed in the list

vi. In order to open and use the app, click the "Installed Apps" at left hand side of the "App Management" page and click on the installed app. In our case, "MapshaperLoader" has been installed and clicking that app name opens it as in figure 8.15 and user can start to upload the required files and import the organization units and their coordinates into the DHIS2 system.

Figure 8.16is the main page of "MapshaperLoader" app that is running in DHIS2.

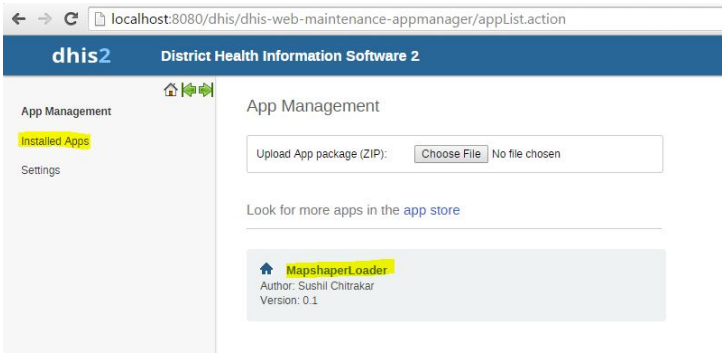


Figure 8.15: Run the MapshaperLoader

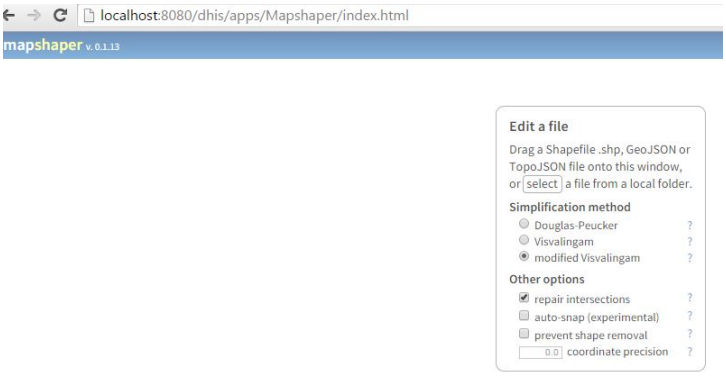


Figure 8.16: Front page of MapshaperLoader