

Saptarshi Purkayastha

The Genus of Information Infrastructures: Architecture, Governance & Praxis

Thesis for the degree of Philosophiae Doctor

Trondheim, April 2015

Norwegian University of Science and Technology
Faculty of Information Technology, Mathematics
and Electrical Engineering
Department of Computer and Information Science



NTNU – Trondheim
Norwegian University of
Science and Technology

NTNU

Norwegian University of Science and Technology

Thesis for the degree of Philosophiae Doctor

Faculty of Information Technology, Mathematics and Electrical Engineering
Department of Computer and Information Science

© Saptarshi Purkayastha

ISBN 978-82-326-0850-8 (printed ver.)

ISBN 978-82-326-0851-5 (electronic ver.)

ISSN 1503-8181

Doctoral theses at NTNU, 2015:97

Printed by NTNU Grafisk senter

To, Almighty and the laws of Karma

Abstract

The complexities in large-scale IT solutions have been acknowledged as a challenge. These complexities arise from interconnections between socio-technical components of what has been referred to in literature using multiple terms such as Information Infrastructures (IIs), Digital Infrastructure, e-Infrastructure, cyberinfrastructure etc. Dividing a complex problem into smaller parts is the prevalent strategy to comprehend complexity in a better way. A similar strategy is to use taxonomy or mental classification to better describe a complex observed phenomenon. In this thesis, I suggest a taxonomy to classify activities within an II as - Architecture, Governance and Praxis (AGP). As taxonomies go, the above classification is comparable to the taxonomic rank of *genus* (plural: *genera*), coming from the Latin (*genus*) and Greek (*genos*) meaning descent, family, type, race, stock etc. I use the large body of existing literature about IIs and use the above taxonomy to classify the activities in IIs. This taxonomy provides clarity to observe II evolution and provides a systematic view to see what activities helped establish the II.

Health Information Systems (HIS) are fragmented and results in duplication of work, ineffective use of resources, incomplete and incorrect information. When data from multiple HIS are brought together, we create an integrated eHealth infrastructure (IeHI) rigged with complexities that are difficult to manage. This thesis covers my participation in the development of two open-source software, DHIS 2 and OpenMRS. These two software systems have communities of software developers, implementers, consultants working at different implementation sites, who come together to establish an II for health. This thesis includes a collection of 6 papers. I start with data collection for HIS using mobile phones (mHealth) and design strategies for scalable mHealth solutions. Then I suggest ways to understand “success” of mHealth implementations. Realizing that design of successful mHealth solutions needs user participation, I suggest OpenScrum, a software development methodology to enable community participation. I then highlight that there is inherent security risk in such contextual software development. Once data comes from multiple sources, it brings forth big data challenges and often data use becomes a problem. We observe that appropriate analytics tools help data use. In the last paper, I highlight that new business models using cloud computing are required to be able to sustain analytics in low-resource IeHI.

I observe that the interplay of the earlier mentioned three genera of activities (AGP) results in what is described in literature as *infrastructuring* i.e. work that is done in conceptualizing, designing, developing, using, and scaling an II. The AGP taxonomy allows observing the interplay of activities within an IeHI at the micro-level and associate them to larger re-combinations at the macro-level. The activities in one genus has effects in other genera. Yet, mapping and quantifying consequences to activities is difficult due to the inherent complexity in II. I call this view of causality, with property of sentience as *Karma*, a term used in Buddhism to describe the interconnected actions and their effects. I use *Karma* to explain that as actors associated with an II, the actors

perceive *effects* of actions as good (stabilizing) or bad (destabilizing), whereas these events have perception of causality that is often a time-limited view of the observer or interpretation of the actor.

The thesis attempts to answer the following research questions:

RQ1: Given attention to the ongoing efforts of developing infrastructures, how can activities in an IeHI be classified using a taxonomy?

RQ2: What are the blind spots created through this taxonomy and how do they affect the infrastructure evolution?

The main contributions are:

C1: A taxonomy to classify II *infrastructuring* activities into Architecture, Governance and Praxis.

C2: Highlight the organizing categories of activities by analysing cause-effect relationship in cases from eHealth implementations.

C3: Articulating Information systems implementation success in terms of meeting local needs.

C4: OpenScrum agile methodology to improve knowledge sharing in OSS communities.

C5: Defining Big Data through Organizational Capabilities that can be leveraged by the use of Operational BI Tools for analytics in IeHIs.

Preface

This thesis is submitted to the Norwegian University of Science and Technology (NTNU) for partial fulfilment of the requirements for the degree of philosophiae doctor.

This doctoral work has been performed at the Department of Computer and Information Science, NTNU, Trondheim, with Eric Monteiro (NTNU) as the main supervisor and with co-supervisors Kristin Braa (UiO) and Hallvard Trætteberg (NTNU).

The work during the PhD was funded through a fellowship awarded by the Norwegian Research Council – VERDIKT programme research project called Global Health eInfrastructure. (#193023)

Acknowledgements

First and foremost I would like to take this opportunity to thank my supervisors Eric Monteiro (NTNU), Kristin Braa (UiO) and Hallvard Trætteberg (NTNU), without whose guidance, support and comments, this thesis would not have been the way it is available today. Eric has been a strong philosophical guide and pillar, on whom I've bounced off many an idea and wonderful discussions on life, living and information systems. Jørn Braa (UiO) has been instrumental in providing his guiding ideology towards Health Information Systems development that is core to my life and work. I would also like to express my sincere gratitude to Sundeep Sahay (UiO) to bring me to this academic community and awaken inclination towards a PhD degree. Kristin has been extremely kind for her support during my early days in Norway and has been more of a friend than only a supervisor.

I would like to thank a number of participants of the HISP network, particularly Terje, Tiwonge, Ime, Knut, Edem, Ranga, Abyot, Ola, Petter, Margunn, Jens, Johan, Arunima, Vasudha, Richa and many others, some of whom have spent many months with me during field work, as well as discussing implementation strategies and paper/thesis ideas. These range from researchers, developers, consultants, managers, administrators, health officials in many DHIS2 implementation countries. I would also like to thank the core developers and implementers of the OpenMRS community for their able guidance and participation to discuss ideas mentioned in this thesis. Naming each one of them on this page is impossible, but I thank them with my deepest gratitude.

Furthermore, the Global Infrastructures research group at Department of Informatics, University of Oslo, with whom I've had close collaboration deserves mention, including faculty, PhD students, as do colleagues and friends in Department of Computer and Information Science, NTNU. I would also like to thank friends and colleagues in India, Malawi, Bangladesh and WHO-SEARO. They have all contributed in essential ways, through these years, be it in academic debates, planning, creating manuals, training health workers, developing and customizing software, or driving through bumpy, clay roads during rainy season in Malawi or being lost in translation in North Korea.

While thinking about bumpy rides, the work during this PhD has been quite bumpy at times. For instance, being thrown away from research sites in India and then complaints made to conferences claiming inappropriate use of data. I hope these happened due to differences in ways of working, rather than hostility. I bring this up only to describe the challenges, but I've apologized, forgiven and forgotten those times. I've had the pleasure to work with and learn from dedicated people all over the world, including flying Calle's plane in Cape Town to late nights at hospital in Shimla to eating Nshima (cornmeal dough) for breakfast, lunch and dinner and venturing into *Malawian Gold* (tobacco) fields.

I would also like to acknowledge the support from family and friends, especially my wife Namrata and my parents, my brother for their patience and continuous encouragement.

Contents

Abstract.....	i
Preface.....	iii
Acknowledgements	iv
Contents	v
List of Figures.....	vii
List of Tables	vii
Abbreviations	viii
Chapter 1: Introduction	9
<i>Problem outline</i>	10
<i>Motivation</i>	11
<i>Research context</i>	12
<i>Research questions and contributions</i>	14
<i>Research design</i>	17
<i>List of papers and contributions</i>	19
<i>Thesis Structure</i>	22
Chapter 2: State of the Art.....	23
<i>Part 1: Theoretical Synthesis</i>	23
<i>Part 2: AGP model of II activities</i>	40
Chapter 3: Context and Research Design.....	50
<i>Research Goal</i>	59
<i>Research Process</i>	60
Study 1: What are some design strategies for scalable mHealth solutions?	60
Study 2: How should researchers evaluate “success” for mHealth implementations?.....	61
Study 3: What software development methodology can be used to increase user participation in open-source communities?	63
Study 4: How do security challenges arise due contextual software development?.....	66
Study 5: How do warehousing and BI tools evolve to make a health system use big data?.....	69
Study 6: What are new cloud computing models that can reduce digital divide in LMICs for analytics in IeHI?	71
Chapter 4: Results	73
<i>Results of the studies</i>	75

Study 1: Strategy of alignment to existing Infrastructure for mHealth apps	75
Study 2: Meeting local needs means “success” in mHealth implementations ..	78
Study 3: The OpenScrum methodology of agile software development improves developer as well as user participation in FOSS communities.....	80
Study 4: Contextual development <i>inscribes</i> security problems in software where contexts of use are insecure	83
Study 5: 3 generations of Operational BI tools and Big Data is dependent on Organizational Capabilities	85
Study 6: Cloud computing can result in sustainable use of analytics tools in healthcare.....	86
Chapter 5: Evaluation and discussion	92
<i>IeHI – in a mode of continuous flux</i>	92
<i>Evaluating the studies using the AGP model</i>	94
<i>Blind spots in the AGP model</i>	104
<i>Reflections on the research context</i>	105
Conclusion	106
References.....	108
Appendix A: Selected papers.....	123
P1: Design & Implementation of mobile-based technology in strengthening health information system:Aligning mHealth solutions to Infrastructures	124
P2: A post-development perspective on mHealth – An implementation initiative in Malawi	147
P3: OpenScrum: Scrum methodology to improve shared understanding in an open-source community.....	155
P4: Towards a contextual insecurity framework: How contextual development leads to security problems in information systems.....	176
P5: Overview, not overwhelm: Framing Big Data using Organizational Capabilities	186
P6: Big Data Analytics for developing countries – Using the Cloud for Operational BI in Health.....	201
Appendix B: Secondary papers	222
<i>SP1: HIXEn: An integration engine for multi-vocabulary health information using REST & semantic metadata mapping</i>	222
Appendix C: PLS output for Paper 5	233

List of Figures

Figure 1: HIE approach to IeHI.....	9
Figure 2: DW approach to IeHI.....	9
Figure 3: Research design - Charting contributions from each study	17
Figure 4: BI framework (Watson & Wixom, 2007)	38
Figure 5: Generative mechanisms (Henfridsson & Bygstad, 2013).....	41
Figure 6: Initial understanding of the AGP Model of II activities	43
Figure 7: Final take on the AGP Model of II activities.....	49
Figure 8: World population to mobile subscribers (ITU, 2011).....	51
Figure 9: Mobile-cellular subscriptions worldwide (end of 2013) (ITU, 2013)	51
Figure 10: Global mobile phone subscriptions (ITU, 2013)	51
Figure 11: Uganda pilotitis (Sean Blaschke) 2011)	53
Figure 12: Placing the studies in AGP model	102

List of Tables

Table 1: Summarizing papers and contributions	20
Table 2: Scaling from standalone applications to Information Infrastructures	24
Table 3: Theoretical background of the concepts.....	30
Table 4: Representing cases in the AGP model	44

Abbreviations

AGP	Architecture, governance and praxis
BI	Business Intelligence
DW	Data warehouse
FOSS	Free and open-source software
HIE	Health Information Exchange
HIS	Health Information System
IeHI	Integrated eHealth Infrastructure
II	Information Infrastructure
IS	Information Systems
IT	Information Technology
LMIC	Low and middle income countries
HISP	Health Information Systems Programme
HMN	Health Metrics Network
ITU	International Telecommunication Union
MDG	Millennium Development Goals
NTNU	Norwegian University of Science and Technology
UiO	University of Oslo
UN	United Nations
WAHO	West African Health Organization
WHO	World Health Organization

Chapter 1: Introduction

Today, health information systems (HIS) range from clinically relevant patient-level data to aggregate national-level indicators, related to quality of health service delivery and health program effectiveness. Research has indicated that these information systems, especially in the context of developing countries, function in their own respective silos both technically and institutionally, and do not necessarily talk to each other (Okunzi & Macrae, 1995; Chilundo & Aanestad 2005; Mengiste, 2010). This lack of collated data from different parts of a country-wide health system results in incomplete and incorrect information (Stansfield et al., 2008). The world has realized the importance of overall improvement of health in developing countries and thus constituted the UN Millennium Development Goals (MDGs) in the year 2000. Out of the eight MDGs, three are related to health. UN member countries reached a consensus to monitor vital health indicators and improve these over time. During the last 13 years, many countries have supposedly made significant progress in improving these indicators. But many countries are also about to miss goals, which were expected to be met by 2015. Countries that have made progress need to continue, which is why a consensus towards Sustainable Development Goals (SDGs) is currently underway. All of this seems excellent. But the data by which countries are generating indicators has been shown by researchers to be incomplete and incorrect (Attaran, 2005; Murray et al., 2007).

Current thinking at both the global (e.g. HMN & WHO, 2008) and national levels (e.g. in India, Rwanda, Philippines) to address the problem of technical and institutional fragmentation is through the design, development and

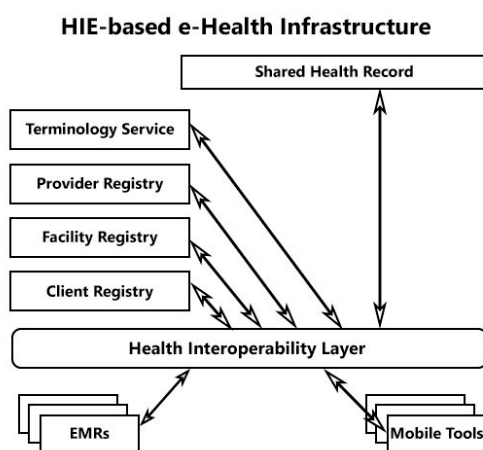


Figure 1: HIE approach to IeHI

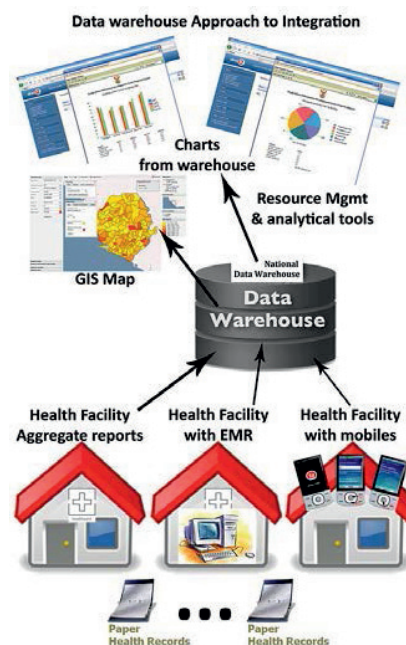


Figure 2: DW approach to IeHI

implementation of country-level integrated eHealth infrastructures (leHI). Sæbø et al. (2011) describe a country-level data warehouse that receives data from multiple sources of patient-level data that is either aggregated manually on paper or through software systems. Figure 2. shows this Data Warehouse (DW) approach of integration where data comes from manual aggregation, electronic aggregation from EMRs or data entry from mobile phones (mHealth) which then allows big data analytics from a central DW. Moodley et al. (2012) describe another approach, which is to create a country-level shared patient record through Health Information Exchanges (HIE). Figure 1. shows the open-source OpenHIE/RHEA (Rwanda Health Enterprise Architecture) initiative that creates an integrated patient-level data store. These are two ongoing approaches in developing countries to create country-level integrated e-Health infrastructure.

Both these approaches can be described as open, heterogeneous, shared, evolving installed bases (Hanseth & Monteiro, 2001) where multiple systems can share electronic health information. In this thesis, I refer to this as integrated eHealth infrastructure (leHI). An leHI is an information infrastructure (II) which deals with all the activities performed in design, development and implementation of systems that support collection, collation, analysis and communication of health information electronically. An leHI involves integration between various components of the health system including members of health staff who work under the guidance of a health systems policy, and use health information systems and applications. Here, “integrated” refers to interchange of information between the components of the health system such that data integration, technology integration and business process integration occurs (Booth et al., 2000). Such integration as will be discussed later in the thesis, is not seamless (Ellingsen & Monteiro, 2006) or without differences in processes across the health system. Monteiro (2003) suggested a more decentralized, multi-vocal approach to integration and this thesis takes similar view to integration when describing leHIs. The health system covered by an leHI might range from a department within a hospital, multiple departments in a single hospital, multiple departments in multiple hospitals, many health facilities in district or even a whole country. An leHI is the coming together of components (people, processes and technology) that enable electronic exchange of health information.

Problem outline

Recognizing the need for co-ordinated efforts in eHealth, the WHO working in collaboration with ITU created the National eHealth Strategy Toolkit (WHO & ITU, 2012). One of the required components used to describe the vision of a healthy nation is

standardization and interoperability between eHealth implementations (ibid, pp 53). Many countries are following this toolkit and creating a vision for nation-wide leHIs.

The two previously mentioned efforts (DW approach and HIE approach) to create leHIs in resource-constraint countries are fairly new. But similar efforts have been going on in resource-rich environments for some time now and have been researched fairly well (Kaushal et al., 2005; Hanseth et al., 2006; Jha et al., 2008; Urowitz, 2008). These researchers seem to suggest that leHI initiatives have not met with critical success. How then do we expect leHI initiatives in low and middle-income countries (LMIC) to be successful? In such contexts due to resource constraints, there is often a debate on whether resources spent on HIS are better spent on medicines, health facility etc. While this thesis does not delve into such discussions, I recognize that to spend resources on medicine, health facility, health worker training etc., one needs to know where need is the greatest and what resources are available. When HIS have complete, accurate and timely information available, one can allocate resources to places where it is truly needed. Yet, information availability should not be automatically considered actionable. This thesis discusses tools that can be used to transform large sets of data into actionable information.

Motivation

The primary motivation for the reader of this thesis is to better comprehend infrastructural complexity and reduce resource wastage in establishing leHIs. As seen from HIE efforts in rich environments, the challenges to establish infrastructures in the digital/electronic world is a known, larger class of problem (Karasti & Baker, 2004; Tilson et al., 2010; Bowker et al., 2010). There are a number of *infrastructuring* (Hughes, 1983, 1989; Scott, 1998) activities that are required to deal with this “class of problems”. My instance of this “class of problems”, discusses specific cases in health infrastructures, which falls within the domain of digital infrastructures. I have looked at the infrastructuring activities through specific projects that contribute to leHIs. Those involved in leHIs and those doing infrastructuring work will see obvious similarities. They should think about using this taxonomy at a more micro-level of project management. Yet, because this infrastructuring work is a challenge in many other types of digital infrastructures that are starting and failing each day, the taxonomy is likely to be useful to a number of cases from the macro-level of organizing participation in II. The thesis and its “opportunistic”, “emergent” research design is unique because it integrates the challenge of alignment (between architecture-governance, governance-praxis, and architecture-praxis) of infrastructuring work, and attempts to present a coherent view of these challenges. While there have been many researchers who have described

alignment and embedding work practices (inscription) in technology, my thesis unpacks the parts of alignment and inscription in more practical and conceptual way (see AGP model in Chapter 2).

To be able to learn from the challenges in leHI implementations, one needs to understand its various aspects with clarity. One needs to comprehend the complexity of leHIs and be able to observe efforts that have led to the evolution of leHIs. This thesis presents a new taxonomy of classification of activities in an II, so that its inherent complexity can be better understood. By dividing the complexity of activities and its effects, we can observe which activities have had a stabilizing or destabilizing effect on the II. Thus the process of evolution of an II can be better studied.

Another motivation for the reader is to gain insight into software engineering practices that bring better alignment between technology artifact and praxis through suggested practices in design and development of leHIs. eHealth systems, along with other ICT4D systems are often designed and developed in high income countries (HICs) due to lack of technology skills in LMICs (Prakash & De', 2007). This disconnect often results in a gap between what is required and what is developed (Heeks, 2002). By suggesting alternative ways to bring together implementers and developers of eHealth systems, this thesis suggests how leHIs can be better aligned to the needs of LMICs.

Research context

In the thesis, I do not study a single leHI being established, but rather the components of eHealth systems that are assembled into an instance of Architecture through Governance principles and implemented practices (praxis). Thus, the objects of study are the various components of an leHI from the level of someone assembling these together.

The research was conducted as part of a fellowship received from the Norwegian Research Council's VERDIKT programme. The project Global Health e-Infrastructures (#193023) received funding from 2009-2014 on which I was employed as a PhD candidate/research fellow/*stipendiat*. The research is part of the HISP (Health Information Systems Programme) network comprising of researchers, developers, implementers, representatives from ministry of health, all of whom share learnings between the nodes of the network. The 18 years' time-span of HISP exceeds traditional 'projects' and is more akin to social movements (Elden and Chrisholm, 1993).

I come from an LMIC myself, having been born and brought up in Mumbai, India. It is a city where I've mingled with the richest of rich of the world and yet worked between

poorest of the poor. Due to my background in computer science, I have over the years contributed to many open-source projects, two of which (DHIS2 and OpenMRS) are core artifacts described in the thesis. All this has given a perspective and bias that I'd like to lay on the table at the start of the thesis. I've been closely involved with the software artifacts and its development teams and thus my participation in the research can be described as that of an "involved researcher" (Walsham, 1995). Most of my research has been conducted through action-research methodology and has been described in thorough detail in Chapter 3. My involvement in eHealth implementations has been continuous, albeit many of which are not part of my doctoral work. In between the doctoral work, I took leave for a year to work on an EMR implementation outside the HISP network in rural India. And towards the end of my PhD for 4 months, as a consultant for WHO on eHealth implementations. I've gained experience in aspects beyond this thesis such as telemedicine systems, hospital information systems etc. in countries like India, Bhutan, Bangladesh and North Korea.

The cases that form part of this thesis are implementations from LMIC such as Malawi, India, Kenya, Bangladesh and development of these technology artefacts in Norway, United States, Uganda and India through a globally distributed team of software developers. The research traces evolution that has taken place when these artefacts were assembled in social context during implementations or with implementers from these countries. The implementers are not disconnected from the contexts of implementations and the developers are similarly connected to the implementation requirements. Thus, the perspective taken in this research is not to separate the context of development and implementation. The artefacts and its development teams are the focus of my research because their actions get inscribed (Hanseth & Monteiro, 1997) in the artefacts that are deployed in the context. Yet, the conceptualization of context is not that "everything is text" (Latour, 1996), but rather specific components that act in their capacities on other components with which they are assembled together to create an leHI. In this vein, there exists a contextual separation between people who develop software (developers) and those who customize software (implementers) and those who use software (users). They do involve their capacities to affect each other and form a relation of "causality" with other components that they interact with.

With the above background, the cases that form part of this thesis are following. The first case involves study of two scalable mHealth solutions in India and Kenya respectively. The case is published as part of a book chapter and focuses on design factors that have allowed scalability of these mHealth solutions. Here the architectural perspective is highlighted, where matching of contextual requirements have resulted in scaling of the mHealth solutions. The second case suggests ways to evaluate "success"

of mHealth implementations. This research was done in Malawi, where the mHealth solution from the first case was implemented with certain modifications. Beyond the scaling criteria for “success” highlighted in the first case, the second case suggests that meeting local needs should be the criteria for evaluating mHealth solutions that are assembled into an infrastructure. The third study presents how local needs can be incorporated into software solutions using a new kind of software development methodology. The new software development methodology is called OpenScrum and is studied within a community of developers which builds the OpenMRS electronic medical records (EMR) platform. This EMR is widely implemented in LMICs and forms part of leHIs in many countries. Importantly, the new software development methodology is studied as an evolutionary process of how local requirements can be better incorporated into software artefacts. The fourth study analyses the security challenges that crop up due to contextual software development. This study looks at the evolutionary process of software development where care needs to be taken to embed local requirements. This study focuses on the need for artefacts to deal with changes that need to be inscribed due to the different requirements of networked and non-networked ways of working. The fifth study looks at the evolutionary process of how data analytics should be performed, once data has been gathered into a data warehouse. This study presents 3 generations of Operational Business Intelligence (BI) tools in DHIS2 data warehouse, which allow users to make use of information during operational activities. This study defines Big Data in terms of organizational capabilities instead of the current definition of Big Data that only uses technological definitions. The sixth and final study looks at the opportunities and challenges presented by leHIs when they are deployed in the cloud. The study examines how cloud computing solutions may provide value in the context of Health IS in developing countries characterised by Big Data situations. The authors draw on the Analytics-as-a Service component of DHIS2 software which is used in over thirty LMICs.

A detailed description of the context and research design is in Chapter 3 of the thesis.

Research questions and contributions

The thesis attempts to answer the following research questions:

RQ1: Given attention to the ongoing efforts of developing infrastructures, how can activities in an leHI be classified using a taxonomy?

RQ2: What are the blind spots created through this taxonomy and how do they affect the infrastructure evolution?

As mentioned in the Problem Outline, that the purpose for the thesis is to be able to better comprehend the complexity of leHI by dividing the activities into smaller parts. Since the 18th century, when Carl Linnaeus first published the classification system for living beings, it has spurred detailed research in each of the subsystems of living organisms. It has brought clarity in studying the characteristics of living organisms. By using biological classification system, researchers have been able to identify physiological similarities and differences between the varied species of living organisms. On the other hand, modern *cladistic* methods of classification based on inferred evolutionary relatedness, have ignored morphological similarities and thus enhanced our understanding of evolution (Laurin, 2010). We see that classification systems have helped in explaining similarities and differences as well as charting evolutionary processes.

The idea that Information Infrastructures are an assemblage of socio-technical components is well researched (Monteiro, 2000; Hanseth et al., 2004; Bygstad, 2010; Henningsson & Hanseth, 2011). Yet, how activities are connected within this assemblage, how is *infrastructuring* work organized and what characteristics of the work activities needs to be studied, has not been described in a coherent way. Few II studies have looked at effects of actions between actors and their relationship within a researcher assembled network. By answering RQ1, we can see how IIs (in this case leHIs) can be established by assembling together different types of activities. RQ1 is also important to answer for improved planning. Segregation of teams and their roles can be better demarcated, once activities are properly classified. I acknowledge that there can be multiple ways to classify activities. This thesis provides arguments for using or not using this classification scheme and the challenges that are faced when attempting to classify actions that are not independent of other actions or have repercussions on other activities. Sometimes there is only a thin line of differentiation. Answering RQ1 helps draw the line between what can sometimes be actions or otherwise effects of actions that in turn become future actions.

RQ2 is an extension of RQ1 in a way that once we have been able to classify activities, we create blind spots that sometimes makes the activities visible or at other times invisible. We want to understand and plan the ways in which causality between these activities affect the evolution of the II. Answering RQ2 can help in planning IIs. It can also help in post-facto studies of II by providing ontology for describing observed phenomenon. In planning phase, it can act as a guide to understand how components will be affected by actions in other components, while it can be used in post-facto studies to pinpoint activities that have had effects on other components. Answering RQ2 can spur *cladistics* in II research, such that evolution of different digital infrastructures

can be theorized better. Tilson et al.(2010) specifically calls for the IS community to focus on evolution of digital infrastructures (I'll call it IIs only) and other researchers for example Henfridsson & Bygstad (2013) and Henningsson & Hedman (2014) have taken cognisance to theorize the evolution of II.

The main contributions of the thesis:

C1: Taxonomy to classify II activities into Architecture, Governance and Praxis

Based on the literature review from the state of the art described in Chapter 2, I have been able to synthesize a genus of activities in establishing an II that can be classified into Architecture, Governance and Praxis categories. Further, the 6 studies that form part of this thesis are demarcated along these categories based on their findings.

C2: Highlight the organizing categories of activities by analysing cause-effect relationship in cases from eHealth implementations

Each of the studies highlights that every work activity results in a different infrastructure from the line that previous work activities were causing. The thesis highlights that the components in themselves do not have inherent capacities. Instead, the components cause effects in other components, depending on the components they are assembled with. The thesis is able to show that, at each level of abstraction, the internal components of a component in leHI can be ignored in the study.

C3: Articulating Information systems implementation success in terms of meeting local needs

In study 2, we have been able to show that *success* of an mHealth implementation is mainly in meeting local needs. Each locale has different needs and meeting those needs should be bigger priority compared to larger geographical coverage, bigger no. of users or other quantitative indicators.

C4: OpenScrum agile methodology to improve knowledge sharing in OSS communities

If meeting local needs can be described as *success* for an IS implementation, developers need to first understand what these needs really mean. Through the study of the OpenMRS community, I've understood that a tweaked agile method can help improve knowledge sharing in open-source communities. This shared knowledge base cannot just improve sustainability of the project, but can also result in better understanding of the implementer or user needs.

C5: Defining Big Data through Organizational Capabilities that can leveraged by the use of Operational BI Tools for analytics in leHIs

Another contribution of the thesis is to define Big Data, beyond the popular terms that only use technological factors. In study 5, we describe Big Data through social factors such as Organizational capabilities. The capabilities of the organization in terms of technology, culture and structure helps define if the organization is able to make sense of the data and convert data into information and future into actions. Contributions C1 and C2 are theoretical in nature, whereas C3, C4 and C5 are practical in nature. In fact, Paper 5 is practitioner focused paper because we wanted to highlight the evolution of the Business Intelligence tools in DHIS2. These tools have been based on user needs and over a period of time are able to deal with big data situations.

Research design

The research design is outlined in Figure 3.

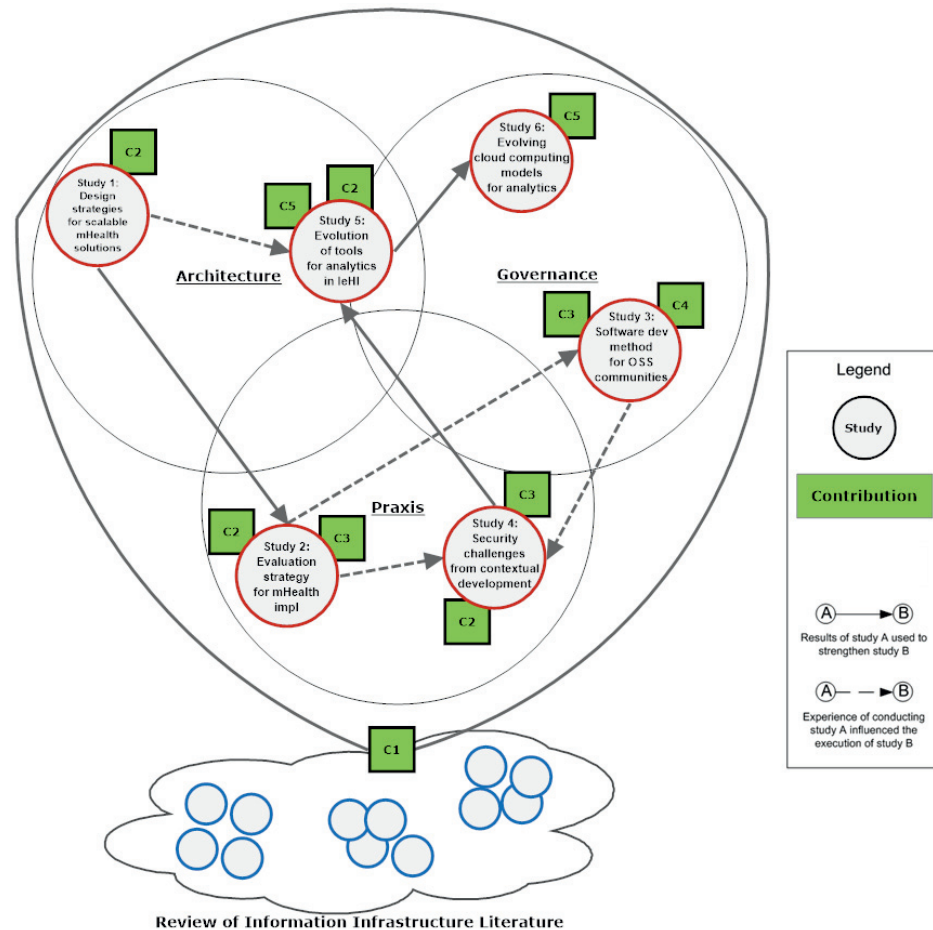


Figure 3: Research design - Charting contributions from each study

The research design shown in Figure 3. has manifested over time of 4 years of the PhD study. The ideas were fluid and were first presented in the GHeI VERDIKT research project under which the fellowship was granted by the Norwegian Research Council. My research was designed in spirit under that project and thus the overarching guiding ideas were taken from that research grant. In the GHeI grant, the primary themes were seamless integration, socio-technical infrastructure, integration between patient-level and aggregate data sources. My research design moved forward with these themes.

The first part of my research involved review of Information Infrastructure literature focusing on activities that shape the evolution of the infrastructure. These studies were analysed through discourse analysis and later synthesized into a classification scheme referred to as contribution C1 in the previous section. Parallel to this effort, I was involved in the design and development of a DHIS2 mobile solution in India that can be deployed on low-end Java-enabled mobile handsets. This work later moved to research in Study 1 where I looked at design and architectural factors that enabled the mHealth solutions to scale in terms of no. of users using the system. This study had a direct relation on Study 2. We made minor extensions to the previously mentioned mHealth solution and took that for implementation in Malawi. The main focus of Study 2 was to understand the changes in mHealth solution due to the differences in practices of the context. We faced a number of challenges during implementation in Malawi, which ranged from infrastructure issues to software development issues. This is described in detail in Chapter 3. While we learnt that solving local issues in Malawian context was important, we needed to understand how contextual development can be better articulated in software development. This led to Study 3 in a large and distributed open-source community, similar in domain, size and focus on LMIC context. Here in Study 3, I learnt that a new software development methodology has helped the OpenMRS community to share knowledge among developers and this has in turn led to improved contextual understanding from implementers. The opportunities and challenges of this new agile software development method are discussed in Paper 3.

While the mHealth implementation was scaling to 5000+ users, we were also required to do a security audit of the HMIS system, so that it could be installed in the Ministry of IT (MoIT) servers. In this security audit process, we saw that a number of design decisions that were taken in the HMIS system were to fit contextual requirements of implementations. Though this made DHIS2 better aligned to user needs, yet the testing methodology adopted by MoIT discovered security problems resulting from these contextual requirements. This formed the starting point for Study 4, where I was closely aligned in making changes to DHIS2 codebase, so that the security certification could be

achieved. The theorization in this study included experiences from the Study 2. We had to change the core codebase in such a way that security requirements should not create usability challenges for the global codebase that is implemented in a number of countries around the world. I also included some part of the open-source community model from Study 3 in this study so that global team and local team where the security audit was taking place could be synchronized in their goals towards security. Here, we realized that while the change from non-networked society to networked society caused changes in processes, aligning to older non-networked processes creates security challenges that are often unforeseen. This is discussed in more detail in Chapter 5 of the thesis.

Finally based on the DHIS2 global team discussions, I started going through the evolution of the DHIS2 Business Intelligence (BI) tools. These tools have evolved over time with a certain architectural style in mind. Initially DHIS2 relied on a distributed, standalone system in health facilities and thus BI tools were developed on this architectural view. Over time the DHIS2 deployments moved to internet-based servers that provides access to BI tools for a large number of facilities. Further, the Operational BI tools from a single DHIS2 instance are now used by multiple regions or even multiple countries e.g. WAHO uses a single DHIS2. This architectural change resulted in BI Tools that are used to manage big data. In Study 5, we discuss the constraints that have led to building tools that can be used to manage big data. Study 5 is influenced by the learnings of Study 1 because we define big data in terms of *Organizational Capabilities*, similar to the factors that were discussed in Study 1. Directly related to Study 5 was Study 6, where we aimed to understand how cloud computing can play a role in managing big data for healthcare in LMIC. In Study 6, we conclude that cloud computing models can improve use of *Operational BI* tools, which in turn mean that information can become more actionable due to access and ease of use presented by the cloud computing model of *Analytics-as-a-service* (AaaS).

List of papers and contributions

This thesis is a collection of six papers and the five chapters presented here forth are discussions on these papers and their findings. A paper from each previously mentioned study has been included in the appendix section of the thesis. All papers have contributed to the theoretical synthesis i.e. C1 – a taxonomy to classify activities in establishing of an IeHI. Table 1 is summarization of the papers and their contribution to the thesis.

Table 1: Summarizing papers and contributions (✓-explicit; * - implicit contribution)

	C1	C2	C3	C4	C5
P1	✓ (RQ1)	✓	*		*
P2	✓ (RQ1)	✓	✓ (RQ2)	*	
P3	✓ (RQ1)		* (RQ2)	✓	
P4	✓ (RQ1)	✓ (RQ2)			
P5	✓ (RQ1)	✓			✓ (RQ2)
P6	✓ (RQ1)				✓ (RQ2)

P1: Purkayastha, S. (2013). Design and Implementation of Mobile-Based Technology in Strengthening Health Information System: Aligning mHealth Solutions to Infrastructures. In I. Management Association (Ed.), *User-Driven Healthcare: Concepts, Methodologies, Tools, and Applications* (pp. 689-713). Hershey, PA: Medical Information Science Reference. doi:10.4018/978-1-4666-2770-3.ch034

Book chapter is an extension of two earlier papers, with additional material from Kenya: Mukherjee, A., Purkayastha, S., & Sahay, S. (2010). Exploring the potential and challenges of using mobile based technology in strengthening health information systems: Experiences from a pilot study. *AMCIS 2010 Proceedings*, 263.

Braa, K., & Purkayastha, S. (2010). Sustainable mobile information infrastructures in low resource settings. *Studies in Health Technology and Informatics*, 157, 127.

The work in P1 is an extension, where I've analysed factors that have allowed scalability of the mHealth solutions in India and Kenya, both of which were focused on simplicity and *bootstrapping* the installed base. Further, the two cases were able to scale because they rely on making use of an established health information infrastructure from an HMIS as backend. This paper has directly participated in contribution C2 because it shows the effects of adopting an architectural strategy for scaling and practices that help in defining local successes in mHealth implementation. Further this paper implicitly contributed to C3, since we took the solution developed in this implementation and modified it for Malawian context and C5, because it highlights the governance regime that is needed to scale mHealth. This study implicitly aids C5, where new models of cloud computing accessed using mobile devices can allow sustainability of scaled mHealth solutions.

P2: Purkayastha, S., Manda, T. D., & Sanner, T. A. (2013). A Post-development Perspective on mHealth – An Implementation Initiative in Malawi. In *2013 46th Hawaii International Conference on System Sciences (HICSS)* (pp. 4217–4225).

All authors have made equal contributions in paper P2 and were equally involved in data collection during field work in Malawi and analysis of the material. The paper writing was initiated by me, to which the other authors added later. The paper was presented at the HICSS, 2013 conference by me. The paper has contributed directly to C2 and C3 by linking architectural decisions to challenges that appear when only considering macro-level success factors. The paper contributes implicitly to C4 because it highlights the need for contextual development for meeting local needs. We conclude in the paper that meeting local needs is the “true” success for mHealth implementation. Repeatedly meeting these needs in multiple locales should be the strategy to scaling, instead of scaling one solution to many users (depth scaling) or many domains (breadth scaling).

P3: Purkayastha (2014). OpenScrum: Scrum methodology to improve shared understanding in an open-source community (under review)

The paper is a longitudinal study on the OpenMRS community, where new agile development model was adopted. This paper’s core contribution is C4, called OpenScrum, which improves shared understanding about the codebase and user requirements. The paper contributes to C3 by discussing how community governance regime can help in understanding requirements. The effects of governance principles on software development practices are implicit contributions of the paper.

P4: Purkayastha, S. (2011). Towards a contextual insecurity framework: How contextual development leads to security problems in information systems. In *Proceedings of IRIS 2011* (pp. 654–666). Turku Centre for Computer Science.

Paper P4 describes the security certification process of an HMIS system and highlights the security risks that have been inscribed due to contextual software development. The paper contributes to C2 by highlighting the effects of non-networked practices of using paper-based systems and the information security risks that are inscribed into networked software in the internet/computer-based systems. In the effort to meet local needs (C3), practice changes are required in both security testing, as well as software development.

P5: Purkayastha, S. & Braa, J. (2014). Overview, not Overwhelm: Framing Big Data Using Organizational Capabilities. (under review)

As discussed in the introduction, leHIs are being established in many LMICs. There is an imminent challenge for use of the integrated information gathered in data warehouses. In this paper, we discuss the 3 generations of Operational BI tools in DHIS2 and argue that big data should be defined in terms of Organizational capabilities and not just

technical stats (C5). The evolution observed here contributes to C2 by highlighting how architectural decisions in BI tools were driven by requirements that were action-led and not data-led. We distinguish between Overview information space and Overwhelm information space in an organization based on its ability to use big data with limited availability of analytic tools.

P6: Purkayastha, S., & Braa, J. (2013). Big Data Analytics for developing countries – Using the Cloud for Operational BI in Health. *The Electronic Journal of Information Systems in Developing Countries*, 59(0).

The paper is an extension of P5, where we look at cloud computing models for sustaining analytical tools in LMICs. The challenge of lower technical capabilities in LMIC has often dented sustainability of HIS efforts. By using cloud computing AaaS model, the digital divide for use of information can be reduced and further IeHIs can become action oriented.

Thesis Structure

In the next Chapter, I describe the state of the art from which I've drawn concepts, notions and theories that have guided my research. Chapter 4 presents the results of the studies which are bounded together to create a coherent nomenclature that is later presented and discussed in Chapter 5. Here, I suggest a new Genus for classification of activities within an Information Infrastructure. Chapter 6 is the concluding chapter of the thesis, which summarizes and presents future avenues for research. In the end, I've included a secondary paper which highlights the future direction of IeHIs. Three appendices add material that have been used in the individual studies, which could not be included in the published papers due to length constraints or scope limitations.

Chapter 2: State of the Art – Theoretical synthesis and contribution

Chapter 3: Context and Research Design

Chapter 4: Results

Chapter 5: Evaluation and Discussion of Results

Chapter 6: Conclusion

Appendix A: (enclosed, selected papers)

Appendix B: basic info incl. abstracts of secondary papers

Appendix C: Additional methodology material, not part of the paper

Chapter 2: State of the Art

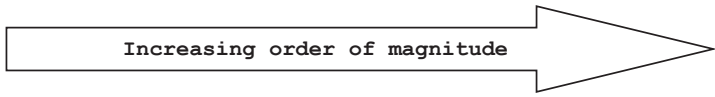
Part 1: Theoretical Synthesis

Information Infrastructures (IIs) have become a widely studied IS phenomenon over the last 20 years. It has been described by terms such as e-Infrastructures, cyberinfrastructures, electronic infrastructures, knowledge infrastructures, e-science, critical information infrastructures etc. With some subtle differences in the above terms, mainly due to domain of use, these terms broadly talk about the multitude of information and infrastructure technologies used by people for their work practices (Grisot et al., 2014). IIs are often described as socio-technical phenomenon, to which classical model of organizational change does not apply (Orlikowski & Hoffman, 1997). IIs are different in every stage - from design, development, implementation and evaluation, compared to standalone IS. In fact design of IIs often results in *drift* (Ciborra et al, 2000) due to multiplicity of user needs and inertia of *installed bases* (Grindley, 1995; Star & Ruhleder, 1996; Monteiro, 1998) and results in *unanticipated effects* (Monteiro, 1998; Hanseth & Ciborra, 2007). Hence, literature describes establishing of IIs as a process of cultivation or as “*infrastructuring*” (Hughes, 1983, 1989; Scott, 1998; Pipek & Wulf, 2009) instead of *building*. While some have described the cultivation process as *bricolage* (Ciborra, 1994), the unending process of improvisation, a sort of situational tinkering (Ciborra, 1997), the goal of this thesis is to highlight that the multiple smaller action-events taking place have deeper, invisible connections, particularly under broad categories in which these actions take place. The theoretical synthesis focuses on some common themes and conceptual underpinning that can be found in the type of cases, described here as bodies of literature.

To understand the background of how II literature describes the phenomenon, a helpful process is to look at the historical evolution of II literature and concepts from which it has evolved. The point I’d like to make at the very beginning of this synthesis is that the bodies of literature described below have large overlaps. As in any active field of science there is exchange of ideas that is sometimes direct and other times more subtle and nuanced. The attempt in the sections below is to classify the vast body of literature, as authors have themselves described, but also to draw correlations and dissimilarities of the concepts and create a base for my theoretical contribution in Part 2 of this chapter. An interesting aspect that I attempt to highlight while looking at this theoretical evolution process is to see how each body of literature carved a niche for itself by arguing against existing theoretical and conceptual views.

One can think of the measure that separates standalone applications to IIs in terms of complexity – the number of irreversible interconnections between the internal components (human and non-human) of the system; size & heterogeneity – the number of different components in the system; domains of use – the user needs that are met by the system. The separation of the bodies of literature is based on these points as can be summarized in the table below:

Table 2: Scaling from standalone applications to Information Infrastructures

	Standalone application	Information Systems (IS)	Information Infrastructures (IIs)			
			Intra-org IIs	Inter-org IIs	National/Regional IIs	Global IIs
Complexity						
Heterogeneity & Size						
Domains of use						
e.g.	Word processor	Shared filesystem	Lotus Notes	Email systems	Health Information Exchanges	Internet

The below bodies of literature are in some sense representative of Table 2. The core II body of literature defines what is general to information infrastructure per se. Cyberinfrastructure, knowledge infrastructures and e-Infrastructures are specific to a domain of use or within organizations or between organizations that want to meet a common goal together. NII/GII bodies of literature look at National, Regional or Global infrastructures that are the widest in terms of complexity, heterogeneity and size as well as domains of use that are supported.

Information Infrastructure (core):

In this section I classify the body of literature as core II, because the authors themselves have seminally described their own work as the base of II theory. The ideas in this body of literature have theoretically conceptualized IIs to be different from traditional IS implementations. This body of literature describes II as the observed phenomenon where experimentations over time results in emergence of complex constellation of local applications and pockets of local knowledge (Star & Ruhleder, 1996). Here, “infrastructural inversion” is used as a term where infrastructure appears only as a relational property and not as a thing (Bowker, 1994). The studies in this body of literature often describe the tensions between standardization and flexibility (Hanseth, Monteiro & Hatling, 1996) and long term changes in the process of cultivating the

installed base. The interesting and recurring point that the authors make here is that the process of standardization is much more nuanced than top-down diktat that comes from standardization bodies. Many researchers (Haux, 2006; Hanseth & Aanestad, 2003) have studied how the move from singular hospital systems to integration between many specialized systems needs gateways and multi-layered approach for evolution. The above researchers also highlight that information flow in such integrated system require constant nurturing. The work of Braa et. al. (2007) has articulated the strategy of *hierarchy of standards* or *flexible standards* as one architectural approach to deal with the hierarchy of information needs in the health sector. Rolland & Monteiro (2002) deal with varied user needs, most IIs need to be designed with a generative framework (Abbate, 1999) as can be seen from the example of building the Internet which involved many small, evolutionary and pragmatic decisions.

The unpredictable speed of change, which sometimes is fast and other times very slow (Hanseth & Braa, 1998; Ure et al., 2009), is another observation in this body of literature that is commonly found in biological complexity that is part of Complex Adaptive Systems (CAS). We see the property of *inertia* in IIs due to which the infrastructure is difficult to change. This installed base in itself acts as a force multiplier when it brings newer actors into the II. The conundrum is such that new actors also bring change in the II. The seminal work for conceptually framing IIs came through actor-network theory (ANT), with minimalistic use of concepts like *inscription* of behavior in IIs (Hanseth & Monteiro, 1997) that happens through the process of *translation* (Monteiro, 2000) which generates the ordering effects such as devices, agents, institutions, or organizations (Law, 1992 pp. 366) and *boundary objects* (Star and Griesemer, 1989) or more specifically *gateways* (Monteiro & Hanseth, 1996) which are objects that allow translation between actor-networks. There have also been fringe attempts at introducing concepts from *Diffusion of Innovation* (Lyytinen & Damsgaard, 2001), *Social identities* (Gal, Yoo & Boland, 2008) for conceptually describing actors in IIs. But the dominant framing has been to treat actor-networks as an assemblage, instead of viewing actors as having distinct identities from their networks. Over the years, II studies have also formed concepts such as *Gestell* (German word for 'framing') as the hidden support for IIs (Ciborra & Hanseth, 1998), *Bootstrapping*, as the process by which IIs enroll the first and later more users (Hanseth & Aanestad, 2003), *Reflexive standardization*, as unintended consequences where when we try to achieve order and closeness we get chaos, openness and instability (Hanseth et al., 2006) and more recently *Grafting*, as a process to balance control and cultivation (Sanner, Manda & Nielsen, 2014). I will digress a bit from this theoretical summarization, and I use the word digress because the next is an important methodological concept, (instead of the theoretical concepts) in designing IIs. The design methodology uses concepts from CAS

theory to derive design rules that can cultivate the installed base and promote dynamic growth of II (Hanseth & Lyytinen, 2010). The concept of Dynamic complexity of CAS is used to link the concepts of bootstrapping and *adaptability* in IIs that are commonly referred to as challenges in establishing IIs in previous literature.

As can be seen in these studies, Science and Technology Studies (STS) has been the native philosophical territory for II conceptualization. Although the cases come from different sectors, the main focus has been on groupware systems, healthcare, oil and gas. The conceptual belonging of this body of literature, along with comparison to other streams is listed in Table 3. To broadly summarize, this body of literature theorizes complexity resulting from the interplay between people and technology, attempting to give equal visibility to both the technology artefact and users. At its core, this body of literature provides a more deeper theoretical view of human and non-humans than structuration theory – structuration theory studies lack the satisfactory level of precision (Hanseth & Monteiro, 1997), while accepting the broad understanding that technology plays an enabling and constraining role (Orlikowski 1991, 1992; Orlikowski and Robey 1991; Walsham 1993).

National/Global Information Infrastructure (NII/GII):

While the previous body of literature was broadly set in the STS field, this body of literature focused on policy and intellectual property rights about the objects in infrastructure and the people building infrastructures (IITF, 1995). While IIs have been studied in a number of fields beyond Information Systems, at this point it is important to highlight that here I only attempt to include views that are pre-dominantly in the IS field. A primary topic was identification of the differences in the policies in law and comparing them with the social and cultural values among the users of the infrastructure (Kahin & Nesson, 1996). A common theme is to reflect upon repeated attempts at standardization and the strategies to implement policies for standards of practice (Kahin & Abbate, 1995). Halamka et al. (2005) highlight the importance of co-ordination through policy framework to create health information exchanges, which are large scale IIs for exchange of standardized patient records. Some researchers have shown that a shared governance mechanism allows integration in much better ways (Porter-O'Grady et al, 1997). In a somewhat opposing idea, Titlestad et al. (2009) suggest that distributed development allows better user participation. Detmer (2003) suggests distributed, yet collaborative efforts are needed to establish IIs. In similar terms Yasnoff et al. (2004) conclude that a central consensus building initiative enables building health IIs. The notion of *property* and *innovation* in GII (Perritt, 1996) is another common recurring theme, with concepts from law and public administration often used as native ideological territory. The protection of rights in terms of privacy and security (Pironti,

2006) in IIs also plays a vital role in such studies, often describing NII or GII as critical information infrastructure (Dunn, 2005; Cavelty, 2007). Pironti (2006) hence defines IIs as all of the people, processes, procedures, tools, facilities, and technology which supports the creation, use, transport, storage, and destruction of information.

Beyond *property* and *rights*, other common concepts used are *efficiency* (Dedrick & Kraemer, 1995), *competitive advantage* (Schware, 1992) and how NIIs could bring some form of *equity* between various users of the infrastructure (TCOJ, 1994). While we are at the concept of equity, it is important to highlight that a number of researchers studying NII/GII talk about complex power hierarchies in IIs. Power, as a concept has been studied by a number of IS studies, particularly as organizational politics (Kling, 1980; Markus, 1983; Robey and Markus, 1984), yet when it comes to IIs, the relationship between innovation, power and organizational change is described through *Networks of Power* (Latour, 1997), which is much more complex and nuanced (Sørensen, 2005; Constantinides & Barrett, 2006). I will digress again, as in the previous section to highlight an important breakthrough in IIs from a methodological standpoint to establishing GII. *Networks of Action* (Braa et al., 2004) is an action-research methodology for IIs, as it considers local action-research interventions as only an element in the larger action-research in GII to ensure sustainability and scaling of interventions to establish IIs. Here the authors use ANT to articulate the GII of HISP that has been doing action research for over 20 years, especially in politically contested terrains with opposing projects and ideologies.

A large of part of this literature often articulates struggles between empowerment and imperialism (Rothkopf, 1997; Mayer-Schonberger & Foster, 1997; Borgman, 2000). With the same thought process, the sub-field of ICT4D often uses IIs to describe the interactions between North-South of the GII, where north has the resources and expertise, and the south is characterized as “laggards” (Mutula & Brakel, 2006) due to the need for technology and lack of resources and expertise. Researchers have also seen a significant relationship between NII and *governance* and also between NII and economic development (Meso et al., 2009). Here, a number of researchers balance concepts between theories such as critical theory (Dahlberg, 1998; Gunkel, 2003) and structuration theory (Evans, 1999; Rolland & Aanestad, 2003). I use the phrase “balancing of concepts” instead of merging, because some researchers have held the view that structuration theory (see previous body of literature) doesn’t have the required granularity to study IIs. This body of literature, as such, has fluently combined concepts from many theories including from actor-network, structuration and institutional theory (Scott, 2001). E.g. the concept of installed base was described using the concept of *institutionalization* (Rolland & Aanestad, 2003), where artifacts and

embedded practices become part of the IIs in such a way that they are difficult to change. Similarly, HISP's struggle to establish an HIS in South Africa has also been deeply conceptualized by combining concepts from actor-network and *routines* and *institutionalization* (Braa & Hedberg, 2002). Using *marginalization* from Castells (1996), *counter-networks* was conceptualized (Mosse & Sahay, 2003) and the use of *innofusion* (Fleck, 1994) to create the concept of *configurable politics* (Sahay et al., 2009) also highlight this balancing act. The above two concepts, make indirect use of actor-networks, but not networks of power (Hughes, 1983; Latour, 1997) as earlier mentioned researchers used it to conceptualize observed phenomenon of politics in cultivating II. A broad summarization of this literature can be described as the focus on policy, politics and power during the cultivation process of a GI/NI. The cases in such studies have been much broader ranging from Internet, defense, GIS, health care, transportation and communication systems.

Cyberinfrastructure/e-Infrastructures/Knowledge infrastructures:

There has been another body of II literature that I could have called miscellaneous topics. But that will be complete ignorance of the commonalities that can be seen in these studies, besides the concepts I've already described in the above two sections. Calling it miscellaneous would also be incorrect, as use of terms such as cyberinfrastructure (as it is known in the USA) or e-science (as it is called in Europe) is central to conceptual framing in study of IIs (Edwards et al., 2009). These have been commonly called e-Infrastructures that are large multi-disciplinary networks, but built for firms, government or scientific enterprises with a specific purpose in mind, unlike the generic purpose of the internet (ibid.). *Alignment* of work practices has been a key concept (Joslyn & Rocha 2000; Bowker & Star, 2000) in this body of literature. Local needs and matching these with larger organizational needs has been articulated by a number of researchers. Some have described this matching of work practices as seamless integration (Schweiger et al., 2007) and many consider this to be largely a technical issue (Grimson et al., 2000; Xu et al., 2000). Yet it can be seen that in many cases organizational challenges in integration, often results in disorder (Ellingsen & Monteiro, 2006). This dialectic of order and chaos in the process of standardization as previous mentioned is referred to from an architectural perspective as *reflexive standardization*. In existing literature, I see two perspectives to be relevant in this view of matching local needs and global use. One perspective is there is a balancing act between the local and global through deep engagement between multiple contexts (Miscione & Staring, 2009). The second perspective is where the theoretical distinction between the local and global vanishes (Ellingsen et al., 2013). These in fact become deeply interdependent and can be considered to be workarounds of one-another, such that there are no separate strategies for implementing global or local infrastructures.

Dynamics, tensions between the socio-technical components of an infrastructure are core to this body of literature. In their charting of infrastructure history, Edwards et al. (2007) look at it from Dynamics, tensions and design classification. Dynamics in IIs can be dealt with using previously listed concepts like *gateways* or through use of other *boundary objects*, but concepts like *reverse salients*, a process where innovation needs to be halted because the whole infrastructure cannot change at once (Hughes, 1983, 1987) and *path dependence* find more often mention in this literature. Reverse salients or Rosenberg's (1994) *technical imbalance* as a concept has strong similarities to the property of inertia shown by installed base in previous bodies of literature. Yet the deeper connotation of the term with capabilities between the elements of the infrastructure, without using actor-networks is found more commonly in this body of literature. While *path dependence*, as a concept in IIs are described through two facets, *cumulative adoption* and *technology traps* (Hanseth & Lyytinen, 2010), both these terms come from diffusion of innovation and innovation theory, on how adoption in infrastructures broadly follows an S-curve (Nakićenović and Grübler, 1991). Path dependence and its related terms come broadly from complexity science and CAS theory. As previous bodies of literature, tensions find mention and primarily viewed as a problem of heterogeneity of needs and a balancing act between these needs (Ribes & Finholt, 2009). In this body of literature there is much more effort to classify infrastructures, which is lacking in the previous bodies of literature. Like Bowker (1996) uses the history of ICD and argues for classification schemes from various sciences as being infrastructures. Similarly, Bowker & Star (2000) describe the invisible mediators that are used to classify research and describe this through an infrastructure lens. We also see the attempt to classify infrastructures into "shaggy dog" and "trimmed poodle" stories (Ribes, 2014) based on their kernel (core), with factors such as research objectives, social organization, objects/tools that are part of the infrastructure. Although this body of literature's focus on classification is on the observed phenomenon, it hasn't done much on theorizing or theory building effort regarding infrastructures.

Co-relation and Dissimilarities: a synthesis

Going through the above list of concepts and their origins, we can see that many of these concepts have large overlaps and deep connections, either because of their common theoretical inheritance or simply because the observed phenomenon is similar and only has been dealt with from different angles. Let's look at the concepts of installed base and its creation. A number of authors have looked at the challenge of enrolling new users into an II – adaptability problem and bootstrapping problem. Adaptability for example has also been described in terms of *flexibility* and *openness* of the II. In fact in

their seminal work Hanseth & Monteiro (1997) define II as a shared, **evolving**, **open**, **standardized**, and **heterogeneous** installed base. Each of these characteristics has a linked theoretical concept embedded in it. The evolving process has been described as *Infrastructural inversion* in the book *Sorting things out* (Bowker & Star, 2000) and has connections to the concept of *Figure and Ground* (McLuhan, 1967), broadly referring to the “medium is the message” in infrastructure, as the study of the members of the infrastructure as studying the infrastructure itself. Figure-Ground, as we know comes from Gestalt psychology, which interestingly in the concept of *Gestalt* where the organized whole is considered to be more than the sum of the parts.

In **openness**, we should realize that a lot of II literature that talks about equity, also talks about competitive advantage, property rights and *counter-networks*. This shows a form of dialectic between control and loss of control (*drift*) in the II. Control is demanded by the creators and powerful actors-networks in the II, whereas in applying stronger control through a **standardization** process, we see a loss of control as described by the concept of *reflexive standardization*. Something that has worked towards *institutionalization* of a practice is *routines*, which is different for different type of users. This differences of routines and for that matter institutionalized practices is described as user needs. This is something where a lot of thought from economists, innovation literature has focused on. Concepts such as *reverse salient* and *path dependence* are deeply rooted in observation with heterogeneity. The point to draw is that complexity science theories have contributed to these concepts and CAS is one of the theories that has been used to articulate these concepts. It clearly shows that II literature has gone much beyond the defining statement as mentioned by Hanseth & Monteiro (1997), which only drew from actor-network theory.

Table 3: Theoretical background of the concepts

Actor-network theory	Complex Adaptive Systems	Institutional theories	Innovation theories
Translation	Reflexive standardization	Institutionalization	Bootstrapping
Inscription	Path dependence	Counter-networks	Reverse salients
Boundary objects	Variable speed of change	Marginalization	Innofusion
Gateways		Routines	Property
			Competitive advantage
			Equity

Broadly though, the ideas of tensions due to user needs is core to this phenomenon of II. As Ribes (2009) or as Sanner et al. (2014) describe it, this balance between varieties

of user needs is essential in cultivating the II. Standardization and Control is often described and in some sense incorrectly understood as only a top-down effort from creators of IIs. For a creator/designer of an II, it is often not a choice to balance the user needs, but rather a necessity so that users of all types of work practices will enrol into the II. While the practices of users differ, there are also number of artefacts that are now supposed to be integrated in the II, were initially created for a specific work practice in the local context. This challenge of global and local is described again through concepts that are from a practice perspective, but never separated from the artefacts that inscribe these practices. Thus, there is often little separation in II studies, between the artefacts and the designers of the artefacts. These artefacts become part of balancing control and the assemblage of actor-networks (users and their artefacts) is rarely separable.

Where this separation of people and their tools has been useful must also be described. The GII/NII literature which talks about “property, rights and equity”, tries to describe the tensions between users of the infrastructure in terms of policy decisions. Policy decisions can be understood as standards of practice (Detmer, 2003) and hence are quite similar to the standardization efforts described earlier. These policies can range from ways in which teams work to inscribe their practices in the artefact or practices of the users who will use the artefact. This distinction is particularly important as the standards are often described in the tools and may not be created within the II itself. Thus, newer standards of practices that are *workarounds* of pre-integration standards of practice are maintained through governance policies and become mechanisms by which designs are enacted during implementation of II. So also the concept of *reverse salients* describes that some parts of the infrastructure are “laggards” and to resolve this, certain policies have to be enacted, by which they can be brought up to pace with the rest of the infrastructure. While the innovation or speed at which the II grows has to be slowed down, there is a view that such policies are necessary to move the infrastructure forward. To delve deeper into what activities need to be done to move the infrastructure forward, let us consider the broader terms on what the above three bodies of literature have prescribed.

Architecture: The oxford dictionary definition of architecture is “the **complex** or **carefully designed** structure of something”. What is this something we may ask? It has been from buildings, computer software or hardware to even organizations and societies. Oxford dictionary also defines architecture as the art and practice of **designing** with a style with regard to **context** – period, place or culture. Architecture has been used to describe buildings through durability, utility and beauty (Alexander, 1979). Yet little of these concepts have found direct references in IIs, though robustness, usefulness and

non-uniformity has described the II phenomenon. In software or hardware architecture, we have seen practice of componentization or building blocks that can be re-used across the system (Shaw & Garlan, 1996). Architecture is often expressed through two ways – (1) using diagrams composed of re-usable or non-re-usable boxes or blocks; (2) using processes and information exchange between the processes and the boxes and blocks. Yet these boxes from architectural diagrams have often been critiqued by architects themselves (Bosch, 2004; Bass, 2007) and have suggested the need to use more than the boxes of Unified Modelling Language (UML) (El-Atawy, 2006). Instead to use more action-events based use-cases or descriptive user stories as narratives to support an envisioned workflow. The concept of architecture in II has mainly been used to talk about assembling together of technology artefacts. Like Foster and Kesselman (2003) in their book described the Grid 2, a new GII that will be a computing infrastructure, similar to the cloud computing initiatives that we are seeing now. Although they describe architecture through technical artefacts, a more infrastructural paradigm to architecture is also commonly seen in the upcoming cloud computing literature. Buyya et al. (2009) suggested that computing will one day be the 5th utility after water, electricity, gas and telephony due to cloud computing. Some of the currently available service models for providing such computing utility include Software-as-a-Service (SaaS), Platform-as-a-Service (PaaS), Infrastructure-as-a-Service (IaaS) and Analytics-as-a-Service (AaaS). These service models describe architecture based on use of the computational services that the infrastructure provides. Another upcoming, but closely linked architectural phenomenon is Big Data. Because of the interconnected-ness of users, services, platforms, data in such infrastructures is more often described by the 4Vs - high velocity, high volume, high variety and low veracity (Ohlhorst, 2012). Big Data is a term widely used in popular media. Words like “Petabyte Age”, “Industrial revolution of data” are common, yet haven’t we had huge datasets of petabytes running on super computers for bioinformatics, space research or other high-performance computing domains for at least the last 20 years? What makes the current times unique is that never before have the “masses” been involved in data creation exercise at this scale, nor has so much general computing power become available. Academic definition for the term is particularly hard to find, but industry reports have defined Big Data with wordings such as:

“Big Data is data that exceeds processing capacity of conventional database systems” (O’Reilly Media)

“Any amount of data that's too big to be handled by one computer” (Amazon)

“Big Data is data with attributes of high volume, high velocity, high variety and low veracity” (IBM)

“Big Data refers to datasets whose size is beyond the ability of typical database software tools to capture, store, manage and analyze” (McKinsey)

From these reports and related literature, we also see that Big Data is not only large volumes of data, but also contains complex interconnections, similar to what has been

described as the property of the *infrastructure* phenomenon. So the term in itself is somewhat poorly descriptive (Manovich, 2011). Similarly, Tsiknakis et al.(2002) use architecture to describe the components (technical and non-technical) that are open to integration in a health information network. In fact, McGarty (1992 pp.235-236) defines II with a detailed definition through the terms: shareable, common, enabling, **physical embodiment of an architecture**, enduring, scale and economically sustainable. There are also certain specifics in the kind of architecture that can support an II. E.g. open Vs closed architecture (Iwata et al., 1997), centralized vs decentralized, top-down vs bottom-up, hierarchical control vs user-control. But architecture in IIs need to be described in more granularity than just these broad classifications (Monteiro & Hanseth, 1996). To summarize and that is the reason I bring up the word architecture is to ask - how do we design IIs, which are off course **complex** but with **careful** attention to **context** needs to be established. So I reword my question – how can IIs be architected? To some people, this wording might sound like that we have a concrete step by step plan that once followed results in a standing building. And so the more commonly used word of “cultivation” or “infrastructuring” provides more fluidity and is less deterministic? Infrastructuring is more of an aphorism, one where you need to first understand the term infrastructure itself. Cultivation on the other hand is a more generally understood term, which according to Oxford dictionary – “prepare and use (land) for crops or gardening or try to acquire or develop (a quality, sentiment, or skill)”. It originates from the Medieval Latin - *cultivat*, which is about crops. The original authors surely drew allegory between the terms of cultivation and infrastructure building, primarily to suggest that not every sowing results in the same output of crop, given the large number of factors involved in determining output. Thus, trying to point to the non-deterministic nature of the establishing of an II.

On the other hand, let me deconstruct the word architecture for you. Design of IIs as we have seen in the literature plays a vital role, in such that it has be generative and also by factoring the context. We have seen that small pragmatic decisions resulted in establishing the internet or in a number of cases where the design was primarily to be able to align the different actor-networks through gateways or translational objects/boundary objects. We have also seen that large GII have been referred through the embodiment of an architecture. Thus, architecture becomes like a vision under which topics of design and contextualizing the design takes place. The word design alone does not convey the contextual and careful articulation of the activities that have been described in literature. Instead, I suggest that the activities described in literature fits well with the meaning of the word architecture that means more than just design and includes more concepts that have been referred to in II literature. By any means, this alone does not list all the activities that “cultivation” or “infrastructuring” convey. This

is just one of the broad headlines under which topics of design, contextualizing, planning can be brought together. Let me describe the next two broad headlines.

Governance: We have seen governance described in GII/NII literature largely through the concepts of **policy** and formulating **laws** by which these policies can be enacted. Rhodes (1997) in his widely cited book defines governance as the processes implemented by policy networks to define reflexivity and accountability. In more lay terms, it is the processes and decisions that define actions, grant power and verify performance (Bevir, 2013). From these two definition, it seems that governance is largely a mechanism by which decisions are made between participants and the role and activities that they are **supposed to perform**. Here supposition is important to understand because through governance mechanisms, one only hopes that its participants will act in an envisioned way and formulates laws to punish/reward participants that do-not-perform/perform in the envisioned way. Consensus in this context is a subtler notion that the participants have decided among themselves what the envisioned way of working is going to be. This has also been described as policy networks (ibid). Policy networks are pragmatically individuals or organizations that follow a policy and make decisions based on that policy (Rhodes, 1997).

For example, I look at global software development (GSD) in open-source software (OSS) as being an instance of policy networks, where decisions are made by software developers working towards a common goal. To bring a form of *Agility* and *Flexibility* to decision making, a commonly used methodology is Agile Software Development (ASD). A recent review by Jalali and Wohlin (2012) highlights that GSD projects with agile methods are extremely rare. This might be primarily attributed to lack of clear mention that an open-source community is following a certain agile methodology. Some researchers have asked if open-source software (OSS) development is essentially an agile method (Warsta & Abrahamsson, 2003). But Koch (2004) mentions similarities, but also points out differences between agile software development (ASD) and OSS development. Early work on ASD focused on defining agile methods (Beck et al., 2001; Highsmith & Cockburn, 2001; Williams & Cockburn, 2003), adoption of agile methods (Boehm, 2002; Nerur, Mahapatra & Mangalaraj, 2005), efficiency of agile methods (Nawrocki J, Wojciechowski, 2001) and then more recently focus on empirical studies about post-adoption issues of agile methods (Cao et al., 2009; Mangalaraj, Mahapatra & Nerur, 2009) and team management (Moe, Dingsøyr, Dybå, 2009). While improved software quality is an observed output, the above researchers highlight “agility” as the most important criteria for adoption of ASD methods. “Agility” in such cases has been used to describe the ability to rapidly and flexibly create and respond to change in the business and technical domains. “Agility” is achieved by having minimal formal

processes. Often used concepts to describe “Agility” include nimbleness, quickness, dexterity, suppleness or alertness. These ideas suggest a methodology that promotes manoeuvrability and speed of response (Cockburn, 2006). OSS communities are generally seen as a collaboration of individuals or organizations that participate in software development without contractual bindings, but rather enjoyment-based intrinsic motivation (Lakhani & Wolf, 2003). Some researchers have suggested change in practices (like OSS 2.0) (Fitzgerald, 2006; Crowston et al., 2008), where OSS development is moving towards commercial participation. There is also more recent suggestion that OSS is still largely a combination of commercial ventures and volunteer contributions (Krishnamurthy, Ou, Tripathi, 2013). Sustainability is often an issue in open-source communities where volunteer contributors “come-and-go” or choose their own tasks (Mockus et al., 2002; Scacchi, 2007). Sustainability of OSS is often described by using the term “truck factor” or “bus factor” i.e. the total number of key developers that would, if incapacitated (e.g., by getting hit by a bus), lead to a major disruption of the project (Stephany et al., 2009). Another challenge that we see in open-source communities is to gather contributors in projects that are for a vertical domain (health-care, finance, human-resources, etc.) In many cases, by strategic planning, paid developers will be assigned to work on open source products in vertical domains. If the revenue model for such planning falls short, the developers are moved to other projects. More than a decade ago the Agile Manifesto clarified about the values of agile software development and put forth principles that can be adopted to meet those values. While much of the practices around agile software development have been promoted by practitioners and consultants, there has been a growing need to conceptualize “Agility” (Conboy, 2009). Here, Conboy suggests that Agility comes from two concepts of Flexibility and Leanness. Although used interchangeably, there are conceptual differences between Flexibility and Agility and also between Leanness and Agility. Thus, to be considered agile, the methodology should contribute to creation of change, proaction in advance of change, reaction to change or learning from change. It should also contribute and not detract from perceived economy, perceived quality and perceived simplicity. These allow producing software which is continually ready, with minimum time and cost required to be put into use (ibid.). While Agility in such terms is an overall measure of the organizational performance to deliver a software product, one should also consider how individual developer productivity is affected by the practice of agile development. While developer productivity has been a hotly debated topic, the 1993 IEEE standard for software productivity metrics defined it as “the ratio of output to the input effort that produced it”. Jones (2000) identified 250 factors affecting developer productivity, while more simplistic summary still lists 15 factors (Endres & Rombach, 2003). So instead of co-relating multiple factors that affect productivity, it is common to measure output such as in Changes in Lines of Code (CLOC) or Non-

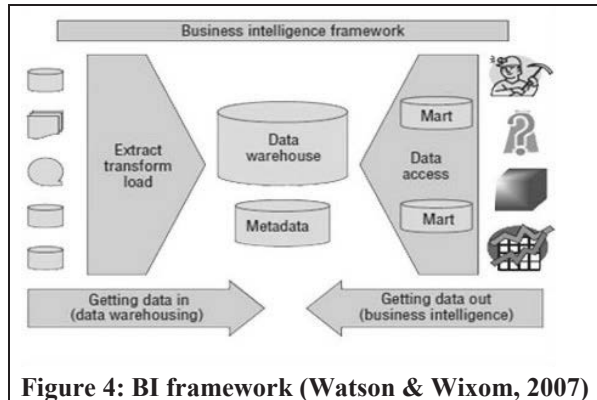
Commentary Source Lines (NCSL) (Mockus, 2009). Another measure of developer productivity through interactive participation has been suggested – mainly through the use of code reviews, comments on other people’s code, number of forks, network analysis of contributors (Singh, 2010). We’ve seen case studies which suggest that communication, co-ordination and control problems in GSE have reduced due to use of agile methods such as Scrum and eXtreme Programming (Holstrom et al., 2006). This and similar research (Paasivaara & Lassenius, 2006; Herbsleb, 2007; Lee, DeLone & Espinosa, 2006) suggests that distributed teams indeed benefit from using agile methods. From all of these cases, we see that there is some level of tweaking done to agile methodology to be relevant to the organization. Tailoring of methods has been observed to play an important role in benefits like reduction of code defects density, delivery ahead of schedule and accurate planning for future projects (Fitzgerald, Hartnett & Conboy, 2006). While this need for tweaking has been well documented, very little has been written about tweaking agile development to open-source projects. OSS projects might simply be GSE projects in the public domain. In all of the above research on GSE, and most research on agile methods (Russo et al., 2009), we see that management control for changing software development practices could be done by a limited number of stakeholders and all these stakeholders were either organizationally or contractually bound.

The work in agile software development is quite similar to the actor-network perspective that is used in II to highlight the correlation between the different parts of a larger network that work towards a common goal (Jackson et al., 2007). Yet it is important to recognize that all participants will not have the same power in forming the consensus. Thus, certain participants will evolve their own workarounds and ways of working that do not break the policy directly, but become a different way of working. Governance for example has been a common theme in the GII/NII literature where it is used to frame policies to bring equity or improve efficiency. Core II literature puts governance more subtly than economists would. Infrastructures often go “out of control” and tactics to govern are more subtle than what the word “management” would convey (Ciborra & Hanseth, 1998). And here the concept of *Gestell* describes a vital role that organizations themselves have limited capacity in governing an II, but rather participate in the II, which has a life of its own, with their own work practices that form part of the II. In this aspect, we can draw on the concept of inscription because developers, implementers or designers who design infrastructure do not just create standards or technology. They also include some of their biases in the design of the II. Governance in such instances, mean the ways in which designers of the II also participate in the II, not just thinking themselves as creators, but participants in the life of the II. Thus, the concept of *Gestell* can very well be understood as the governance principle of

providing support to the II by participating in its practices and not “designing” it. In the same vein, David (1987) points out three dilemmas of (a) narrow window policy, as the short time when policy changes can be made; (b) blind giants, are powerful individuals but who lack the visions to change the course of II’s evolution; (c) angry orphans (similar to concept of *marginalization*), as the individuals who will be left behind by the changes in IIs. To deal with these dilemmas there are also a number of recommended tactics. The reason to list these dilemmas is to show that governance deals with these kinds of issues within the II. We have also seen that governance is used to describe the *configurable politics* where actor-networks change their behaviour to align with more powerful networks. The enabling criteria for such change can be broadly put under the topic of governance because as we have seen from the definition earlier, that reflexivity is an important concept in governance. Based on the stronger network and its effects of power on *counter-networks*, we see a form of alignment and changes in the practices in the weaker network. This is the form of reflexivity that is described in policy network. We have seen in the GII/NII literature that *efficiency* and *equity* are important concepts. These concept describe a form of measurability or as in the definition of governance, accountability of the II. The GII/NII literature talks about framing policies or governance rules to be able to implement efficiency in the II. While we are not suggesting that efficiency is purely for the better and as we have seen from the balancing of user needs, some users have to change their practices more than others. This brings a form of *inefficiency* to their participation in the II, but obviously they have other benefits from participation in the II that they enrolled in the first place. This shows that topics such as these come under the broad classification of the governance.

Praxis: We have seen a couple of important concepts that talk to each other in terms of enactment, embodiment and actual standards of practice. The words actual or enactment is important because it helps differentiate between visions of how an infrastructure should be used and the way in which it is being used, as we have seen involves many unintended consequences. Thus, while Architecture and Governance are visions and attempts at defining and describing how the infrastructure should be used, its use in practice is different. Some architectural principles of II in fact highlight this correlation. *Flexibility* and *Generativity* as concepts describe this fact of successful infrastructures, where participants (actor-networks) of the II create more incentives in the II and fit it to their own needs. Thus, flexibility to fit it to ones needs and generativity, by creating newer ways to expand the infrastructure are critical to the way processes or activities take place in IIs. In the quest to be able to expand use, users expect architectural visions to include tools that will enable ease of work. For example, the architecture of cloud computing models, expects users to use computing as a utility. But one might ask whether there are adequate tools included in the infrastructure that will

allow users to understand Big Data through cloud computing? Over the last 50 years, Business Intelligence (BI) has been expected to change the practice of decision making in organizations. These have been over the years re-invented as Decision Support Systems (DSS), Expert Systems and Executive Information Systems (EIS) (O'Brien, 1991). Much effort has been tied to automation or support of human decision making.



Still, systems that provide process-based information for decision making have only become top priority for Chief Information Officers (CIOs) in the last 10 years (Watson & Wixom, 2007). Even the creation of job profiles for Chief Information Officers started in the early 90s with the implementation of ERP

systems to ensure that the organizations make use of the potential of these systems (Earl, 1996). This long hiatus can be attributed to the complexity of business processes and lack of availability of data from all parts of the organization that can enable making a holistic decision. BI in this perspective is primarily used for strategic decision making by looking at data over a period of time by top-level managers in an organization. BI provides insights to managers and information officers to make more informed decisions. White (2005) classifies BI into 3 main parts, namely strategic, tactical and operational. These are mainly classified based on Business Focus, Primary users, Time-frame and metrics of data. The basic premise in classifying the different forms of BI comes from the level at which information is used and when the information is used. In the health sector, information needs to be made available to people at all levels. This information is required for operational activities such as what diseases are more prevalent, where additional drugs and workforce is required, how to make them available given the available resources. As mentioned earlier, this has been referred to as “use of health information for local action” (Stoops et al., 2003), where local practitioners can make use of information and adjust their work practices. With this conceptualization we see that Operational BI is used for the day-to-day activities of the users, who are information generators as well as information consumers at the same time. Keny and Chemburkar (2006) provide a slightly different conceptualization of Operational BI. They present the idea of granularity of information as the characteristic separating Operational BI from traditional BI. They suggest that while traditional BI relies on Key Performance Indicators (KPIs) to derive a holistic perspective on corporate performance, operational BI provides much more granularity to address the needs of

operational functions. This characteristic of Operational BI is similar to the concept of “hierarchy of standards” (Braa et al., 2007), where it is advocated that each level of the health system should be able to manage their own set of indicators, with increasing granularity as we go to the lower levels. The higher levels only need the aggregate view of indicators from the lower levels. Lungu et al. (2006) highlight the importance of BI tools in the building of EIS. They refer to EIS as systems that are designed to improve the quality of strategic level of management in organization and place BI tools at the centre of such systems. They also suggest that just having information for executives available is not enough. These need to be assistive and intelligent enough to support the work-practice of the users. Presentation of information is as important as the accessibility of the information and hence architectures of EIS should be created with BI tools at the centre. Today, even though data from organization wide operational processes are available in a data warehouse, most organizations only view operational data from their primary ERP system without any intelligence applied to the data to make it information (Imhoff, 2005). In the context of developing countries, this problem of not able to use information in intelligent ways is even more exaggerated. There are numerous other challenges that have been identified by implementers of BI tools. These include, but not limited to - assuring data quality, supporting complex conceptual data, integrating with other applications, support for real-time data, implementing security infrastructure (IDC, 2008). Similar challenges of data quality, integrating with other applications, availability to data, and accessibility of data are common problems faced by researchers implementing information systems in the developing world. Traditional BI has been complex to implement because it tries to capture all processes and business complexities. Thus, traditional BI implementations involve tricky data modelling and require highly experienced BI developers and data modellers. On the other hand, Kobielus (2009) suggests that Operational BI tools can be used as “Do-It-Yourself” business intelligence, where users themselves can configure analytical instruments like indicators, metadata, data sources and create “mashups” (a view of data from different sources) using graphical tools. In IS research, it has been identified that mashups will become the basis for Web 3.0, where user-driven programming of the web (Wong & Hong, 2007) happens, as users themselves connect the different pages of information on the internet and create new information. This common conceptualization of Operational BI and Web 3.0 through similar approaches is interesting for future research and direction of Operational BI.

As defined earlier, people, processes, procedures, tools, facilities, and technology (Pironti, 2006) are the actor-network components in an II. All of these have some form of enactment in the II. The term praxis can be used to describe this enactment. Sometimes this is also incorrectly understood to as agency, yet I do not want to get into

the larger and much debated critique of agency in the actor-network worldview. The word *Praxis* comes from the ancient Greek word (πρᾶξις). “Praxis is the manner in which we are engaged in the world and with others has its own insight or understanding prior to any explicit formulation of that understanding” (Schrag, 1997). The word has been deeply described in the philosophy of science since the time of Aristotle. He for example describes three types of knowledge: theoretical (*theoria*), to which the end goal was truth; poetical (*poesis*), to which the end goal was production; and practical (*praxis*), to which the end goal was action. I consider this third type of knowledge to study infrastructure and classify activities to be important, since the theoretical terms that we have seen to describe infrastructure, talk about enactment and actual practices. A number of philosophers have also used praxis to describe continuous improvements and transformation – reflection and action upon the world in order to transform it (Freire, 1985); agency embedded in a totality of multiple levels of interpenetrating, incompatible institutional arrangements (contradictions) (Seo & Creed, 2002); the key activities of individuals who have reconstructed their knowledge and brought organizational change (Spender, 1996). Thus, *Praxis* in many ways describes the *Bricolage*, which happens in IIs and the challenge of *reflexive standardization* or *workarounds* between global and local work practices. An interesting and important theoretical view though needs to be highlighted here. While, many authors and philosophers describe praxis through an individualistic notion - “human praxis”, there is limited attempt to group such praxis (plural) and even question if there is such a term as collective praxis (Roth & Lee, 2004). Yet many researchers have described *collective praxis* in education (Kemmis, 2010), sociology (Schulz, 1998), psychology (Cahill, 2007) etc. And this falls in line specifically with a lot of action-research studies that have been done in education, sociology and psychology, where actor-network way to thinking can be well recognized in their philosophical paradigm of collective praxis. Thus, collective praxis of the components, including the human and non-human networks in an II can be understood and studied, as has been shown by these other fields of science.

Part 2: AGP model of II activities

As we have seen, the three topics of Architecture, Governance and Praxis convey both the envisioned as well as practical activities that are needed in establishing IIs. These topics cover the wide range of theoretical concepts that have been used to study IIs and hence these topics cover the formative as well as reflective model of activities in an II. In the formative model, Architecture, Governance are visions by which an II needs to be established and Praxis includes that activities that are performed to enact the visions. Architecture and Governance do include activities that are performed to form visions (technical designs, process design, workflow, policies, laws etc.) and guide a way in which work needs to be done, yet they are philosophical different from the *bricolage*

that happens in the Praxis topic. Yet, the process of evolution of IIs and the range and contingencies of causal structures in its evolution has not been studied in depth before (Henfridsson & Bygstad, 2013). While these authors identify three generative mechanism or causal structures (Sayer, 1992) that have powers to instantiate events that bring evolutionary changes in the II, they do not highlight **where** and **when** (space-time) these generative mechanisms take place. They do not mention the topics under which the generative mechanisms of **Innovation**, **Adoption** and **Scaling** need to occur.

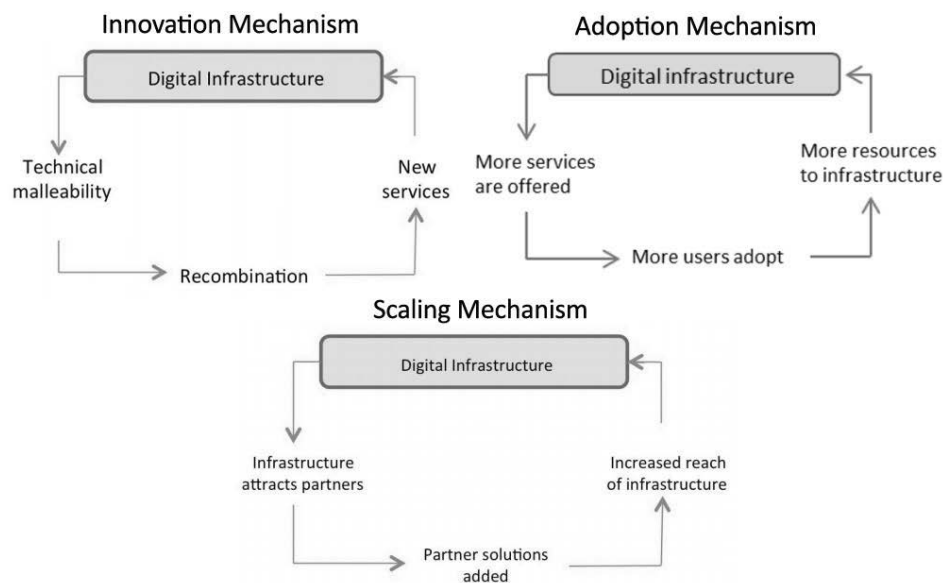


Figure 5: Generative mechanisms (Henfridsson & Bygstad, 2013)

Henfridsson & Bygstad (2013) highlight successful configurations that have established IIs by reviewing a fairly large body of II literature from an evolutionary perspective. They describe these configurations through the actualization/unactualization of the 3 generative mechanisms. So, in the first stage of **innovation mechanism** within an infrastructure, the causal power of technical *malleability* (sometimes they also refer to as infrastructural malleability) results in recombination within the infrastructure and results in new services that can be claimed as innovation. In the second stage of **adoption mechanism**, they observe that more services are offered, resulting in more users adopting the infrastructure and eventually resulting in more resources to be allotted to the infrastructure. The third stage, called **scaling mechanism**, attracts partners to the infrastructure and the partners add their own solutions to the infrastructure. Thus, the infrastructure due to these mechanism increases the reach of the infrastructure, not just in terms of what it can do (new services), what users it can

serve (new users) and what domains of use (new partners), but a combination of these. This authors describe the recombination as reasons for successful digital infrastructure.

Yet, here we should ask ourselves, if these mechanisms are peculiar to digital infrastructures. What is unique about the three generative mechanisms that they apply only to digital infrastructures? The authors do not argue this differentiation of digital infrastructure anywhere for us. The cases for analysis are from digital infrastructure and any claim that it can be applied to all infrastructures would have been quickly dismissed as faulty generalization or inductive fallacy. But let us think of these mechanisms in Hughes' (1983) Networks of Power and the electrification of western society. The **innovation** of the appliances that could be attached to these networks, helped users adopt new services on the power networks. The Alternating Current (AC standard) became in use because of the appliances that were available, resulting in more users continuing to **adopt** AC. And with more cities and localities wanting to be electrified, their industries, houses and in fact other infrastructures like roads, water lines were also getting **scaled** based on the electric power lines infrastructure that was being created. We can anecdotally apply the generative mechanisms to many other types of physical infrastructures. The reason why I raise the point of the need for specificity with regard to digital infrastructures is because I believe that modern-day digital infrastructures or IIs or more specifically leHIs are much more nuanced in their space-time contexts (where and when) compared to physical infrastructures, which can be more clearly articulated in space-time. For instance, email systems are inter-organizational IIs that are used by individuals across organizations, connecting different email service providers, with different locales and attachment types, different email clients working with multiple technical standards. Some of these standards are particularly designed to exchange interoperable date and time when email was sent. Other standards are designed to connect various domain names. Each component in the email system performs a number of activities that are relevant to space-time such that the activities can be translated by gateways that provide interoperability between the networks. Without knowing the space-time context, much of the translational gateways would not be able to perform their functions. In the same case, innovations in email systems like webmail, threaded conversations, embedded instant messaging have found their starting points in certain space-time description that is important to highlight. Without going into further detail with this case of emails, I'd like to suggest that the AGP model extends the generative mechanisms to study II evolution by adding space-time properties to them.

Henfridsson & Bygstad (2013) do study some contextual conditions of digital infrastructure using properties of architecture and mode of control. Yet, they do not provide clarity to what each of these contextual conditions deal with and what are the

activities that go into these contextual conditions. They also describe the contextual conditions through simplistic classification of architecture – loosely coupled or tightly coupled; and control – centralized or decentralized control. The AGP model can be considered an extension of the generative mechanisms for IIs in which the causality powers of each of the activities performed under one of the topics, manifests or do not manifest, but still have the powers of causality (Fleetwood 2009, p. 362-363) on the activities in the other two topical blocks. So, an architecture of either loosely coupled or tightly coupled components, open or closed system; governance of top-down or bottom-up policies, centralized or decentralized control; praxis of global or local, similar or dissimilar, planned or unplanned, intended or unintended etc. can be described and studied in more details based on the activities that are performed in each of the blocks.

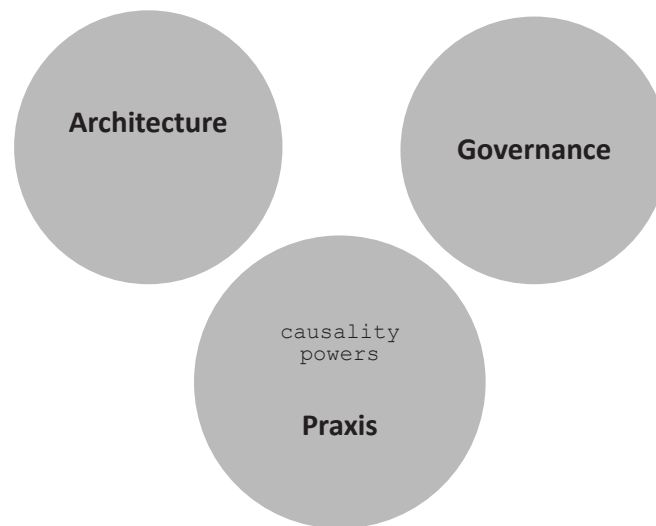


Figure 6: Initial understanding of the AGP Model of II activities

Table 4: Representing cases in the AGP model

Cases	Architecture (A)	A-G	Governance (G)	G-P	Praxis (P)	A-P
Case 1: e-Government in Germany: An II to support work processes connecting the state government located in the capital with the German Bundesrat (Pipek & Wulf, 2009; 2006; 1999)	- tightly coupled system - centralized file system - one groupware to paper and then another groupware	- self organizing groups for requirements - cooperative word processing in a specific ministry	- co-ordination between 3 ministries (SC) - top-down - centralized control	- appropriation and re-appropriation of groupware - POLITteam project ended and negotiations between SC, SRB and IT dept resulted in new groupware	- task shifts - process innovations - breakdowns and move back to paper - reintroduction of “vote preparation” process	- technical issues due to centralized architecture - overwriting of templates by users due to central file system architecture - communication and collaboration breakdown
Case 2: Health IS in India: the design, development and implementation of an HMIS tool (Titelstad, Staring, Braa, 2009; Sahay, Monteiro, Aanestad, 2009; Puri, Sahay, Lewis, 2009; Braa et al, 2007; Sahay & Walsham, 2006)	- Flexible web-based architecture - centralized or decentralized deployments - multi-level design - open-data model	- distributed software development - modular approach to local implementations	- co-ordinating node in Norway - local capacity building by hiring PhD students in Norway	- participatory HIS networks	- software developers and implementers travel for requirements - local capacity used for training - political reconfigurations	- changing politics results in removal of II - central node focuses on generativity vs local nodes focus on local needs
Case 3: International Financial Communication Network: SWIFT infrastructure used in over 200 countries (Scott & Zachariadis, 2014; 2012; 2010)	- common standards for financial transactions - open-architecture - heterogeneous nodes in network	- started as a closed network - moved to packet switching with increasing members	- distributed governance owned by member banks - negotiations between banks to add their bank identifiers	- from closed society to open network - common principles and laws for financial security	- community of practice - no change in internal practices of banks to join network - separate interaction between local and SWIFT transactions	- separate networks and practices between SWIFT and local banking - limited changes in organizational architecture for adding node to global SWIFT network

Case 4: Danish electronic patient records: SEP project expanded from local initiative in 2000 to national IeHI till 2009 (Aanestad & Jensen, 2011; SEP Report, 2003)	- modular architecture - open system and allowed multiple systems - heterogeneous information types - XML standards, central hub was designed and maintained by MedCom	- single company MedCom established VPN network for hospitals to connect	- distributed control - expanded from one organization - no deadlines for joining network, hospital's choice - national hub created by MedCom with regulation for security	- presented as supplement to B-EPR project instead of replacement	- practical solution - focus on clinical practice and not information exchange	- technically reliable for record exchange, but did not deal with local work routines - multiple changes to SEP specs and guidelines with each new connecting county
--	---	--	---	---	---	---

What is the AGP model?

Is it a framework that can be used for hypothesis testing? No, that will make it too positivistic in terms of scientific philosophy, whereas my view of the II phenomenon is mainly interpretivistic. Is it a conceptual framework more akin to social sciences that is a system of concepts, assumptions, expectations, beliefs and theories that supports and informs research (Miles & Huberman, 1994; Robson, 2011)? No, it is not a framework in these terms, since it is not as broad and encompassing. It also does not focus on adhering to a specific methodological (quantitative or qualitative), theoretical or epistemological belief like the generative mechanisms focus on being called critical realism. Also as we have seen from the literature review, the concepts come from a number of theories that are not always on the page in terms of their view of philosophy of science. Rather, the AGP model is an organizing technique, a taxonomy exercise that is the result of assimilation of bodies of literature. It does not extend a theory like extending it through a new concept, but rather a strategy to organize teams and activities when establishing infrastructures or retrospectively for organizing thought when studying IIs. Such objects that allow organizing thought, generally created through a literature review are commonly called research models that can provide reason and rigor to science (Ravitch & Riggan, 2011). So is it purely for research design (Maxwell, 2013) where it explains what is going on or what is going to be done, as part of the research? I would say much so, but with a small caveat. AGP model introduces a single conceptual guideline. I introduce this concept to describe sentient causality that exists between the events in an activity block and the causality powers. Sentient means, “the ability to feel, perceive or subjectively experience” and is distinguished from “reason” or rational objectivity. I refer to the causal powers in the activities of infrastructuring work to possess sentient causality because unlike Newton’s 3rd law of action-reaction force pairs, sentient causality is not deterministic. Beyond being non-deterministic, sentient causality is only a perception or subjective experience of the designer or performer of the activity in the II. Does the activity really make an effect to the II, or is it that the II is just supposed to change (by its nature) to accommodate and grow? You cannot claim that actions in one of the activity blocks will result in a known amount of predictable change in the other activity blocks. Causality powers may or may not manifest in each of the instance (Fleetwood 2009, p. 362-363), yet they do not directly show their existence (at least in realist terms) until they make a change in the other activity blocks. Thus, in the AGP model instead of questioning the existence of such powers in every action that happens, I rather suggest that they be studied to have simultaneous existence. To be more precise and practical, instead of asking how (and how much) the event was caused by an earlier event, we are more interested in the when and where the earlier event has happened. So with the AGP taxonomical lens, one is not as much so interested in the “equal and opposite” reaction, but more so in that the forces of action and reaction are acting on

“different objects”. The equal and opposite reactions to an action is only too widespread among the complexities of an II. Instead, the objects or as I have put it, components on which *infrastructuring* activities happen, are affected by these reactions in the taxonomical blocks of architecture, governance and praxis.

Sentient causality as Karma

Let me harp on the question if evolution of an II is inevitable, that it either reaches adoption and scale or it is not adopted and is shutdown. Is there a third path that the II goes on as a constant without any change? I do not think any II just stays as is. There is a constant of change, due to the activities that happen and misfits. Even if these activities might be considered routines by some, there are always users who want to find better ways to do their work or make a better fit between their work and work tools. Organizations themselves strive for improvement and efficiency that is never-ending. So all IIs are evolving and like we discussed earlier, the sentient causality plays an important role in evolution. But if it is perceived or subjective experience of the designer(s)/creator(s) of II or the researcher(s) studying II, it may also be ignored or overlooked. Similar to the unpacking of an assemblage (Latour, 2007) of actor-networks, this sentient causality can be packed or unpacked by the observer between the three activity blocks of Architecture, Governance and Praxis. In Buddhism and Hinduism, a term that is core to the philosophy of evolution is *Karma* (or *Kamma*). It comes from the Sanskrit root ‘kri’ which means action, affairs or activity (Mulla & Krishnan, 2008). The word *Karma* is often used in these philosophies together with the word Yoga as “Karma yoga” to describe action or renunciation of action to achieve salvation. I use the word *Karma* in its core root form and not with the connotations of Karma yoga. *Karma* is described as sentient causality by a number of researchers who study Buddhism. Tilak (1915) describes *Karma* as the perception of an individual for happiness or sadness, which is considered outcomes by the doer (actor) of the action and the doer relates these to good actions or bad actions respectively. Gandhi (1946) says about *Karma* - “We must become the change that we want to see”. Here he explains that we have to change ourselves by changing our perspective and only then we will be able to change the outside world (ibid). Again perception and perspective are important points with regard to *Karma*. These good or bad actions, for the designer/creator of the II are the ones that *stabilize* or *destabilize*, the establishing, adoption or scaling of an II. Yet, these are only perceptions based on what the actor consider as outcomes of the action. One can off course argue that the adoption and scaling are indeed measurable and “real”, then how do I claim these to be perceived? I consider these to be perceived, just as Buddhism does; because the next turn of events may actually highlight that these actions – adoption brought in users who want to take the system down (hackers) or scaling happened beyond the planned capacity and the system cracks down under its own

pressure. In these instances, the perceived good actions, in fact resulted in destabilizing the II and then the observer perceives these actions as bad. *Karma* has also been used to describe the wheel of life and death, where each *Karma* determines the re-incarnation of the qualities of an individual in next life (Kapleau, 1989). Consider this notion of *Karma*, in case 1 in Table 1. The breakdowns in communication and collaboration in POLITTeam v1 resulted in closing down of the project and a number of activities in Praxis block were causes to the “death” of that project. Later, these activities resulted in the SC, SRB and SC’s IT dept. to work together and give “birth” to a new groupware implementation for the same purposes, but by considering better fit to work practices and working with internet technologies. We can see that the *Karma*, or sentient causality that was perceived by the researchers, as powers that resulted in the death of one groupware, but the birth of another improved groupware. This “improved” is again only a perception and hence well described through the concept of *Karma*. Win (2008) provides a closer comparison to II by describing Karma as the *continuous causality* (complexity science) that can be seen in open systems. Thus, to summarize, *Karma* is perception, in fact time-limited perception of the causal powers of the activities in an II.

Representing II cases in the AGP model

Architectural block contains all the activities that relate to design, contextualization of the design, fitting practices and processes to design. The vice versa of fitting design to processes falls in between praxis and architecture because these are a bricolage of tussles between how the design is enacted and visions of the architecture. These activities in the overlapping areas are also the ones that have the most causality powers to cause events in other blocks. For example, look at the case 1, where the centralized file system architecture caused continuous problems to the practices of the users (Pipek & Wulf, 2009). They would make a copy of the template and start editing the copy, but at times this process would be lost and direct edits in the template would result in overwriting of the templates for everyone (Pipek & Wulf, 2006). Thus, this practice results in change from the centralized file system PILOTteam v1 to LINKWORKS groupware. There were task shifts, changes in work processes and occasional breakdown and move to paper for the “vote preparation”, the breakdown in the communication and collaboration architecture (like busy lines), caused the most disruption and changes in the architecture, with finally moving to an internet-based system (ibid). Governance block contains all the activities that relate to formulating laws, developing ways working, co-ordination or conveying decisions. Each of the activity blocks involve some decision making such as design decisions to contextualize or decisions to use a tool in unintended ways. But the decisions to manage and manipulate the entire infrastructure are supposed to be placed in the governance block. For example, look at case 2, where the central co-ordinating node in the University of Oslo

in Norway makes choices in what is generic and can be used by all the countries that use DHIS2 (Titlestad, Staring, Braa, 2009). But this centralized co-ordination node also has organized its development team as distributed software development. Developers and implementers often act as boundary objects, traveling between the implementation sites and move knowledge between the local country implementation and the software development that happens locally and software development that happens centrally. Such distributed development and modular approach to architecture that are placed in the A-G intersection, has large causal power, such as the change in requirements and new features that are under-development in DHIS2. The Praxis block contains all those activities that relate to enactment of visions, work processes, unintended consequences, workarounds and breakdowns. In case 3, we see that there is a community of practice around SWIFT infrastructure. The financial institutions that adopt SWIFT have member banks, who can have their existing independent practices and still become part of the SWIFT II. They just need to extend their endpoints to communicate with the SWIFT network.

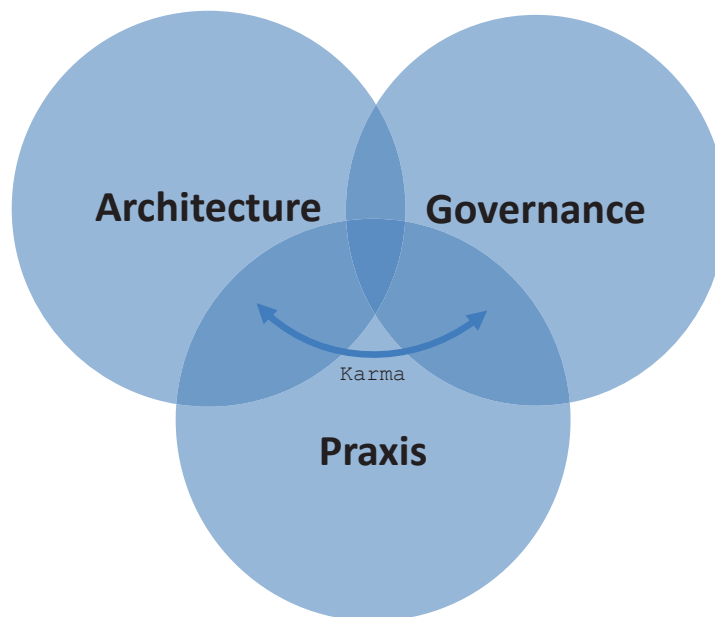


Figure 7: Final take on the AGP Model of II activities

The AGP model is applied to my research in the chapter 4 and chapter 5, where the establishment of leHI is brought under the AGP taxonomy.

Chapter 3: Context and Research Design

As mentioned in the introduction chapter, the work that is part of this thesis has been done in a number of low and middle income countries (LMICs). While that may seem to be the only site where infrastructuring work has been done, it is important to realize that large part of the software design and development happened in high income countries like Norway and United States among others. Since, this thesis is focused on the infrastructuring work that happens in building leHIs, I would like to shed some light on the context of my research.

I was recently in conversation with a noted realist working on biomedical ontologies. He was particularly upset with the use of the word “context” by researchers with interpretivist view because it means something so specific that only the interpretive researcher can experience it. I’ve realized from that conversation that context is a loaded word and it ranges from meaning everything to nothing, depending on who you talk to. The idea of this chapter is to be detailed and descriptive, as much as possible, what I mean by the context of my research. I use context as the social setting, situation or environment where I’ve conducted my research. As you might have seen from the collection of papers, my work can be summarized in the AGP model as well. From the paper summaries, you might have also realized that the papers describe wide range of settings or contexts. These are sometimes referred to as geographical regions, but what I really want to convey is the experiences of the project members, including me is here forth described as the context. So when I say India, Malawi or Bangladesh, it is supposed to mean my and the project’s physical presence in the country and the experiences by being part of the health system of that country. I will describe the setting of each of my study in some detail here, something that I may have missed in the individual papers, due to limitations of length, complex relationships of authority in the context and limited understanding at the time, when individual papers were being written. As time passed by and I was able to unpack the AGP model, that is also in some sense my research design, the context itself became clearer and this chapter brings that bigger understanding that the individual papers might be lacking.

The use of mobile devices, particularly phones and its use as a work tool has been of great interest to me. Before starting as a researcher, I was involved in mobile application development and was introduced to the University of Oslo research group due to my interests in mobile application development. The use of mobile technology in health services delivery is referred to by the term mHealth in literature. Mobile phones over the last several years have been the fastest growing information and communication technology in the history of mankind (ITU, 2012).

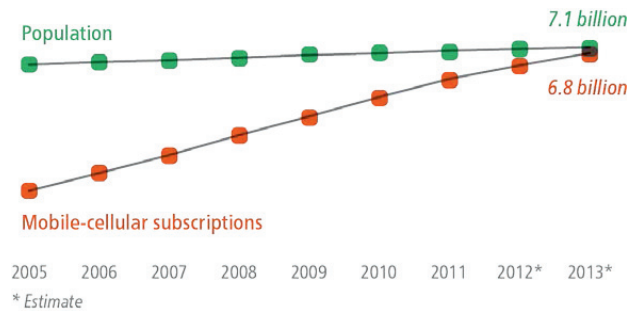


Figure 8: World population to mobile subscribers (ITU, 2011)

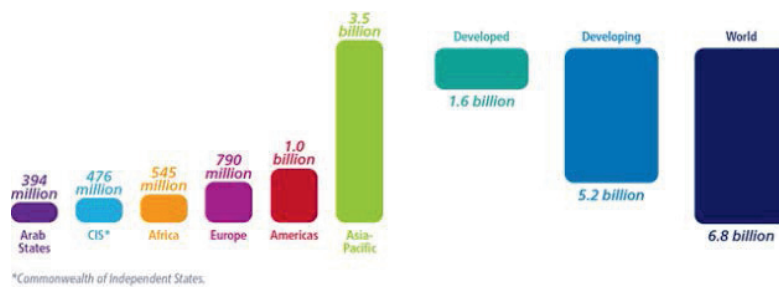


Figure 9: Mobile-cellular subscriptions worldwide (end of 2013) (ITU, 2013)

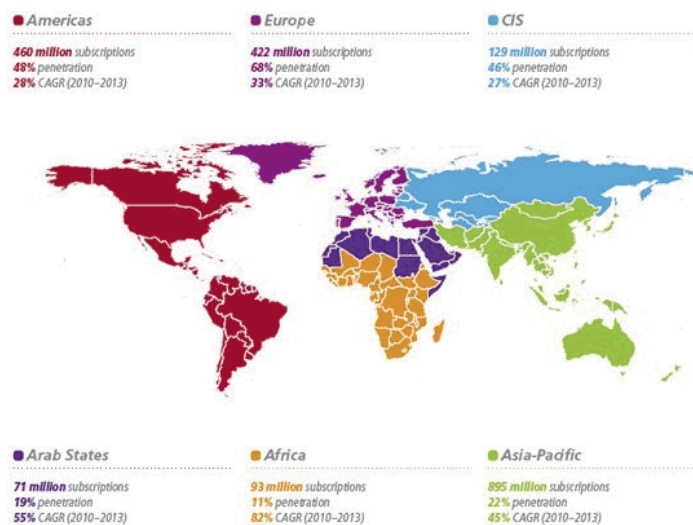


Figure 10: Global mobile phone subscriptions (ITU, 2013)

Particularly interesting with the rapid growth of mobile phones is that while the number of personal computers has stagnated or even decreased in the recent years, mobile devices have rapidly taken their place as the device to access the internet (ITU, 2013). Also

important to note, as seen in figure 7 and figure 8 is that LMICs have seen the highest growth of mobile phone users in the recent years. This large installed base of mobile phones has brought voice, data services, internet and applications in the hands of users.

This has hence opened up a large opportunity for countries to improve information access and communication for providing health services. This opportunity to capture data from the lowest level of the health system (as seen in figure. 2) and use it to gather more complete and timely data, brought me as a researcher to the HISP network.

The context of my research is within the HISP network sites, where researchers work side-by-side with people from health systems around the world to critically improve health using information systems. Health Information Systems Programme (HISP) is a research project initiated in post-apartheid South Africa in 1994, as a synergetic collaboration between public health activists from the anti-apartheid struggle and information system developers from the Scandinavian action-research tradition (Braa & Sahay, 2013). The 20 years' time-span of the research project exceeds traditional 'projects' and is more akin to social movements (Elden and Chisholm 1993). This network is called HISP due to the origins in the research project and comprises of researchers, developers, implementers, representatives from ministries who share knowledge and learning between the different nodes of the network. The design approach followed by HISP is with a goal to explore ways in which disadvantaged communities, regions and countries could appropriate ICTs for their own empowerment (Braa & Sahay, 2013). Participatory design and quick prototyping in the context of use, combined with training and building capacity at multiple levels of the health system becomes the basis for the development approach used to pursue this goal. The project is co-ordinated from the University of Oslo, but researchers like me and my supervisor are from NTNU, there are researchers from a number of different universities around the world, who work together in the HISP network. The main software platform through which health information management is accomplished is called the District Health Information Software, currently in its second iteration as a web application. Paper 5 and paper 6, of this thesis includes details about how the web application is used for information management and the innovative Business intelligence tools that have evolved over the years. I was recruited into the HISP network as Director of R&D at the Indian node, but later enrolled into the PhD program, as a research fellow at NTNU. My research fellowship was funded by the Norwegian Research Council in the grant for Global Health e-Infrastructures (#193023) under the VERDIKT program in 2010. The GHeI project had three PhD students at University of Oslo who have all passed out from the PhD program. My work should be seen as a continuation and complementary to the work of Johan Sæbø, Edem Kossi and Bob Jolliffe.

My work with mHealth began in India on a mobile application that is used for reporting aggregate health data by health workers to the ministry of health. It began as a pilot in 5 different states in India. While many initially questioned whether this should be

considered part of the information infrastructure, to me it is clearly the installed base that was enrolled in the leHI. Like mentioned earlier, mobile phones are already embedded in the social fabric of many LMIC countries (Toyama & Dias, 2008). The sub centre data registration and transmission (SCDRT), as it came to be called (see paper 1), was a proof-of-concept to see if middle-aged health workers from different parts of India, with challenging infrastructure such as low data network, limited electricity were able to use mobile phones for reporting their monthly facility aggregate data to higher level facilities. The proof-of-concept was as much as a technology pilot, as it was to see the usability and organizational capabilities of use of electronic information. The project selected 5 different regions to look at diversity of health workers, administrative differences, geographic differences like hilly areas, plains, lack of connectivity, electricity availability to see the feasibility of such a solution. It was among the first pilots that the ministry of health in India would be supporting, to evaluate efficacy of mobile phones in diverse health systems. The main principle that the mobile is part of a whole (Manda & Sanner, 2013), with strong emphasis on linking it with not only existing infrastructure of information systems (DHIS2 in this case), but also adopting design ideas to match the infrastructure, is the main finding of this research. It is documented and explained in detail in Paper 1. Though what has not been clearly articulated in the paper, which I can see clearly now due to coherence between the next few cases, is that this work has had larger consequences in not just DHIS2 project, but in the field of mHealth as well. The research established the fact across ministries of health in many countries that they should not allow mHealth projects to be separate, standalone projects without links to

existing health information systems. E.g. the ministry of health of Uganda in January 2011 issued a moratorium that all mHealth projects should stop if they cannot interoperate and work with the national-level health management information system (ICT Works, 2012). By this time in Uganda, Nigeria, Kenya, Zambia and few other countries, the mobile application that I had developed, was starting to be customized and implemented. Uganda at the time of this moratorium had over 150

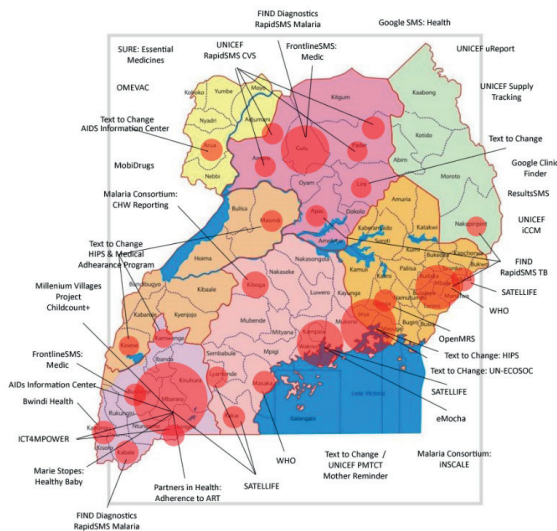


Figure 11: Uganda pilotitis (Sean Blaschke)

different mHealth pilots going on, referred to as mHealth pilotitis. Figure 9 from UNICEF Uganda is a representation of this disconnected situation of mHealth pilots, where multiple organizations wanted to build their own silos, without integrating with existing HIS. The case was similar in India and other LMICs, where the opportunity to use the installed base of mobile phones was seen by many, but not integrating the installed base with existing health system. The moratorium was for the first time that the ministry of health of an LMIC created a diktat of sorts to ensure an leHI. This is important period of change in my view because it was at around the same time that we were starting to implement in Malawi, a customized version of the tool that I had developed initially for India.

Going back to the story of the SCDRT pilot, we saw that the same aggregate reporting mobile application gathered interest in a number of countries. Nigeria was among the first to pilot the same application (Asangansi & Braa, 2010), Zambia followed soon after and then Tanzania and the list kept growing. Since, this was an open-source project from the onset, a number of software developers, implementers, and researchers in the HISP network started contributing to this application. Now it was more than just SCDRT, it came to be called DHIS-Mobile application. Different improvements such as offline storage, individual patient information, using data services instead of SMS, flexible data elements etc. all got built into the application. In Malawi, when we started implementation, we already had a fair bit of understanding of the mHealth projects in the country due to a colleague who was working in the HISP network, was part of this implementation. We worked closely with the ministry of health in two health areas to test out two different types of solutions. In India, for the pilot as well as the scaled implementation in some Indian states, it was always considered useful by the ministry of health that new mobile phones that would be used as “work phones” be given to health workers as part of this deployment. But in Malawi, this was not the case. The ministry due to resource constraints did want to provide new mobile phones. Based on our survey of existing devices with health workers in the two health areas, we realized that the mobile application would not run in most mobile phones due to diverse device capabilities. At this stage, we decided that one health area would use the mobile application while the other health area would use the mobile site access through the phones web browser. We provided a limited number of mobile phones to the health workers, but we also encouraged the health workers to use their own mobile phones by paying of their monthly data services bills up to a certain limit for reporting health data. This turned out to be another mixed approach to fit our application with local needs in the two health areas. We articulate this philosophical approach of the HISP network of prioritizing local needs over global standardization in Paper 2, using a post-development perspective. The critical aspect of the study by challenging existing notion of reaching

scale with a common solution is core to that research's contribution. Due to fuel shortage at that time in Malawi, the cost of travel, the risk of driving in rainy season with bad roads, limited electricity supply due to fuel shortage, all of this provided excellent demand for adoption of our project. But along with this demand, we also faced a lot of technology challenges in customization, limited local capacity causing breakdowns in the implementation of the mHealth solution (Matavire & Manda, 2014). These breakdowns and challenges resulted in some realignment of the architectural vision that we had worked with in India or other countries previously. This realignment process is better described by colleagues as Grafting (Sanner, Manda & Nielsen, 2014), as a way of infrastructuring, when breakdowns occur and changes are required to move an artefact from one context to another.

Just before the implementation in Malawi began, I was in parallel involved in the security certification of the DHIS2 software done by an implementing country's ministry of IT. Due to my previous industry experience of working on scalable and secure financial applications, I was entrusted by the HISP network to lead this security certification process. The security certification process is described in detail in Paper 3 of this thesis. An important point to highlight here is that being an open-source project, the source code for the DHIS2 project is worked on by a number of developers, but has a robust review process by core developers in Oslo. While this review process ensures that the code is technically sound, the process does not take into consideration whether the use-cases for which features are written are particularly secure. This was one of the reasons why the certification process found a number of issues with the DHIS2 software in the first round of testing. The first round of testing treated DHIS2 like a medical record system since it was considered a health information system. Instead we had to explain to the certification agency that DHIS2 is a management information system (MIS) used in the health sector, that too in places with limited internet connectivity, offline systems and through a number of different deployment strategies. Each deployment strategy had a different rationale and should be tested separately with separate security requirements and constraints. While the response to this was that the global OWASP testing standards were being followed by the certification agency, the researchers realized that the testing process and the design and development process of DHIS2, both needed to be more cognisant of the implementation requirements and challenges of inscribing insecure practices from a non-networked world in a networked world. Paper 3 articulates these security challenges. I would like to highlight another critical aspect of the changes that were being done to meet the certification requirements. These changes were not being done by the global DHIS2 team directly. They were reviewers to the code that was submitted to the core of DHIS2, but did not have an active role in the certification process directly. They received a summary of the documentation, but were

not part of the communications, negotiations and the rationale of explanation that were provided to the security certification agency. It is important to note that all of the changes made during the certification process did not get released as part of the global DHIS2 software. The country where the implementation was being certified also had a separate fork (codebase) of DHIS2 that made it challenging to merge changes between the central DHIS2 release and the code for local country implementation. While paper 3 does not highlight this important aspect of software development challenge in global software development (GSD), I've articulated this aspect in a much better way in Paper 4. This is when I realized that infrastructuring needs to be viewed not just as an implementation exercise, software design or development exercise or organizational use of an information system, but rather as a holistic infrastructure where changes in all of these aspects are correlated. I am lucky to be as much as a software developer, as a business analyst or an implementer of systems as part of my research. But this is an exception rather than a norm and thus what I see as a natural holistic view to infrastructural research might not be as clearly evident in existing bodies of literature. In their paper, Monteiro and Hanseth (1996) said something relevant about being specific about the technology, but I see that more lacking even in the current bodies of literature due to limited efforts of tracing all the three aspects that I highlight as necessary for the holistic view of an infrastructure.

Thus, as more time passed on, I began tracing the processes by which software development could be geared such that the end product meets with local needs. This is particularly challenging in open-source projects that are community driven and a distributed team of developers and implementers focusing on their local needs. In the countries where DHIS2 is implemented, we often see that there are a number of facility-based electronic medical records (EMR) systems. For example in Malawi, we saw the Baobab system, which was a fork of the OpenMRS data model, but instead of Java they had used Ruby on Rails due to availability of local resources competent in Ruby. This was an interesting fork because they continue to work separately from the OpenMRS community in the software development, but watched the changes closely and replicated it in their tools to be compatible with other EMR systems in Malawi that were using OpenMRS. My work during implementation of DHIS-Mobile in Malawi also allowed me to closely look at the Baobab system, interact with them about new changes in OpenMRS as well as answer their queries regarding how they could adopt the new changes in the OpenMRS data model. Since, I am also a core developer for OpenMRS, I was providing them with pathways to merge the code base. Thus, in the context of Malawi and other places, I see that OpenMRS is one of the EMR systems at the facility-level that is part of the leHI (see Figure 2 on pg.11). Here, at this point it is important to highlight a difference in the development model of DHIS2 and OpenMRS. Although both

the projects are open-source health information systems, DHIS2 is a much more centrally governed system and OpenMRS is a community governed platform. DHIS2 is made to be useful out-of-the-box and the customization that needs to be done is mainly at the content level. On the other hand, because of its practice-oriented nature, where workflows need to be customized, OpenMRS needs to be customized at the content as well as the workflow level. Thus, new modules are often written in OpenMRS by community of implementers for their specific needs instead of a global, generic need. On the other hand DHIS2 core team tries to take as much of the generic workflow into the main release of DHIS2 and thus, the decision making process of the features that are incorporated as part of each version release is centrally governed. In open-source projects, OpenMRS is much more of a norm than the DHIS2 governance model (Raymond, 1999; Fitzgerald, 2006). Thus in paper 4, I use OpenMRS as the test case of observing changes in the development methodology, so that the software releases can be more suited to meet local needs. Here, working alongside the community I was able to observe and be part of the changes in the governance model of the software development process. The OpenMRS community moved from a distributed global software development process to an agile software development (ASD) methodology. The process of change took about 2 years to completely change into the new methodology, first starting from the core software development and later including core modules and then the community developed modules. The changes highlighted a new way of ASD that is peculiar to the way open-source communities' work and hence the customizations to the methodology helped make the community more attentive to the requirements from the field.

In parallel to the study of software contextualization and fitting to use, we were also implementing DHIS2 in a number of countries in Africa and Asia. In earlier instances, although DHIS2 was a web application, its deployment was rarely expected to be a completely online system. The main argument against deploying a fully online system was that districts office in the health system did not have internet coverage in African and Asian countries. Kenya was among the first countries to deploy DHIS2 as a fully online system where data from health facilities was directly entered into DHIS2. One of the main reasons supporting this data entry was that the mobile internet coverage had spread to villages and began covering all of the country. This deep penetration of mobile networks, which I first encountered during the mHealth projects was again seen as the substrate on which leHIs could work and replace paper as the medium of information flow throughout the infrastructure. The leHIs were able to enrol the "last-mile" as it is called in popular media, into the network. Data started coming in directly from health workers at the facilities. Business intelligence (BI) tools that were earlier available at the district-level administrators and handed down to health workers at facilities, were now

also available to the health workers at the facilities. Some changes in the BI tools became necessary due this additional inflow of users and analytics needs of these users. I've documented some of these changes in the BI tools in Paper 5. In this study, I wanted to understand how the countries that were deploying data warehouses with BI tools as the part of the leHIs were able to make sense of data. To put it in another way, I was back to asking the same question, what did "success" mean to the implementers and users of large scale warehousing system? Although we saw in the mHealth projects that "success" was in meeting local needs and then continuously meeting different local needs at different places to reach scale, we did not know at a higher-level in the leHI, what this "success" meant since it wasn't really local anymore. To understand this notion of success, we tried to find out the factors and the correlation between the factors that made "information for action" possible in large datasets. We realized that most users were calling this "Big data" for lack of a better word or the buzz word that has been created around the term of large interconnected datasets. In paper 5, we describe the different countries in which we surveyed implementers and users of DHIS2 software and BI tools that are useful to the users. We correlated the organizational capabilities framework (Gold et al., 2001) with the data warehousing success factors (Wixom & Watson, 2007) to find out what were the factors that enabled users to make successful use of leHI's data. While the paper is written with a practitioner focus, we have done due diligence for it to be rigorous research. We had to skip some of the important details of research methodology and academic writing due to the paper's practitioner focus. Those details are mentioned in the later section of this chapter, when I discuss the methodology of each paper in little more detail.

While we found the correlating factors and came up with the Overview-Overwhelm framework in paper 5, we were not satisfied with our findings related to building and improving organizational capabilities in that paper. So even though we were able to describe the factors that enabled "information for action", we were unable to see the ways in which the organizational capabilities could be improved within a health system. Could there be ways in which capabilities could be gained or purchased or outsourced from other places or external organizations? Could there be models where capabilities that were missing in the health system could be found in other organizations and could those be used in bootstrapping the infrastructure? Cloud computing enables business models where computing is provided as a utility by other providers that are external to the organization. Computing becomes a utility similar to water, electricity etc. that is used without having the capabilities to implement a whole pipeline system or an electric system. Similarly, cloud computing models, such as Infrastructure-as-a-service, Platform-as-a-service, Software-as-a-service and Analytics-as-a-service, we were able to describe ways in which the DHIS2 software can be deployed to support the

organizational capabilities of health systems from the outside. We surveyed the tools that were available, the different deployment strategies that companies and businesses in the DHIS2 ecosystem were thinking and what researchers in the HISP network had encountered. We also surveyed the core DHIS2 developers to understand the future directions in which they thought implementers could use capabilities of vendors to improve the deployment and use of DHIS2. We document this in Paper 6 and suggest innovative business models of cloud computing that can be used for DHIS2 deployments. The paper 5 and paper 6 has been co-authored with Prof. Jørn Braa, who founded the HISP network along with others in South Africa and has provided a historical, evolutionary and multi-site perspective in these papers. The context of these papers, hence cover much more than just the period where I have been involved in the project. We also used different methodology in these papers, compared to what has been traditionally interpretative research in the HISP network. The papers use descriptive documents, as well as surveys and interviews as data and we have analysed those using mixed methods, which range from PLS regression techniques to realize correlations between the independent variables, to interview coding to detect common concepts. In the next section, I mention the research goals for each study and rationale for using the type of methods and research design.

Research Goal

As you might have realized by now, the research has been dynamic in nature, observing the effects of actions that have been taken by researchers in the HISP network. This has been referred to as Scandinavian action-research tradition, but more formalized as “Networks of action” (Braa et al., 2004) methodology. The research goal thus broadly is to understand the consequences of the actions to create leHIs in different LMICs. As highlighted in chapter 2, these actions have been referred to as infrastructuring work. My research goal was to create a way in which infrastructuring work can be classified, so as to be better understood and organized phenomenon.

Each of the 6 studies form a coherence, which they are part of the infrastructuring work that is done to create leHIs. They look at different parts of the work, at different points in time, but with a realization that snapshots in time are rarely holistic views. Rather evolutionary study helps explain the sentient causality or Karma in the infrastructuring work. Since leHIs have been attempted and have faced challenges in high-income countries, there is no reason to believe that they will work better without planning and understanding the phenomenon in LMICs. So like all action-research, the goal is to improve the action of infrastructuring and do the necessary contemplation on the observations as research. I would like to repeat the research questions here again to

highlight the commonality between the research goals and questions that I've attempted to answer through this thesis.

RQ1: Given attention to the ongoing efforts of developing infrastructures, how can activities in an IeHI be classified using a taxonomy?

RQ2: What are the blind spots created through this taxonomy and how do they affect the infrastructure evolution?

Research Process

In this section, I have describe the research goals/objectives of each study. The six studies are described based on what we went out to achieve and how the research was conducted. Details about the methodology that are missing from the papers have been included for each paper here to elaborate and explain the reasons for using the methodology.

Study 1: What are some design strategies for scalable mHealth solutions?

As described in the earlier part of this chapter, the *SCDRT* mobile project has been developed through Scandinavian action research tradition in IS development, such as user participation, evolutionary approaches and prototyping (Sandberg, 1985; Bjerknes et al., 1987; Greenbaum & Kyng, 1991). The research is part of the HISP network of action researchers, who aim to create knowledge by taking part in the full cycle of action-research ranging from design, development, implementation, use and analysis and continuously iterating through these cycles. I was part a team of researchers from the HISP network to pilot an mHealth application in five districts in 5 different states of India. This pilot study and its finding have been documented in another paper (Mukherjee, Purkayastha & Sahay, 2010). This application then scaled across the whole state of Punjab and then to many other countries (with modifications off course). The action-research steps are done together with all the involved actors such government of India/state of Punjab, mobile phone operators, handset distributors and health workers. Before every action-research cycle, the interventions are adjusted accordingly, and the next cycle begins again (Susman & Evered, 1978). The book chapter that is part of this thesis is a result of the participation in this action research network for the last 3 years and provides insights from the central role in design of the mHealth application and participatory role in implementation and maintenance of the system after implementation. The research has been done within the framework of interpretive research (Walsham, 1995). Data was collected through group discussions, requirement meetings, and developer discussions, feedback reports along with one-to-one interviews in health workers, health officers, doctors and ministers.

The first phase of the research for this chapter was started as part of the pilot and gathering requirements by involving the state health department and their officers through interviews and group discussions. After the requirements gathering phase, prototype demonstration were made to the national and state ministries of health working under the National Rural Health Mission (NRHM) programme. The feedback from these demonstrations were recorded in meeting notes. These meeting notes were analysed by the team of researchers for concepts that need to be included in the research design and interview constructs. These meeting notes have also helped in improving the software and served as further requirements for the next iterations. The next phases involved development of the software, which have been done using Agile Methodology of software development. Regular iterations and emails to the developer mailing lists have served as data to interpret the development of the software and project as a whole. The next phase of the research was my close involvement in user training and after the user training, recording feedback on a set of questions from the health workers. These training were conducted during the pilot ranging over 30 days of field training with health workers in 5 different states in India. Each training day would last anywhere between 5-7hrs along with meetings with local health officials, administrators and health workers. I was also involved in the setting up of mobile phones, advising purchasing decisions made by the health departments and negotiating with mobile operators and plans. This deep involvement is of essence for this research, to become active participants in the implementation of the system along with the members of the health system. The data collected has been analysed and quantitatively represented in the report given to the ministries of each state, using which they have used in monitoring the progress and usefulness of the application. The research covers involvement in customization and training of the application and the data interpretation involving documents from implementations, health system manuals and being part of meetings. The research for Kenyan case that is also part of the book chapter, has been done over a period of 6 months, by looking at the documents produced by the designers of the system and communicating with the field workers of that project over email and Skype. The involvement with the Kenyan case is extremely limited compared to the deep involvement in the Indian project. But it is used as a reinforcing case, about the concept of using existing infrastructure and making design decisions to use/reuse the installed base. The data interpretation for this case is through ex-post-facto observations of the implementation and documents made available by the implementers of that project.

Study 2: How should researchers evaluate “success” for mHealth implementations?

My involvement in the mHealth implementation in Punjab did not result in complete evaluation after the deployment and use of the system due to a number of reasons,

hence that case is not used for evaluation. Rather, I was involved in the implementation of the same mHealth application in Malawi. The implementation in Malawi mainly comprised of four researchers from the HISP network and local partners from the ministry of health in the two health areas where we focused our implementation efforts. The main activities of the researchers consisted of developing free and open-source software systems (FOSS) including but not limited to this mobile application, implementing them in conjunction with local partners, capacity building and improving the system through analysis of implementation activities. The mHealth project is based on the DHIS-Mobile application that was first developed in India and then modifications to it were made by globally distributed software developers. The DHIS-Mobile is tightly linked to the global DHIS2 project and aims to share learning between the different nodes of the network.

The 3 authors of this paper are particularly involved in the implementation of mHealth applications in partnership with the Ministry of Health, Malawi and were involved in the pilot of the DHIS-Mobile in 2 health areas in Lilongwe, Malawi. Particularly, two different approaches were planned based on the initial analysis and discussions with the ministry of health – a JavaME mobile application and a mobile browser based DHIS2 interface. The research is conducted as Critical Action Research as described by researchers based on Habermas (1987) and developed further by researchers like Kemmis (2001) and Carr and Kemmis (2005). Our research is also guided by the *network of action* approach. This approach is an addendum to the idea that local Health Information Systems research can be made more robust and sustainable by being part of a larger network and sharing experiences between the different nodes of the network. The main challenges of the project centred around two issues. Firstly, the limited technology capabilities of the mobile network operators in Malawi and secondly in the configuration of imported mobile phones to connect with the existing network capabilities in Malawi. Although this customization was technologically easy, due to limited skills in the personnel at the mobile operator, the project took longer than planned for deployment. More details of these challenges, described as breakdowns is part of a colleague's publication (Matavire & Manda, 2014). Key informants for our study include medical personnel, health surveillance assistants, and statistical clerks, from all 17 health facilities that are part of the pilots. Training sessions we conducted for would-be users on the solutions under pilot involving three stages. Firstly, we conducted focus group discussions, with participants, covering topics such as existing paper-centric routine health data collection and reporting practices, data use at health facility level. There were a total of 4 focused group discussions, each of about 2hrs during the initial phase of the project. These were recorded and later transcribed by the researchers. In these the group discussed what sort of feedback health facilities get from the District Health Office, if any, on monthly reports they submit. Secondly, we had hands-on training on the DHIS Mobile solutions

under pilot. These were 4-5hr training sessions covering datasets, data elements that are captured and the meaning of the data elements and how these have to be calculated from health facility registers. Finally the training covered using the application and how to report data to the different levels of the health hierarchy. The third part of the training was a feedback session on all matters covered during the training. The feedback session was a combination of two things. The first is a survey form about the usability of the application. The second is a set of open-ended questions on the issues, improvements that should be made in the software. This was done through another round of discussions and completion of pre-designed feedback forms. We've conducted focused group discussions and interviews with 22 community health workers, 2 health facility managers and 2 district-level health department officials. Every iteration of the mHealth application involves feedback and critical analysis of the data collected through the interviews and changes are made to the application based on the feedback. For the purpose of understanding the context, we've collaborated with researchers working in other health information-related projects in Malawi, local master students from the Chancellor College of the University of Malawi. One of the authors of the paper has also been involved in review of mHealth projects in Malawi and understands the local culture, context and language of Malawi.

Study 3: What software development methodology can be used to increase user participation in open-source communities?

While the mHealth implementation in Malawi made us realize the importance of understanding the local context and the contextual needs, we wanted to understand ways in which the software development process can become more aligned to the user needs. Participation of users in the design, development and implementation of software was understood to be necessity. Thus, based on an open-source community that is part of the leHI, I studied the methodology by which user participation in the software development process could be increased. Although user participation started as the core need of the project, it was quickly realized that beyond user participation, developer participation and knowledge sharing became important criteria by which health information systems can be scaled and made sustainable in meeting user needs. The research followed a case study methodology to understand the effects of agile software development methods in its natural context. Software development in the OpenMRS community is done by using online project management tools. The Case study method is useful to study post-facto effects, where theory and research are in their formative stages. The research employs a mixed method approach. Initially using quantitative methods and interpreting results. Later, using qualitative methods of group discussion, semi-structured interviews with interpretive analysis by coding these into terms and concepts used to describe Agility and Knowledge sharing. Data collection was

done by querying the issue tracking system (JIRA). Individual work units in JIRA is henceforth referred to as tickets. We analysed emails from mailing lists (developer [n=18318]; implementer [n=8316]) and source-code, covering the period from January 2009 to January 2013. Over 3000 tickets were analysed for factors such as assignee, reporter, priority, creation time to resolution time, linkage to source-code and linkage to a sprint or software release. This was done through the use of JQL queries that allow retrieving issues based on selective options from JIRA. Source-code was analysed in correlation to the tickets and measured according to the changes in lines of code per developer, number of commits, refactoring of existing code, unit tests and code comments. The research covers code from OpenMRS repository in subversion as well as git. Migration of code from subversion to git happened in August 2012. An Ohloh.net (an online project and code analysis tool) project was created for code analysis by listing various code locations. Additionally, a tool called Fisheye from Atlassian Inc. was used for analysing activity by developer in terms of code commits and code reviews. Nabble.com was used to get aggregate information about individual contributors on the mailing list. Text mining was not done on the contents of the mailing list, but analysis was done only on the name, email and known organization from the sender's list. Documents on wiki pages which describe design, development and use were analysed through an interpretive perspective. The wiki is used to collect summary information about discussions and often as a knowledge base about design decisions taken by the community. IRC logs were analysed for the number of active participants in the IRC, as well as the number of lines of communication was collected to measure the activity in the IRC, similar to that of the mailing list. The mailing list was used to differentiate between developers and implementers. Individuals who have more than 10 emails to the developers list are identified as developers, where as individuals who have more than 5 emails to the implementers list are identified as implementers.

This quantitative data was interpreted in relation to the different concepts of agility as presented in the previous sections. This analysis was then shared with each individual core developer through a set of semi-structured interviews which last about 45min to 1hr. A total of 25hrs of interviews were done and 3 group discussions were organized with the core developers. The interviews were transcribed and entered into Nvivo, qualitative data analysis software. Then performed coding based on concepts of "learning", "agility", "knowledge", "release cycle", "participation" and performed thematic synthesis. The resulting themes from the analysis were matched against quantifying words like "more", "less", "increase", "decrease" to verify that the interviewees described the concepts across the interview in the same increasing or decreasing order. Beyond discussing interpretations of quantitative data, opinions were asked on a wide variety of topics such as community participation, developer workload, project management and software development methods in the OpenMRS community.

This resulted in deeper understanding of the phenomenon and allowed drawing upon interpretations of core developers. These discussions helped meet the principles of interpretive research such as - principle of contextualization; principle of interaction between researcher and subjects; principle of dialogical reasoning; principle of multiple interpretations - each of which helps bring rigor and validity to the findings.

As in any research approach, case study has its strengths and weaknesses. Case research is important for this type of research, as it allows for study of a large number of variables in a given setting, while these variables do not have to be previously defined. The weakness of such case research is that it is hard to make generalizations or be able to draw conclusions that can be claimed to be valid for all open-source projects. But I take the view that OpenMRS is indeed representative of many similar open-source software communities that work in a vertical domain and have a similar governance and participation model. The OpenMRS governance model is community-driven. Issues are created by community members, weekly developer meetings, weekly implementer meetings, design discussions are on public mailing list or during the weekly meetings. Code review happens in public, voting is used to prioritize features etc. There is a newly formed OpenMRS Foundation with an executive board and community members vote to put a member on the board of directors. Most day-to-day decisions are not taken by the board, but instead through community discussions. The leadership of the OpenMRS community has tried to model itself similar to Mozilla, including having the ex-CEO of Mozilla on the OpenMRS board to get a better understanding of governance principles. In the paper, I attempt to contextualize ASD in OpenMRS as much as possible. Krutchen (2013) highlighted the importance of contextualizing. However, due to length constraints, we don't describe the context of ASD using the full "frog and octopus" model, but make maximum attempt to describe all areas, although not as separate sections. OpenMRS is a software platform and a reference application which enables design of a customized medical records system with no programming knowledge (although medical and systems analysis knowledge is required). It has a modular design, where modules are add-ons that extend the functional scope of the system. There are 76 modules installable from the OpenMRS module repository, 125 modules have their source-code in OpenMRS svn. While there are close to 220 OpenMRS modules that are openly available from different sources (github, bitbucket, sourceforge), yet this is only a rough estimate of available modules. Most modules are developed by developers who are not part of the core team. These modules cover broad range of functionality and there is a clear separation of openmrs-core, which has distinct software development lifecycle from modules. While the focus of this research is openmrs-core, we include some modules which are distributed along with the reference application called core & bundled modules. These include FormEntry, HTMLFormEntry, Logic, XForms, DataEntryStatistics, SerializationXStream, Reporting, ReportingCompatibility,

HTMLWidgets and PatientFlags. Unless mentioned otherwise, the paper refers to OpenMRS as the “core + distributed modules”.

I have been involved in the project as an independent developer for about 6 years, without direct funding from any organization to be part of the software development process. I have also spent a summer internship through Google Inc. at OpenMRS in 2008, through which closer engagement in the community had started. I have been identified as a contributor to the core for many years and have been actively engaged in different roles - as developer, implementer and consultant at for-profit and not-for-profit entities that use the OpenMRS platform. I have participated in many design discussions, roadmap decisions and overall community management discussions before this research. Over the years, I have developed few open-source modules that are used by implementations all over the world as well as proprietary modules that are used by for-profit and not-for-profit global organizations. All of this highlights that I already had a deep understanding of the community and its practices (implicit and explicit) including roles of core developers and other community members. Walsham (2007) classifies such style of involvement as “involved researcher” while doing interpretive research. Yet, the motivation and the decision-making process of changing to agile method, (specifically a customized Scrum method) from a global, distributed software development model were not known to me clearly before this research. This is because the decision was taken by the OpenMRS leadership group and was announced to the community through the developer mailing list. More on the motivation and decision-making process for adoption of agile methodology is covered in the next section.

Study 4: How do security challenges arise due contextual software development?

While the deployment of mHealth project in Malawi was starting, I was involved in the security certification of an HMIS system. I use the pseudonyms, *HMIS system* and the implementation country as *AFIN*, as is the standard in information security papers to prevent malicious use of research. The HMIS system is an open-source management information system used in health systems in more than 30 countries around the world. The software is developed through Scandinavian action research tradition in IS development, such as user participation, evolutionary approaches and prototyping. The above mentioned steps are done together with all the involved parties before the interventions are adjusted accordingly, and the next cycle begins again. I was involved in all the phases of research mentioned above, particularly in *AFIN*. I participated in the action-research for the last 3 years through a non-profit organization, which has been involved in the implementation of the *HMIS system* for more than a decade now. I played a central role in the process of security certification of the software, done by the Ministry of IT of *AFIN*. The research has been done within the framework of interpretive

research. Data was collected through different channels of communication with global community and local developers of the *HMIS system* in *AFIN* on one side and the security testing agency on the other side. I was involved in the customization and training of the *HMIS system* and the interpretation involves documents from implementations, manuals and being part of meetings to customize the system in different states in *AFIN*. The research for this particular study was conducted over a period of 2 years, where more than a year was spent in customization and development of the system and about 8-months as part of the certification process. Along with the security testing, functional testing and performance testing was also simultaneously conducted for this *HMIS system*. I was involved in the testing processes as part of a larger team of developers from the non-profit organization. Most of the data collected from functional testing and performance testing is not part of this paper, but that data has provided me with insight in interpretation of the observed phenomenon. A large fee was charged by the Ministry of IT to perform this certification and was an important factor for the system to be implemented on government infrastructure. I was employed by the non-profit organization during the period of research, but in between I made transition to start working on my PhD at NTNU. The *HMIS system* has been designed, developed and implemented in developing countries around the world. The action research project that is at the core of this *HMIS system* is used for research in developing countries in Asia and Africa. This system is thus built around the idea of supporting health systems in these developing countries. The system has been in use over 30 different countries around the world, sometimes as pilots or district-sized implementation, but also has been implemented as country-wide health management information system. In *AFIN*, it has been used by a number of states, but is not implemented as the national information system. Nevertheless, these state-wide implementations are web-based systems that can be accessed over the internet and there are separate implementations for each state. There are anywhere between 100-5000 facilities that report data into these systems and thousands of users in each state using this system. Thus, in *AFIN*, the system can be called as a large-scale web application.

Before the use of electronic systems and computers, the health system in *AFIN* has primarily made use of paper forms to report data from health facilities. The health system continues to be a hybrid of paper and electronic medium for data collection and transmission. These paper reports are created by community health workers who work at facilities and provide health services to the community. The community health worker maintains registers of patients and the services provided to them. These registers are classified separately, based on the type of health program or services offered by the health worker. There is on an average 20 different registers at the lowest level of health facility. Thus, the patient record is created by the community health worker and is available at the facility in which the health worker has provided services to a person.

This data from the registers is then manually aggregated by the health worker every month and reported according to a standardized facility form and its data elements. The aggregated reports are submitted to the higher level, which then aggregates all the reports received from health workers to create another form. This form is then sent higher and the higher level aggregates and sends it higher. This hierarchical chain ensures that all data from lower levels are seen by the higher levels and the higher levels can allocate resources to the lower levels.

The *HMIS system* has been designed to mimic the organizational hierarchy of health system and data can be entered by the respective facilities/organizational units by logging onto the internet-hosted web application. The users from an organization unit are able to see data from their own unit and all the units below them. There are access control rules that can be created in the system, which allows the administrator of the system to limit the levels/datasets that is available to any user. The *HMIS system* is also designed to allow flexible methods of aggregation when viewing data at higher levels. There are many useful analysis and reporting tools that can be used the facilities to manage their own data and analyse it for their own activities. In the patient module of the *HMIS system*, the patient record can be opened by any facility from the organizational unit hierarchy. This is because in the context of *AFIN*, migration is a common phenomenon. As an example, it is the social norm in *AFIN* that after a woman gets pregnant; she goes from her in-laws' house to her parent's house for delivery of the baby. The previous treatment received by the woman needs to be available at the other facility where the woman has migrated, so that continuity of care can be provided. At the time of writing of this paper, the *HMIS system* can only deal with migrations that happen in the same state, i.e. if the organization unit to which the patient has migrated is in the same implementation of the system. This means that exchange of records across different installations of the system is not possible and only patient migration in the same deployment of the *HMIS system* is possible. This is an important distinction to understand, so that we understand that patient records are technically not exchanged between systems, but only accessed by other users from the same database.

In *AFIN*, not all the health facilities have access to internet and many do not have a computer available. At such places, the *HMIS system* is deployed at a closest facility where computer is available, but due to lack of internet, offline installations of the system are also made. These offline installations export data to the central online system via USB sticks which are imported into the online system from computers located at another village/town where internet access is available. This combination of online and offline systems is a reality in most countries where the system is implemented and is an important characteristic of the context. The offline installation is exactly the same application as the online application. Only that they are not part of the internet and are restricted in use only by the facility in which it has been installed. Thus, there is a hybrid

model of deployment in offline and online modes. With respect to the patient module, in cases where there is no internet or computer available, the whole patient record on paper moves to the higher location where computer with internet access is available. This patient record may be entered into the *HMIS system* by the health worker themselves or by data entry operators who are hired at district offices because the health workers generally do not have enough computer skills. This is an important property of the context because it means that the data entry operator acts as a proxy between the health worker and the system.

The software is customized according to the requirements of each implementation. In *AFIN*, this means that every state has different data elements, organization units, reports. There are new features that are requested by these states very often and local developers are involved in customization of the *HMIS system* for the state. These local developments are then analysed by a global team of software developers and after negotiations are made part of the central application that is available for use in other countries.

Study 5: How do warehousing and BI tools evolve to make a health system use big data?

Since DHIS2 was deployed as a web application in a number of countries, its use and capabilities have scaled enormously. It has been referred to as the largest open-source health information system (Weber, 2012). Information for action is a term used to refer to the use of data for taking decisions. The BI tools that enable data analysis for deriving actionable information is described as part of this study. The study was conducted in two phases. The first one was to do a survey of implementers and users of the system in different countries to create a correlation between the factors that determined successful use of BI tools and the organizational capabilities to use those tools. This was important to understand, so that we could explain the second part of the study, which highlights the evolution of the different BI tools in DHIS2. The DHIS2 software moved from a MS Access-based desktop application to a web application.

The survey was done online by sending forms to implementers and different ministries of health. The ministry of health nominated their representatives as respondents for the survey in 30 countries where DHIS2 is implemented. The survey was taken from the Watson & Wixom (2007) study to understand data warehouse implementation success and we integrated it with the organizational capabilities framework (Gold, Segars, Malhotra, 2001). The survey questionnaire was circulated among the HISP researchers to verify that the questions were appropriate. Based on the feedback, some changes were made to reframe the questions and provide additional explanation to each question and its likert-scale. The survey received 119 responses and all received

responses were complete, with all the questions answered. The following is the questionnaire survey that was used for the study:

Country for which you are completing the survey *

Name of the country

1. Information champions are found at the following organizational levels:

Information Champions are people who are proactive in collection, reporting and use of data

Level1 (Community) Level2 (Facility) Level 3 (District) Level4 (Region) Level 5 (Ministry)

2. Information Champions are generally from the field of:

What would be their likely background (they might be a combination, but highest expertise)

• Information Systems	Statistics	Health/Medical Sciences				
3. The Project was Adequately Funded		1	2	3	4	5
Resources available to the project in terms of money						
4. The Project had enough team members		1	2	3	4	5
The no. of people on the implementation/decision making						
5. The Project had enough time		1	2	3	4	5
The time available for implementation of the project						
6. Implementers and users worked together		1	2	3	4	5
Participation level						
7. Users were assigned tasks by implementers		1	2	3	4	5
Users are given tasks for implementation activities (e.g. customization task)						
8. Users perform tasks and continue to work on their own		1	2	3	4	5
User independence level						
9. Common data/indicator definitions existed		1	2	3	4	5
Form standardization level						
10. Data sources/Programs were diverse		1	2	3	4	5
Fragmentation level						
11. Significant standardization activities		1	2	3	4	5
Standardization effort level required during DHIS2 implementation e.g. revisions of reporting forms						
12. Adequate protection of data		1	2	3	4	5
Data security/privacy level						
13. Adequate technological infrastructure was available		1	2	3	4	5
Technology infrastructure availability						
14. Implementers had adequate technical skills		1	2	3	4	5
Implementers technical level						
15. Implementers have good interpersonal skill		1	2	3	4	5
Interpersonal skill level						
16. Overall management was encouraging		1	2	3	4	5
Management encouragement level (MoH or other top-level management)						
17. User satisfaction has been a major concern		1	2	3	4	5
Satisfaction level						
18. Use of Pivot table		1	2	3	4	5
Online or offline pivot tables (e.g. mydatamart or other excel sheets)						
19. Use of Monthly/Quarterly/Annual Reports		1	2	3	4	5
(Aggregation/disaggregation reports of data). Use of reports generated from entered data						
20. Use of Dashboards/Charts/Graphs		1	2	3	4	5
How much visualization used						
21. Use of Maps/GIS/Mapping tools		1	2	3	4	5
How much data visualization is done on maps						
22. Use of interpretations & sharing interpretation		1	2	3	4	5
How much data interpretations shared with a community of users						

23. Use of validation rules/comments/outliers	1	2	3	4	5
How much of the above tools is used					
24. Use of completeness (of reporting) reports	1	2	3	4	5
How much of expression based tools used					
25. Political & organisational resistance to change	1	2	3	4	5
How much resistance to changing e.g. reporting forms, abandon systems etc.					
26. Change in organizational structure in DW implementation	1	2	3	4	5
How much organizational change					
27. Support from all levels of organizational hierarchy	1	2	3	4	5
28. Met with all initial planned timelines	1	2	3	4	5
How was the implementation success on time					
29. Cost did not exceed budget	1	2	3	4	5
How was the implementation success on cost					
30. Provides the envisioned functionality	1	2	3	4	5
How was the functionality of the implementation compared to vision/plans					

The received responses were then analyzed using the Partial-least squares (PLS) path modeling analysis, where we tested the weights of the strength of correlation between the different constructs of our hypothesis. The PLS-PM and its underlying PLS regression analysis is useful in social sciences, because we are able to test the different hypothesis with a limited number of responses. We used to this understand the different success factors in implementation of DHIS2 and its BI Tools. We discuss the evolution of the BI tools that are used to manage the large set of data from the facility-level to the country-level. We took a interviews that were conducted with the developers of DHIS2 and analyzed their interviews with the historical perspective of how the BI tools evolved. This was important to make the paper practitioner focused, so that other developers and implementers can realize the design decisions and architectural deployment choices that are available for BI tools that are useful to manage big data. Our main realization from the paper is that big data is not just a technology term, but is closely related to the organizational capabilities to manage large datasets. This is reinforced from the PLS analysis, but also from the design decisions taken to develop the BI tools in DHIS2.

Study 6: What are new cloud computing models that can reduce digital divide in LMICs for analytics in IeHI?

Data collection for this study has been conducted as a set of interviews with the core developers and implementers of DHIS2 to understand the future direction of analytics tools in DHIS2. While organizational capabilities are important to manage Big data, we realize that the core capabilities of the health system is managing health care, rather than computing resources. We interviewed a total of 12 individuals for the study, each interview ranging between 1-1.5hrs. These were semi-structured interview focusing on issues of deployment, organizational capabilities, computing resources and using vendors for online deployment of DHIS2. These interviews were then transcribed and coded based on the terms used in the different models of cloud computing that are described in the paper. The study highlights the different needs of implementation

countries and health systems with regard to computing resources. These could be purchased as utility from cloud computing providers. Particular examples of implementations from Kenya, Rwanda, and Bangladesh have been provided in the paper, to highlight the distributed requirement of resources in different parts of the world.

The paper uses an interpretive approach to data analysis, with researchers who have regular interaction with the implementation sites highlighted in the papers. The researchers are directly implementers or work closely as consultants with the implementing countries or organizations within these countries. Thus, we are involved researchers in these implementation sites and bring our experiences in the paper as narratives to explain how some of the models are currently in use and how the other models could be used by implementers in the countries.

Chapter 4: Results

From the previous chapters you saw the theoretical understanding from literature to support the work around infrastructures, and then how my own infrastructuring work took place. This chapter focuses on what were the findings and results of the six studies. I use the results to link my contributions and describe the ways in which each study assumes significance in correlating the different themes around infrastructuring work. The results point to two types of contributions. The first contribution is to help answer the research questions through theoretical lens of the AGP model. The second contribution is to help organize practical work of establishing an infrastructure. The first contributions, which I described as C1 and C2 in the introduction chapter, are mainly theoretical contributions that will be better elaborated in chapter 5, when I discuss the results and correlate the results together into the AGP model. The second set of contributions C3, C4, C5 are explained in more detail in this chapter. This chapter is organized as answers to the questions that were asked in studies 1-6 in the last chapter. These weren't necessarily coherent research questions, but rather what we wanted to discover or unpack as part of an implementation, pilot or interviews that led to those studies. The results are empirical descriptions of the infrastructure work that is undertaken in each of the AGP blocks with the goal that it can meaningfully reduce resource wastage in establishing leHIs. Let us hence firstly understand the current efforts in LMICs to establish leHIs and the results of the study are part of these efforts.

In the introduction chapter, I mentioned two ongoing approaches to establish leHIs in LMICs. These were referred as Data warehousing approach and HIE approach. The HIE approach focuses on standardization for formats of data exchange and the services that each registry/application in the HIE is supposed to support. This clearly focuses on a top-down approach to design, where standardization bodies or communities are created to manage the best ways to interoperate and exchange information between the applications. The standardization is on both semantic and syntactic interoperability. The approach is largely based on building informal consensus through meetings and formal voting to determine how the participants and their electronic systems communicate with each other using an HIE. The data warehousing approach is on the other hand less structured in building consensus. There is off course eventual metadata standardization to describe what data is being exchanged, but the process is much more bottom-up in nature. Standardization happens inside the warehouse at the analysis stage, where normalization of data through algorithms, mapping between elements, hierarchy of source of information, etc. is done to collate data and perform analytics. Let us think of these two approaches in terms of how cases have been largely organized in the core II literature. The HIE approach is the centralized, top-down approach similar to the phone

networks, EDIFACT or other infrastructures that are created with a basis in standardization bodies. The processes and information systems are standardized and the main effort in establishing the infrastructure is spent on creating these standards of representing and communicating information. The data warehouse approach on the other hand is a decentralized, loosely organized and bottom-up approach similar to establishing the internet. Standards were indeed created to establish the internet and continue to evolve to this day, but that standardization process is secondary. A sort of side-effect to effectively work with a plethora of local processes and information systems that already work well for users. Both these approaches have established large and long lasting infrastructures, albeit some researchers argue that one approach is better than the other. I do not adhere to such views and believe each approach has its strengths and weaknesses. My view on the matter is that the nature of the context, participants, processes and information systems that come together to establish the infrastructure determine what works or does not work. I suppose that the AGP model and my results do not take bias towards any one of the approaches. Given the interest in LMICs to establish leHIs, it is important to understand the ways in which this work can be understood and organized. I call this work as *infrastructuring work* as a reference to the phenomenon of information infrastructure is an ongoing, dynamic and evolutionary process, as you've seen in the state of the art chapter, which others have referred to as cultivation.

The thesis attempts to answer the following research questions that will allow you to understand the infrastructuring work in leHIs, using the conceptual parts of the taxonomy and as a practical way to organize the work when planning leHIs.

RQ1: Given attention to the ongoing efforts of developing infrastructures, how can activities in an leHI be classified using a taxonomy?

RQ2: What are the blind spots created through this taxonomy and how do they affect the infrastructure evolution?

It might be important to highlight that there is more than one way to answer RQ1. Hence, I use 'a' taxonomy and not 'the' taxonomy. I also realize that there are multiple levels of study, and depending on the level, taxonomies can also be organized as hierarchies. Similarly, I have also discovered that taxonomy, just as any other organizing lens creates blind spots or highlights certain things over others. It is clear from the state of the art that different researchers have decided to look at the phenomenon of information infrastructures from different angles. While some focus on technology aspects, others on management aspects, work/practice aspect, security aspect, equity aspect and I might have missed others in my state of the art chapter. I harp on the above two research questions specifically in Chapter 5, because these are answers that beg discussion. I don't have an answer like "*42 is the Answer to the Ultimate Question of Life*,

the Universe and Everything” from Hitchhiker’s guide to galaxy. Rather, the results presented in the next section for each study contribute to the taxonomy and I summarize them into a table to analyse them using the AGP taxonomy.

Results of the studies

Study 1: Strategy of alignment to existing Infrastructure for mHealth apps

It is important not to look at the mobile phone as a standalone device, but with a systems perspective including various other kinds of infrastructure – such as the paper registers at the sub-centres, the computers at the district levels, the mobile phone networks, internet access, the servers at the state level, and also the basic infrastructure required to support the mobile phone use such as power to charge mobile phone, support centres from mobile operators and handset manufacturers, network coverage and government policies around health information exchanges. This study looks at the strategy of how mHealth applications should be designed to contextual requirements or as described in the second chapter, architected to assemble components of an infrastructure. Here, the architect of the infrastructure needs a strategy of alignment. Alignment with certain aspects of the infrastructure that will allow the solution to scale and participate in the scaling of the infrastructure in itself.

From the two cases of mHealth applications in India and Kenya, I studied the following aspects of Information Infrastructure:

1. **Enabling:** An mHealth infrastructure that is designed to support a wide range of activities, not especially tailored to one is likely to be more actively used. It is enabling in the sense that it is a technology intended to open up a field of new activities, not just improving or automating existing activities. Rather than just capturing data from health workers, mHealth applications can become part of the work culture of the health workers and assist them in their day-to-day activities. In the two cases that were studied, we see this as one of the establishing principle. Although both applications were designed for data collection and reporting, the infrastructure enabled other forms of use. These include peer-to-peer communication at different levels of the health system, communication between citizens and health workers, requests for leave and other HR activities and information exchange about the new developments in the community. Health workers in the Indian case were given mobile phones which had a digital camera in them. This made it possible for health workers to capture images of water-logged pits that were mosquito breeding grounds or skin rash in patients, so that further action can be taken on these by different parties. When these images were shown during monthly meetings at the district-level, officers responded in a better way and health workers felt more

empowered. Thus, the mHealth infrastructure created by these applications acted as enablers for new ways of working in the health system. This enabler of change may sometimes be expected by the designers of the mHealth applications or at other times may be unintended consequences.

2. **Shared:** The mHealth infrastructure brought more partners and users into the network. Mobile phones assist in communication between the medical officers, state health & data officers and the field-level health workers. The infrastructure allows shared information between members of the health structure. In the above cases, we see that the information received through mobile phones is shared between different health staff and officers. In the Kenyan case we see that different programs have been integrated with the advent of this mHealth application. Since the same health worker provides data for different health programs like HIV/AIDS, PMTCT, Malnutrition and Immunization, it makes sense that the same application can report for all this programs. Thus, it avoids duplication of work for the health workers and increases the efficiency and interest of health workers.
3. **Open:** The mHealth infrastructure was open in the sense that there are no limits for number of user, stakeholders, vendors involved, nodes in the network and other technological components, application areas or network operators. There are already different standards that allow open-access to technology in mobile networks. GSM, CDMA, SMS are standards that allow communication between different mobile handsets. There are also standards like XForms that allow standard communication of data and data collection forms on multiple devices. When you look at the Kenyan case, use of plain-text SMS allows different kinds of keywords that can be created by the administrators of the system and change the way in which they report data. This openness in the infrastructure allowed the system to be spread to other programs and health workers. The Kenyan mHealth application also allows new health workers to register themselves and start reporting data. This allows inclusion of many more people into the system and training for the usage of the application is done by one health worker to the other through the word of mouth. In the Indian case, we see that the back-end system of DHIS2 allows creating many flexible datasets and this can be used with mobile phones for reporting. In both the cases, we see that the solutions are open for use with any mobile operator, any mobile handset and other applications continue to be developed for the mobile phones used by health workers.

4. **Socio-Technical Networks:** mHealth is not just a technology piece in the health system, but rather enables a social network of users connecting through technology. Although ICT and mobile phones are part of the infrastructure, they are to be symmetrically seen with the people and processes involved in the system. Infrastructures are heterogeneous concerning the qualities of their constituencies. They encompass technological components, humans, organizations, and institutions. We see from our cases that the designers of the mHealth solutions have given importance to how their users are going to use the system and made quick changes based on the user's feedback. The human aspects of the system like user capacity, health system hierarchy, processes of reporting were considered in design of the systems. The designers of the system did not look at the mHealth solutions as a technology tool in standalone, but as an assemblage of the health worker and the mobile phone + software application. Also the assemblage is between this entity and different elements of the health system was considered by the designers of the mHealth application. The use of closed-user groups (CUG) in the above cases is an example of how business models that support social networks were tapped and better use of the infrastructure was done to foster communication.
5. **Ecologies:** Infrastructures are layered upon each other just as software components are layered upon each other. This important aspect of infrastructures was considered in both the cases that were studied. In mHealth, there is an ecological dependence between mobile operators, handset manufacturers, health workers and software applications. All these players act together to make an mHealth system work and finally might result in better health services delivery. From the Indian case, we see that the implementers first negotiated with the handset manufacturers and mobile phone operators for best deals. The low price and Java-enabled compatibility of the handsets were important components of the ecology that enabled the solution to scale well. The mobile operators providing CUG connection at low prices was also important factor of the ecology. The mHealth solution in both the cases provided a win-win situation for all the parties in the mobile ecosystem.
6. **Installed base:** There is already an installed base of mobile phones, mobile networks, health data capture forms, health information systems. There is inertia within an infrastructure and designers of mHealth application realized this inertia. For e.g. certain areas have better mobile networks from certain mobile operators. People from certain areas are better used to certain languages and certain handsets. This inertia of the installed base is hard to change and

mHealth applications need to align themselves to the installed base. In the Indian case, we see that the DHIS2 system was an already existing system which is widely used by the different states for management of the health system. It was important for the mHealth application to make use of this installed base and is one of the important factors for its success. Similar installed base of a large number of Java-enabled mobile phones being available in the market improves the successful scaling of this solution.

The strategy is to first analyse the existing infrastructure and continuously adapt the solution to fit the infrastructure. The infrastructure should be analysed on the previous mentioned six aspects and the technology should be designed to fit these aspects. The infrastructure is expected to change over time and the mHealth application also needs to evolve with the infrastructure.

Study 2: Meeting local needs means “success” in mHealth implementations

Through the mHealth project implementation in Malawi we realized that local needs are important to define “success” in such projects. This is more important than other forms of technology or components of infrastructure that are not used as a personal device. The mobile phone due to its personal ownership nature is much more situated in the local needs to health workers and work in the community. Earlier forms of ICT have not integrated with the social fabric as much as mobile phones and thus mHealth as much as needs to look at health outcomes, at other times its greatest benefit may lie in the organizational efficiency and ease of work that it may offer, as we saw in our project in Malawi. The results of this study particularly highlights that local needs are quite often different from needs at the higher levels of the health systems or global goals of implementing mHealth. The results of this study highlight 5 main local needs that were met

1. Efficiency of Information delivery – this included overcoming challenges of transportation peculiar to our implementation health areas, but aggravated due to the fuel shortage in Malawi, bad roads during monsoon and limited staff for delivering paper reports to the central health facility.
2. Cost benefits – This might be true to a number of other mHealth implementation sites, but particularly important in places where mobile phone use is cost effective and data services are used by many health workers. One local issue in our health areas in Malawi was the limited supply of paper reporting forms, and printed facility registers. In our case, we observed that health officers needed to spend approximately Malawi Kwacha (MWK) 1500/- for a monthly trip to the district office. There are additional problems of stationary that the health

facilities or officers have to bear out of their own pockets. Mobile data services on the other hand cost between MWK 200-500 per month. Thus replacing paper with a cost effective digital medium of mHealth and thus, solved a local need that is not understood by people at the country-level, where such stationary is adequate or at the global-level, where printers might be available at health facilities.

3. Data analysis and feedback – A common theme discovered in our conversations was lack of feedback from the higher levels of the health systems to the facilities. This is as much as a communication problem, as is a computational problem of performing manual data analytics on paper reports. Electronic systems are known to automate data analytics and create reports for the users. But customized feedback to the health facilities is also possible due to technology that would otherwise take longer in a resource-constrained health system. . The district health offices complained to us that the reports were not delivered on time and hence they were not able to give feedback. They also suggested lack of training or resources to analyse the data that was received at the district health office. We have realized that visibility and power struggles are in place in the context and even if individuals at higher levels of health systems hierarchy were not able to analyse what was being sent, the lower level health workers still needed to submit the reports. In the post-development literature we see that this challenge of visibility of work is not just between so-called “developed” cultures and so-called “under-developed” cultures, but also between individuals because of the perspective of some people being more developed than others. Although we haven’t piloted a solution for the same, we’ve realized that simple automated delivery notification that can be sent from the servers will help build confidence within the health workers at the lower levels and will help motivate them to send reports more regularly
4. Simplicity and ease of use - We have seen that health workers are burdened by the amount of work that is need for reporting data. It is important that the mHealth application should be simple and easy to use. From the design and development of the application, we have tried approach simplicity through two different solutions. We assumed that the browser based solution would be harder to use compared to a custom JavaME application and hence wanted to compare the usability of the two applications. After our research, we realize that the health workers in Malawi were able to use the browser forms as easily as the application. The health officers highlighted the fact that they liked the simplicity of the mHealth applications because they replicate their existing expertise of

paper forms. Both the application, as well as the browser form is similar to the paper forms that the health officers were used to from earlier. The fact that they asked for simplicity as a priority for mHealth application is not highlighted in the reviews of mHealth application that we've mentioned at the start of this paper. Thus, we see that although from a global development level, simplicity of mHealth is not a priority, at the grassroots level, it is important for mHealth to be simple if it has to make developmental impacts.

5. Peer-to-peer communication - In some of our communication with health workers, we have seen that they would like mHealth applications to allow improved peer-to-peer communication. This is interesting to note in the light of how social networks because of the use of technology have skyrocketed in terms of use. The grassroots health workers want to communicate more between their peers to be able to discuss and solve issues in the health system on their own. This highlights a difference in perspective from the global development level where improvements in health systems are primarily derived from communication between top-down and bottom-up levels of hierarchy. Here we observed that horizontal communication in the health system is a priority as much as vertical communication. If mHealth solutions have to make developmental impact, we suggest that solutions should prioritize horizontal communication in the health system and not just focus on vertical communication.

While these local needs were addressed by our pilot projects, when we went back and reviewed what we had discussed as priorities at the national-level or at the global-level, we realized there were clear differences in priorities. The national-level priorities were more on measuring health outcomes, improved data analysis and policy implementation such as quality of data, quality of health service delivery and measuring quality through indicators. Although, you might think of these priorities are only different because they are at a different level of study in the organization, this is not the case. When we initially set out with the higher-level priorities, we design the application differently and implemented only those parts in the mHealth application that prioritised these aspects. But when we went to users of the mHealth application, we realized that the priorities needed to be changed in order for the application to be accepted and used.

Study 3: The OpenScrum methodology of agile software development improves developer as well as user participation in FOSS communities

OpenMRS had a software development process that was used for nearly 6 years before switching to OpenScrum. While it is a continuously improving methodology, some core

concepts have evolved in the last 1.5yrs. The paper refers to this tailored Scrum as OpenScrum. OpenScrum in essence is ideologically the same basis as the original idea behind Scrum, as coming from Rugby (scrummage). A quick restart of the game (sprint) happens after an infraction. During this restart, a front-row of highly skilled forwards, pushes with itself a team with a common goal. The OpenMRS team is led by such high-skilled core developers and push together with the community developers on a common set of activities during a sprint. Organizing the work of software development is an issue of praxis, but OpenScrum is essentially a way in which work is envisioned to be done. It is a set of policies, rules that guide the developers and users of the community to work with each other. Thus, governance of the community is the major goal of creating such a methodology. While it results in a praxis of sorts, the results of the study is to recommend a governance model or methodology, which if followed, might result in improved participation in the community. The prescribed way of working is described in Paper 3 and is called OpenScrum, since it is a tweaked Scrum methodology that is popularly used in different organizations. Its use in a FOSS project is novel and below are some of the differences between Scrum and OpenScrum as observed in practice within OpenMRS.

Table 5: OpenScrum compared to Scrum in agile software development		
Feature	General Scrum	OpenScrum
Sprint Planning	A product backlog is created and sprint targets are created	Announcement made to the community. Community shows interest and participates in deciding the features that need to be built and completed. Design is discussed and available for sprinting
Backlog	A product backlog which is all the wish-list and stories from the stakeholders	A sprint backlog takes focus over a general wishlist. This is a list of available tasks that can be done together by the team. Publicly available for prioritization by community members and contains design documents for coding during the sprint.
Scrum Master	A person responsible for tracking and co-ordinating activities. Helps team to avoid distractions of changing requirements	Scrum Master needs to actively engage community members to participate. Get their views into design calls. Monitor progress of sprints and organize standup meetings. Bring community inputs into a sprint, while not considering

		them as distractions, but improve deliverables based on these inputs.
Product Owner	A person responsible for defining a Product Backlog and confirming after a sprint that product is ready for delivery.	Much more loose and difficult to define. The community plays the role of the Product Owner by voting on issues and prioritizing tasks.
Information Radiator	A large display board that is shared between the participants in a sprint	Similar to Scrum, but more important as it gives live updates to incoming community members to understand what tasks can be taken up by them, when joining between sprints.
Sprint Retrospective	A meeting during which features developed during sprints is demoed and lessons learnt are shared	A presentation made to the community through videos, community calls and mailing lists. Crediting participants for their activities and sharing burndown charts and timeline of changes to the information radiators
Sprints	A time-bound 4-weeks effort to create an update to a product	Smaller duration with group of developers with different backgrounds working together. Individual developers might participate in “spikes”, which are experimental, but related to ongoing sprints. Developers share experiences in the same meetings as the full sprint team.

In open-source projects, the main objective of using agile methods might not be agility. When developers coming from disparate backgrounds and interest work together, sharing knowledge becomes quintessential. In OpenMRS, certain modules were developed by individual developers and other developers did not know the inner-workings of the module. Many production environments of OpenMRS implementations used these modules and hence it was logical to make these modules bundled with the core OpenMRS distribution. Since only one developer was actively working on the module, it meant that if this developer moved on to do other things or leave the community, there would be no one to maintain the module. Even if a new developer started to maintain the module, it will take a lot of time to learn about the module after the original developer is gone. Such low bus-factor can only be increased if the community actively spends time and resources to understand how the module works,

while the original developer is still with the project. Such processes of shared learning, when creating new modules results in better code review and more people are watching the code that is being written. This should generally improve the code quality, but this research has not looked into aspects of code quality, other than just reported bugs against releases. More substantial research of the code base needs to be done to understand quality improvements due to OpenScrum. The other goal of using OpenScrum is that best practices which are learnt during a sprint have been actively created by engaging a number of developers. These shared best practices automatically get transformed into organizational practice. This means less ambiguity in the community in reaching consensus about best practices that should be followed. While leanness is a good-to-have outcome from agile processes, it should be fairly obvious that factors such as economic gains or manoeuvrability are less significant for communities that are working without direct economic bindings. Open-source software development has generally adopted ways of working that are simple for anyone to contribute and leave. This simplicity might be somewhat lost when a community works on building consensus and working together in sprints. Yet, as we have seen, specifically for domain-heavy projects, being able to retain individuals with knowledge is of utmost importance.

The study concludes that OpenScrum, a tweaked agile methodology has empirical basis by which it has helped improve bus-factor in the OpenMRS community. This can be useful to a number of open-source projects that would like to retain developer knowledge and focus on knowledge sharing between developers and users. To answer the specific research questions – agile methods like OpenScrum have improved community participation and developers know much more about each other's code-base. This in turn answers the second research question of sustainability. More developers continue to know and contribute to more modules. The contributions have widened and bus-factor for a number of modules have increased. We also conclude that Agility might not be an appropriate measure for open-source projects. Instead increasing bus-factor through knowledge sharing, increasing community participation and increasing communication are more important measures in open-source projects.

Study 4: Contextual development *inscribes* security problems in software where contexts of use are insecure

Contextualization of software to context, using the concepts of fit, alignment or meeting user needs, as was discovered in our case, has been an important topic in IS literature. This focus of IS researchers comes from a vast body of literature including organizational change – task/technology fit (Goodhue and Thompson, 1995; Zigurs and Buckland, 1998), design science (McKay & Marshall 2005; Schön 1983), decision sciences – cognitive fit (Vessey, 1991), ICT4D – design-reality gap (Heeks, 2002). Although many

researchers including the ones above described the challenges with contextualization, we continue to see researchers differentially study technology design and its use in many environments. Particularly, in the field of information security, I discovered that there is limited understanding between design and use, which leads to challenges in security certification, implementation of large-scale IS and security in II is rarely understood from the broad aspects of move from non-networked world to the networked world. While I will not describe the whole field and argue for the challenge that it represents, the findings of the study point to some methodological flaws in certification process, as well as design of information systems, when moving from non-networked world to networked world.

The study particularly looked at the problem of information systems design and the security challenges that are met when designing systems to replicate existing contextual behavior. As mentioned in my paper, and acknowledged by the researchers above, the problem is not singularly limited to design of IS or evaluation/certification of such designs, but rather an assemblage or combination of design-evaluation that needs to be understood from the lens of inscribing behavior in II (Hanseth & Monteiro, 1997). Although the study doesn't make important theoretical contribution, but rather makes an important practical and beginning of a more critical framework in information security to better study the embedded of behavior in information systems. Thus the main reason why the study is part of the thesis is because it highlights the need to understand the correlation between praxis and design. The study particularly challenges the current practice in security certification process, where reasons for design choices are rarely acknowledged and never integrated as part of the testing process. Rather, the functional testing is a separate aspect that looks at meeting requirements and the choices that were made by designers and if they meet the functional specifications described by the designers.

The study thus is an example to show that current infrastructural understanding is limited in specific domains, information security in this case. The practical challenge that arises due to this is shown in the paper. The efforts in improving security are important, particularly as digital infrastructures are implemented in different domains. Health care in particular affects lives in a big and this IeHIs need to be secure and robust. But care needs to be taken by designers of such infrastructures that contextualization often results in taking practices from the non-digital/paper world into the digital world, can result in embedding insecure practices and embedding these work practices results in information security problems.

Study 5: 3 generations of Operational BI tools and Big Data is dependent on Organizational Capabilities

This study helps improve the understanding of how the HISP network and the DHIS2 implementers use the Business Intelligence tools in health care infrastructures. Although leHIs are supposed to be under implementation and expansion in many LMICs, we have limited understanding of how the data inside warehouses or collected as part of health information exchanges are used by the users of the system. We also examined the use of the term big data and define it not just in term of technological definitions of size, complexity, speed and reliability, but rather in terms of how users can make sense of data and use it. This is particularly important in health systems which have limited resources and often debate between the priority of medicines, health staff versus information systems. One important criteria that allows information systems to be resourceful in these context is if they contribute to saving lives as much as the other resources do.

In this vein, firstly the paper describes that big data is not purely a technical phenomenon, but rather a challenge of resource management and capabilities within the organization to manage and use data. In that sense, big data challenge has been present even in paper systems where data is too big to be managed efficiently and thus the need for information systems. Using organizational capabilities framework, we find the correlation between data warehousing implementation success factors and the capabilities that are required to make use of data. Table 6 shows the correlation and the results of testing the hypothesis that came from the organizational capabilities framework.

Table 6: Hypothesis evaluations (Purkayastha & Braa, 2014)

Organizational Capability	Co-relation
Technology	Moderate
- Higher Standardization efforts will be associated with higher use of BI Tools	Supported
- Higher Infrastructure availability will be associated with higher use of BI Tools	Not supported
Structure	High co-relation
- Higher resources (capital, people, time) will be associated with higher use of BI Tools	Supported
- Higher management support will be associated with higher use of BI Tools	Supported
- Lower resistance to change will be associated with higher use of BI Tools	Supported
-	
Culture	High co-relation
- Higher team skills will be associated with higher use of BI Tools	Supported
- Information Champions at more levels will be associated with higher use of BI Tools	Supported

- Higher user-independence will be associated with higher use of BI Tools	Supported
Acquisition	No co-relation
- Higher disparity of data sources will be associated with higher use of BI Tools	Not supported
Application (Use of BI Tools)	Moderate co-relation
- Higher use of BI Tools will be associated with meeting organizational impl. success	Supported
- Higher use of BI Tools will be associated with meeting impl. success	Supported
Data Protection	No co-relation
- Higher data protection will be associated with meeting organizational impl. success	Not supported
- Higher data protection will be associated with meeting impl. success	Not supported

The findings from this study highlights that organizational structure and culture and definitive factors that determine the capabilities of an organization to use big data. Availability of technology and applications have moderate correlation. The results of the study is to be understood as an addition to popular technical claim that management of big data requires special technologies, algorithms, data processing tools etc. Beyond these tools, our research shows that the factors of structure and culture within the organization determines whether the tools can be used to manage big data.

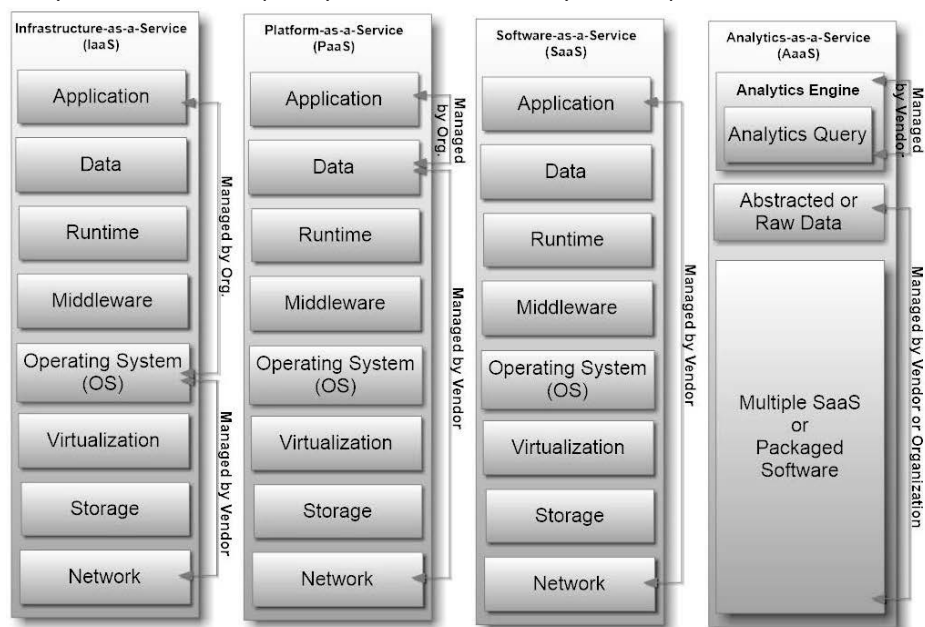
Another contribution of the study is to describe in detail the evolution of the big data tools in DHIS2 over the years and how they play a role in use of information and managing big data. This is important to describe to a designer of such tools, that beyond the technical prowess, the tools need to be designed in a way that they enhance the organizational capabilities. The tools and their evolution also highlights that as infrastructure evolves, the technology and tools need to evolve. Thus, the user needs to which the infrastructure is initial fitting, later continues to evolve, so that the fit between the needs and the users also continues to evolve. This is important for systems and tools to be relevant. Replacing existing tools or components are not necessarily required, but rather to evolve these tools over time, as the needs change is an important finding of this research.

Study 6: Cloud computing can result in sustainable use of analytics tools in healthcare

While we see in study 5 that organizational capabilities of structure and culture are important for warehousing success, in this study we see what are the factors that create challenges for sustainability of leHIs and use of analytics tools. We study this through the lens of digital divide in the LMICs and describe the challenges through 3 levels of digital divide. The access problem of digital divide is well understood due to the

limitations of the reach of the internet, access of health information, access to applications and data analytics tools and technologies. The advent of mobile phone networks to bring internet was studied in our mHealth projects (Study 1 and Study 2) and we have seen immense progress through the use of mobile phones to bridge the access divide in LMICs. Even though it is difficult to claim that access to health information has reached every citizen, the progress cannot be denied. Particularly, the implementation of leHIs to be deployed over the internet infrastructure such as DHIS2 in the LMICs is a big change compared to the earlier paper-based systems. This change has in many ways solved the access problem of digital divide, as data and information management tools are available across the health system. The next level of digital divide deals with capability divide. Here, although the data and tools for information management are available, there is limited equity in the capabilities to use these tools. This is particularly what we see in Study 5 has the capabilities problem of structure and culture in health systems. Training, access to tools, improving culture and inculcating information champions are some strategies to improve capabilities. So, also BI tools play a role in simplifying the complexities of understanding data. These capabilities divide are also multifaceted, in the sense that some of technology implementation, ownership and servicing are dependent on funding agencies that come from outside. This results in a form of dependence that may be good or bad, depending on where this dependence lies. If this enhances the outcomes without creating other challenges then it might as well serve as an enhancer for capabilities of the health system. Outsourcing is often a term used for getting resources from the outside to perform tasks outside the organization. If health care systems core competencies are to manage health and not technology, then the technology management may well be outsourced to other organizations. The outcome divide is thus the third level of digital divide. While the outcome from the use of technology is dependent on a number of factors, particularly in leHIs it is important to understand that data in itself is only a means to an end of good health. Improving health indicators is less of a function of the sophisticated information management tools, but instead depends on how well resources are used to manage

healthcare and provide better health services to the citizens. We describe 4 different business models by which outsourcing or cloud computing might be used in leHIs. Cloud computing in essence is using computing like a utility, similar to how one would use power or water. Conceptually thought of these are separate infrastructures on which other infrastructures are bootstrapped. But based on the number of interconnections, one might view them to be part of the same infrastructure and the boundaries between infrastructure is often something that the researchers or designer needs to assemble. In that sense, the study assumes that there is close ties between the organizations that provide computing resources to the health system, generally through the Ministry of IT in the LMICs, but might also be from other private enterprises who have contractual bindings to rent the infrastructure based on the business models shown in the previous diagram. The business models are important to understand from the perspective that the digital divide doesn't need to be bridged for a limited amount of time. These have to be sustainable in a way that they become part of the infrastructure itself. Hence the particular business models that we suggest are based on how computing capabilities in leHIs can be externalized from health systems, yet remain sustainable and do not create uncertainty of dependence. We particularly discussed and understood that resource sharing between implementation of DHIS2 across countries or regions in terms of analytics as a service was useful. Thus common metadata, indicators would enable ways to understand data and improve the capabilities of the users in the health system. The AaaS model in that sense bridges the capabilities divide and yet does not completely make the health system dependent on an external



provider. As more and more local capacity gets built, these AaaS providers could be internalized within SaaS, PaaS and IaaS provider. So in that sense, our discovery was that there is a layered approach of dependence or non-dependence between the services that could be provided by an external organization to support the capabilities of the health system in terms of analytics activities.

The next part of the chapter discusses the contributions and how each study answers the questions that were raised in the first chapter and third chapter of this thesis. Particularly, the goal is to tie together the research questions from each of the papers, to the answers from each paper and the contributions to the overall research questions of the thesis.

Overview of the contributions

Table 7: Papers, contributions and research questions					
	C1	C2	C3	C4	C5
P1	✓ (RQ1)	✓			
P2	✓ (RQ1)	✓	✓ (RQ2)		
P3	✓ (RQ1)		✓ (RQ2)	✓	
P4	✓ (RQ1)	✓ (RQ2)			
P5	✓ (RQ1)	✓			✓ (RQ2)
P6	✓ (RQ1)				✓ (RQ2)

You might have realized by now that the main focus of my thesis lies in understanding success, design strategies to reach success and build upon this success in establishing digital infrastructures in health care. All these activities come under the label of infrastructuring work, which I've attempted to organize under the categories of Architecture, Governance and Praxis – referred to as the AGP model. My work and nearly all cases that study the phenomenon of II will tell you two aspects that I would like to highlight here, but I will discuss in Chapter 5 in depth:

1. **Non-sequential:** That the work of understanding success, strategizing towards that success and scaling the success of the infrastructure is not a linear or sequential process. As in my own research I initially looked at successful strategies to design scalable mHealth solutions and in that process discovered that the success of implementation may not lie in the scaling of the same solution across the whole health system. While scaled solutions, due to its apparent use might be considered successful, non-scaled solutions might also be successful when they meet local needs.

2. **Indirectly causal:** That the work in one of the categories of AGP may not directly cause changes in the work done in other categories in AGP. Sometimes the researcher, implementer or user perceives causality and places such work in the intersection (like in a Venn diagram) of the A-G-P space, but at other times there is indirect ways in which work is affected. Did we design software using OpenScrum for scaling or enhancing big data capabilities of the health systems? Yet one could argue that the need for such a software development methodology arose from large-scale implementations that were a result of scaling success.

C1: Taxonomy to classify II activities into Architecture, Governance and Praxis

Paper 1 mainly dealt with the design strategies that allowed better fit between mHealth solutions and the existing infrastructure. So while we discover that bootstrapping the installed base enables to scale the architecture, we do not realize that each locale has a nuanced need that is different from another locale in the same infrastructure. The infrastructure is understood in paper 1 to be heterogeneous, but the understanding that the architecture needs to be flexible to accommodate these multiplicity of needs dawns upon much later, mainly in paper 2. Thus, paper 1 directly contributes to the architectural work in leHIs and contextual design of mHealth solutions, and indirectly highlights the contribution that there is a form of causality between mHealth implementations in India and Malawi.

C2: Highlight the organizing categories of activities by analysing cause-effect relationship in cases from eHealth implementations

The causality is not just that the mHealth application was developed in India and brought to Malawi. The phones, the local capacity with mobile phone operators, the type of devices etc, all worked to create a different infrastructural praxis that was envisioned in one way by the architecture from India and based on national-level requirement assessment, but was practiced (praxis) in a different way. The scaled mHealth solution from India faced many challenges in Malawi, but even within the health areas we discovered multiplicity of needs that only praxis studies in the context can highlight. Even with an interpretivist lens it is often challenging to see as an external researcher and a critical lens is often apt to understand the contexts of multiple locales. Contribution 3 is a subtext across all the studies because the relationships and the perceived causal effects are central to the AGP model's lens.

C3: Articulating Information systems implementation success in terms of meeting local needs

Paper 2 thus makes three contributions. Firstly, it provides a praxis study of the infrastructure to better understand success of mHealth implementation. This helps establish the AGP lens' architectural infrastructural work in leHI. Secondly, it highlights

the interplay between local needs and adapting architecture to those needs. Thirdly, it places the multiplicity of needs of each locale, as a challenge to scale and puts forth the need of architecture to be flexible to meet this challenge.

C4: OpenScrum agile methodology to improve knowledge sharing in OSS communities

Paper 3 brings a novel methodological contribution to software engineering. It describes a tweaked Agile methodology for software development that not only builds software that caters to user needs, but also ensures knowledge sharing between the team that builds the infrastructure. Thus, users and developers become participants in the architecture based on a governance model, also described as a way of working. These rules of how communication takes place between developers and between users and developers is formalized and described in Paper 3 as OpenScrum. The methodology is also unique that it is at least from the deep review, the first attempts by an open-source community to adopt an agile methodology in all its practices. The longevity about the adoption and long-term effects are yet to be fully understood, but this contribution is highly relevant for the governance of open-source projects that are domain-heavy like healthcare and need knowledge sharing between users and developers. This knowledge sharing in essence makes the community realize the local needs and make attempts to meet those needs.

C5: Defining Big Data through Organizational Capabilities that can leveraged by the use of Operational BI Tools for analytics in leHIs

Paper 5 contributes through the study of evolution of Operational BI tools in DHIS2. We discover that big data can be defined through organizational capabilities and tools need to be evolved over time to improve the organizational capabilities. The paper understands the architecture of the tools and how these evolved with the changing praxis and user needs in the infrastructure. This falls in place with the taxonomy of describing architectural infrastructuring work, but also shows the perceived causal effects between user needs and evolution of the BI tools. The contribution that big data is beyond technical terms and needs to be understood through an organization lens is another contribution of Paper 5. In that sense, Paper 6 also contributes to this aspect that organizational capabilities cannot be considered to be a one-off moment of realization, but needs to be sustained practically by bringing other participants in the infrastructure. This means using cloud computing business models to sustain the use of operational BI tools and bringing providers of such business models as participants of the infrastructure.

Chapter 5: Evaluation and discussion

The results and the contributions in Chapter 4 highlighted aspects of the work done from each study that is part of this thesis. In this chapter I would like to evaluate the results of each study using the AGP model, bring coherence between the infrastructuring work categories, and discuss the constructs in a way that the reader is convinced that I've answered the two research questions that were put forth in the introduction chapter. In this chapter, I will also attempt to highlight some of the blind spots that are created by lenses/framework of research models. Particularly what are the blind spots that the AGP model creates and how does this relate to other concepts, models and frameworks that have been used to understand the II phenomenon.

leHI – in a mode of continuous flux

leHIs in LMICs is a moving train that has left the platform. For better or for worse, global as well as local forces are moving the train of leHIs forward. It is in the process of being implemented from at least two perspectives that I've understood and explained in the introductory chapter – namely warehousing approach and HIE approach. These approaches largely come from the efforts in HICs, where leHIs have been established and as mentioned in Chapter 2, have faced numerous challenges. Yet, for many, the advantages of leHIs outweigh the challenges. This thesis is a small subscription to the idea that these challenges are being thought of with deep introspection.

As with all studies of II, health information infrastructures are more and more focusing on being integrated. Multiple systems have complex interconnections for information exchange. These systems and the users of the systems are continuously innovating new ways of working the infrastructure. I refer to this as leHI and acknowledge that it is in a state of continuous flux. Take for instance, the mHealth solution that was developed in India as a pilot, then evolved through use in Punjab and was later scaled to a whole state. The code base regularly changed and was used across many other countries. The flexible architecture of the application allowed sometimes simple (like in Punjab) and other times complicated (in Malawi) reconfiguration with the existing infrastructure. The context of use highlights that the components in the leHIs need to be flexible to scale and survive in the leHIs. This flexibility in the components of the leHI, if anything describes what others have described as malleability of II (Klashner & Sabeth, 2004; Kallinikos, 2002; Nielsen & Aanestad, 2006). It is important to discuss that malleability is something that is part of the components that make up an infrastructure. Many researchers (Tilson & Lyytinen, 2010; Ritcher & Reimer, 2013) consider malleability to be part of the software components that are part of the infrastructure. I consider this to be only one half of the story. As we saw from our results of the organizational capabilities study in the use of operational BI tools, that our tools evolved along with the

user needs. User needs were not always met linearly first by the software or infrastructural medium (internet for DHIS2), but the users were also flexible to change their needs based on what the tools could provide. Monteiro & Rolland (2012), Braa & Hanseth (1998) and many other praxis focused studies describe this malleability of user needs and workarounds beautifully. This aspect hence verifies my need about the AGP model. That such a holistic lens can help view the misses of concepts like malleability that are seen in the infrastructure, to limit only to software/architectural flexibility, but should also be understood as governance flexibility and praxis flexibility. Consider the agile methodology of OpenScrum as a change in the governance process in the OpenMRS community towards software engineering. You would come to see that the change results in ways in which developers collaborate with each other and the community of users. The share knowledge better with each other and communicate much more about the changes that are required to meet the user needs. Agility in this sense is less relevant, as one might have hoped for. As developers or designers of infrastructures, we would like to change directions faster to meet with changing needs, but as we can see OpenScrum does not necessarily improve Agility, as much as it improves understanding of user needs. May be with time, the governance process also improve agility, but even if that does not happen, we can claim that malleability of the infrastructure is not necessarily deterministic in nature or predictable by the designers or governance law makers in this case.

The continuous flux of the leHI is thus due to the malleability of its components, including the architecture, governance and praxis in the infrastructure. How much of it because of each of the category is difficult to determine. As I mentioned in the last chapter, evolution in II is non-sequential. In fact as highlighted by many researchers it is sometimes fast, other times very slow. As was found in the case of the OpenScrum methodology, the rate of change of software releases was gradual due to the change in the governance policies of software development. The agility of OpenScrum is indeed not its main contribution to the OpenMRS community or the leHIs that use OpenMRS. Instead, it is the rapid increase in knowledge sharing between the community members that has been discovered as part of the governance changes. The non-sequential nature of causality in the evolution process of II is also apparent in the first two studies of mHealth application. While scale was reached from the pilot to the state-wide and international implementation, the success factors were understood much later when we customized and adopted the solution to the different locales in Malawi. The architectural flexibility of the software allowed it to be reconfigured to infrastructures in Punjab and other countries, in Malawi the highly different local needs in the two particular health areas compared to national-level or global needs made the implementation difficult. Until we realized that we needed to better understand the praxis and in turn the local need, we would not be able to call the implementation as

successful. This critically made us realize that governance changes in how systems get developed are required to make the implementation successful, rather than just malleability in the software architecture. To clarify further, what we understand as sequential in software development or implementation science or action research is rather an ideal state that is rarely seen in infrastructures. We often describe action research as a process of sequential steps starting from problem formulation to evaluation as an iterative cycle, but infrastructures often cause breakdowns in between the cycles as was in the case of Malawi. Instead, often the workarounds or malleability of praxis results in multiple cycles around say problem formulation step itself and then that in itself adds to evaluating our initial assumptions of the problem. Action-design research (Sein et al., 2011) is an interesting exercise in this vein, where the sequential nature of action-research or design-science is surpassed by this understanding of malleability of the infrastructural components – be it user, technology or in fact an assemblage of two. ADR thus is an interesting avenue for future research in infrastructural evolution, especially in the design and implementation of leHIs.

Evaluating the studies using the AGP model

The AGP taxonomy can guide any of the concepts that have been studied in the II phenomenon, to provide a more holistic view. Let us discuss each of my six cases through the AGP taxonomy. Table 8 summarizes this analysis:

The focus of the Study 1 is about the architectural work that is required to match existing infrastructure in mHealth applications. The study highlights the weightage that needs to be given to design factors, so that it can be aligned with the existing infrastructure. The study lists out 8 design factors that need to be evaluated in design of mHealth solutions. In both the cases that were part of the study, we saw that the architecture of the mHealth system was flexible and enabling. It allowed changes to the data that could be reported and additional forms of communication using peer-to-peer (P2P) messaging systems proved extremely valuable to the users of the system. The enabler of being able to take images of cases and get feedback from doctors empowered health workers and motivated them to use the infrastructure. The open architecture allowed enrolling multiple types of users, devices and applications into the infrastructure. There was no lock-in of a particular provider, but the Closed-user group (CUG) effectively meant that people on the same mobile operator had unlimited communication with limited cost attached to it. The existing installed base of DHIS2 in the Indian HIS infrastructure proved to be extremely useful, since that architecture allowed shared use of datasets and data reported through non-mobile devices. So, data from health workers coming from mobile phones could be correlated with data coming from hospitals and household surveys due to the shared architecture. This kind of workflow of correlated data was not envisioned in the pilot study, where it was only meant to test the feasibility and usability of mobile-

based reporting of data. It could be argued that the governance regime in both the cases that were part of the study was already enabling and participative. It is difficult to analyse if the enabling, open and shared architecture improved or was aided by this governance regime that was participative. There was no clear indication that such governance regime was a result of the architecture or the architecture was a result of such governance regime. In later part of the story of this same mHealth application in Punjab, described in Section 5.4. in paper 5, in fact what we could see is that there was conflict between the governance regime to capture big data and actual praxis of health workers. The governance regime wanted to monitor the behaviour of health workers because mobile phones and the open architecture enabled such possibility to capture and generate big data. But the health workers themselves did not want to report such data since it was not necessarily reflecting adequately the work that they were involved with. The participative governance regime did effect the praxis of health workers by expanding the use-cases in which the mHealth application started to be used. Users become active participants in reporting data and making/sharing decisions between each other and between different levels in the health system. Here Karma could be an effective vehicle to justify whether or not the application/mHealth system enabled the praxis of P2P communication. It was Karma that later in Punjab there would be a sort of revolt from health workers against using the mHealth application to monitor the day-to-day work of health workers by the health ministry. In essence, we could question if technology of mobile devices made the top-level management start monitoring health workers or was that motivation always there and was enacted through other forms even earlier. The mHealth system probably only made it visible to the health workers. The focus of Study 2 is on the praxis aspects to understand the local needs of health workers who use the mHealth application. The study argues that success does not necessarily come from scale as was seen in the Study 1. From our implementation experience in Malawi we saw that the organizational efficiency was the biggest need of the health areas where we implemented the mobile application and faced a number of challenges due to the existing infrastructure. Particularly, the project changed its focus from a JavaME application to a browser based application due to the praxis that was in the infrastructure. The hybrid deployment of the mHealth architecture was a result of the praxis that was seen in the infrastructure. The health workers did not have the same type of mobile devices, they reported on different datasets and expected individual feedback based on the program they worked in. This in turn led us to employ an architecture that met with the demands of the praxis. Due to limited human resources, expensive fuel due to shortage, bad roads due to rainy season and lack of paper forms and reports, the mHealth adoption rose rapidly for all kinds of requests outside the scope of the initially planned datasets. The open architecture resulted in a wide variety of devices participating in the infrastructure. Originally we had brought cheap JavaME

capable devices from India that were incompatible with the mobile operator networks in Malawi. This breakdown became the opportunity to allow any mobile browser to report data and we configured our study to reflect this. One health area used the JavaME application, while the other health area used the browser based reporting system. Both met the local needs but after a number of breakdowns, workarounds and challenges that were unplanned and changed a lot of architecture of the mHealth solution in the process. The open-source nature of the software applications allowed rapid customization and help from the global community in meeting the challenges incurred due to breakdowns. The breakdowns were thus considered effects of the user needs of each locale. But weren't the local needs always different? It was only when we started facing the breakdowns that we realized the needs expressed at the national level or global implementations of the mHealth application were different from the local needs. The Karma in this case was only the visibility of the effects that were already present in the local needs. We as researchers and partners in the implementation were only looking at what we knew from the national level group discussions and our experiences from implementation in other countries. The Karma that we needed to better understand the local praxis, resulted in an introspection to derive new methods of governance for software design and development. This is where we saw an urgent need to better understand how we can organize our development methodology to meet the local needs.

Paper 3 thus became an eventual effect of the challenges faced in Study 2. This becomes evident only when we look at the broader connections between the cases. This cross-study effects or causality can again be understood as Karma. The OpenMRS community which is part of the leHI in many countries faced similar challenges to meet the local needs. As an EMR system, its solution to meet local needs had always been in its modular architecture. Local customizations and needs were developed as modules that could be installed over the core application. These modules were community developed and managed by the implementation that needed the specific module. While this worked well for the local needs, there was a challenge of sustainability whenever the original developer of the module decided to leave the project or another implementation that had similar need, but still needed some customization of a module. The local need required some minor tweaking to the module to meet its need, but since they did not know the inner workings of the module, they would either start from scratch or face lots of difficulties in tweaking the module to their needs. Thus, the OpenMRS community decided to adopt an agile methodology that would allow more speed to adopt to the local needs of an implementation. With agility as the focus, the OpenScrum methodology was started to be used in the OpenMRS community. The OpenScrum methodology can be understood as a set of rules and practices around collaboration in software development. These were divided on tasks that developers would do and how

they would communicate in writing code with each other. The peculiarity of a large and diverse community such as OpenMRS needed to tweak the Scrum methodology to achieve its goal. But instead of agility being the expected outcome, improved knowledge sharing and communication between users and developers became important highlights of the change in software development methodology. An important observation to reflect on here is whether the side-effect to all this is only a time lag and agility will come once everyone has adopted the new working practice for a number of years. This is an open question that only time can answer. What is important to observe is that the side-effect is essential to solve the praxis problem of better contextualization of software along with sustainability of open-source projects. Did the architecture of OpenMRS change due to the new governance regime of distributed yet coordinated software development? Not really since OpenMRS from the beginning was a modular architecture. Instead, what we perceived was that the praxis had changed due to governance. Obviously, the goal was this new governance regime was to change the way software was being developed, but the praxis changed the way modules became the dominant focus of the software releases. Earlier multiple implementations or each individual developer released their own fork of a module, but now since the whole community got together and focused on modules for a short period of time, the effective productivity decreased since everyone was doing everything. But in the longer run this enhanced the overall productivity of the team as a whole. This change might not be something that is desired for a number of software projects, particularly may be in proprietary companies where productivity enhancement is the goal of such efforts. In an open-source community, that too in the health domain where contributors need a lot of domain knowledge, sustainability and knowledge sharing become key elements. Thus, the contribution of OpenScrum to the OpenMRS community is eminent.

One of the problems that we discovered in contextualized software was that of information security as part of Study 4. Though inscription is a well-researched topic, also described with the recent term of *sociomateriality*, its aspects in infrastructure alignment and contextualization is less understood. When the security certification of a globally developed open-source HMIS was being done, we saw that a number of problems that were identified using the OWASP methodology, we not in fact bugs according to the users. The users had requested much of these features because either these were the actual praxis in the context or they simplified the work at hand. The process of security certification did not take cognisance of the user needs when testing, but rather only looked at a one-size-fits-all paradigm to security problems. On the other hand developers did not take cognisance that there are differences between networked and non-networked world. This is particularly important for patient-level (cont'd on page 101)...

Table 8: Analysing cases through the AGP model

Cases	Architecture (A)	A-G	Governance (G)	G-P	Praxis (P)	A-P
Study 1: Design and implementation of mobile-based technology in strengthening health information systems: aligning mHealth to infrastructures (Purkayastha, 2012)	-flexible/enabling architecture - open architecture - shared resources of infrastructure to report data	- Limited causality with governance activities	- enabling and participative governance regime - Users are encouraged to create their own applications and customizations to the mobile devices	- peer-to-peer communication enabled through CUG plans - capture daily data from health workers (big data) - group communication over SMS using special codes in the Kenyan case	- users expand new datasets, use-cases on the mobile reporting and SMS application - open-source technology allows modification to code base and different freedoms to innovate	- Limited causality between architecture and praxis
Study 2: Post-development perspective on mHealth – an implementation initiative in Malawi (Purkayastha, Manda and Sanner, 2013)	- a mix of JavaME, browser based application for reporting - open architecture where devices with different capabilities can join	- Billing plans for data services to monitor the differences between users of JavaME application and browser	- decentralized decision making on use of mobile plans, devices and reporting application - centralized feedback mechanism to compare facilities on health indicators	- decentralization results in improved articulation of user needs - praxis evolved its own ways to report to the health system and improve reporting rates	Organizational efficiency, such as reporting rate, cost savings, feedback were better received than planned - measuring health outcomes became secondary to project evaluation	- change from single JavaME app to a mixed architecture of browser and multiple mobile providers - a device agnostic platform for data reporting evolved instead of initial plan of JavaME app

Study 3: OpenScrum- Scrum methodology to increase bus factor in an open-source community (Purkayastha, 2015)	- increased focus on modularity - loose coupling between core and modules	Limited causality between architecture and governance activities	- a tweaked methodology to work in sprints - community participates on same codebase for short bursts of time - Sprints, spikes, code reviews, standup meetings improve knowledge sharing in the community	- developers feel more empowered and understand codebase better - community feels greater ownership of the codebase	- sprints decrease productivity of top developers, but brings common productivity among developers - users participate more often as product owners, and developers have better understanding of codebase - less agile, but better understanding of needs	- application releases focuses on more modularity - releases of modules speeds up compared to slowness
Study 4: Towards a contextual insecurity framework: How contextual development leads to security problems in information systems (Purkayastha, 2010)	- centralized architecture of sharing patient records and aggregate data - aggregate data is architecturally separate from patient-level data	- Limited causality between architecture and governance activities	- distributed software development - global team does not closely monitor activities for local security certification	- Limited causality between governance and praxis	- in paper systems patient record is available on at facility - passwords are shared between users due to limited internet connectivity - users lock themselves out	- password reset not available with user and needs admin intervention - patient records are available centrally to all health facilities - encryption not required for aggregate statistics, but essential

						with repeated incorrect passwords when they do not want to work	for patient data that is centrally shared
Study 5: Overview, not overwhelm: Operational BI tools 1 for Big Data in health information systems (Purkayastha & Braa 2015)	- internet based tools from offline Access software - high performance analytics tools that are flexible - 3 generations of BI tools are based on shared and open architecture	- centralized data storage and processing - core team that globally determines needs and flexible architecture	- hierarchy of standards based on what is required at each level of the health system - decentralized decision making process - BI tools allow mapping and comparing health facilities	- Limited causality between governance and praxis	- organizational structure and culture determine success of BI tools - even with availability of technology, limited use due to structure and culture factors - low sustainability of technical skills		- Limited causality between architecture and praxis
Study 6: Big Data Analytics for developing countries: Using the Cloud for Operational BI in Health (Purkayastha & Braa 2014)	- new scheduler for analytics based on computing resource availability - open architecture that can scale on multiple servers	- new architecture for deployments enable use of cloud computing providers - resource sharing between countries possible due to shared infrastructure	- move IS management to external vendor - use cloud computing models of IaaS, PaaS, SaaS or AaaS for using external services	- integrate capabilities of external vendors into the infrastructure	- Limited changes to praxis due to new business models		- Limited causality between architecture and praxis

records and how they are stored in the system. The centralization of aggregate records are not complex because they do not involve privacy issues, but still needs the other pillars of InfoSec - integrity, non-repudiation and authenticity. On the other hand because of the centralized architecture, patient records need to be made secure in terms of how it's stored and distributed. This is a challenge that in the contextualization of software does not exist in the paper world. But the designer of system need to understand that there are other considerations when moving from networked to non-networked world. Another likely challenge when you think about the complexities of hybrid deployment (some facilities/districts using offline, while others use web-based), is that training of the users when using the system. In a lot of countries, due to the infrastructure the HMIS system is deployed with a hybrid strategy for better fit with the infrastructure. But this complexity means that the praxis in the offline and online systems are quite different and the HMIS system needs to be designed flexibly in such a way that it meets the praxis needs of the users. Security certification agencies and their available methodologies fail to recognize these differences and rate security on a common set of parameters. Another connection between Praxis and Architecture that is seen only when looked from this holistic lens is factors of usability, such as password reset. Security certification standards suggest a password lockout mechanism when a user provides an incorrect password for a number of times. Along with a centralized password reset service, this causes an observed phenomenon where users regularly lock themselves out of the system. This causes unnecessary delays and reduces efficiency, which a lot of managers say is deliberately done by users who do not want to work. Even if we ignore this view of the managers, the praxis solution to have simple, common passwords across multiple users is not acceptable for any deployment of a modern networked application. This conundrum between praxis, user needs and security problems is what I refer to as contextual insecurity. There is probably much more to be articulated with regards to the framework, but the essential component of such a framework is the Karma between Architecture, Governance and Praxis. Designers perceive causal relationships between these blocks and inscribe behaviour in the infrastructure that is considered insecure by the security certification agencies. The security certification agencies on the other hand perceive that praxis and user needs do not need to be inscribed, as they are the cause of the insecurity. Karma is a useful concept to highlight the sentient causality because it is only perception of reality of the actor viewing the infrastructure. There is likely a realist or positivist view that there are indeed incorrect practices that need change. The reality that there have been data breaches or opportunities for data breach due to these insecurities is not under debate. The debate is whether the certified changes that are supposed to make the infrastructure more secure will make users change their praxis. The question is whether

changing the praxis or finding workarounds will not inscribe other contextual insecurities since you would still consider user needs to be priorities.

The papers 5 and 6 are problems of managing scaling and how we've made attempts to manage the problems of scaling. While paper 5 is about the way we define big data with organizational capabilities, it is also about how the tools have been designed to meet and enhance organizational capabilities. This is the reason why it is placed in the Architecture block, but closely aligns with governing the healthcare infrastructure by developing these organizational capabilities. Culture and Structure are the formative factors in defining how big data will be managed in the leHI. While these are in fact enacted in praxis, we look at the design of tools and architecture of deployment. Hence these are visions and policies of how the tools need to be used. The enactment in many ways can only be understood through praxis studies that look at how users actually use these tools and whether they continue to be overwhelmed by the big data or are able to create an overview of the data. The level of study in such cases will need to be at a much more micro-level. That is the argument for placing this study in between the Architecture and Governance block, but does it represent a Karma or causality for it to be placed in that intersection?

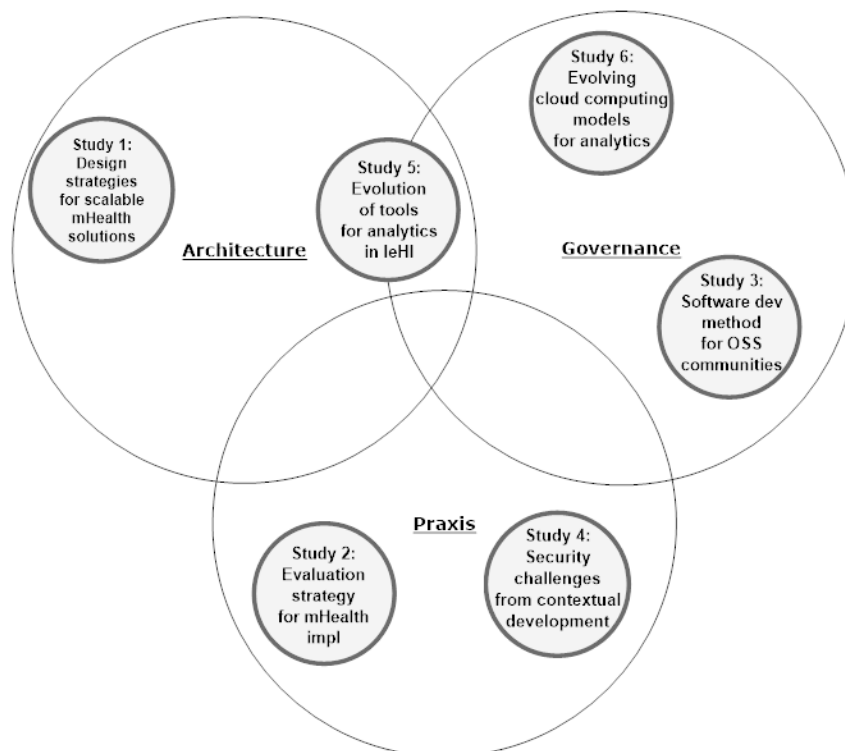


Figure 12: Placing the studies in AGP model

Let us see in Table 8, about Study 5 and some of the architectural considerations. The evolution from Microsoft Access based application to a web-based application was the biggest change over the years for DHIS2. Over the many years that health infrastructures have existed in Africa, the primary argument against integrated or web-based deployments to integrate data from multiple sources has been lack of internet penetration. Manual aggregation using emailing of Access databases to integrate was done from 1994, but due to lack of internet penetration this was never done in real-time and thus, as we argue did not allow for Operational Business Intelligence tools that leHIs now offer. In the meantime, while DHIS2 was being developed, mobile networks with internet access started to reach scale in Africa. DHIS2 was deployed in hybrid mode in Sierra Leone (Braa et al., 2010) with the hope that in other places around Africa, internet penetration would allow direct access to facilities. This infrastructural change was one reason why DHIS2 and its Operational BI tools could scale so well. Now, if a realist would look for causality and causal forces, it might be argued that DHIS2 and its scaling depended on internet penetration. You could also observe that mHealth applications allowed getting data from the last mile due to internet penetration. But was the causal force in mobile networks or DHIS2 or mHealth apps to scale? Was the innovation in the infrastructure's medium or the users who innovated on the mobile internet infrastructure? To the AGP lens it doesn't matter where the causal force resides because it should be considered only a perception that is limited by space and time view of the infrastructure. The designers of the tool continued to be flexible to match the infrastructure. Sometimes through intermittent solutions in BI tools, as you see the three generation of BI tools. Each of the generation of BI tools represents a perception of the causality for the designers of the BI tools and those deploying it, using it and making it part of the infrastructure. The mHealth application from India that was taken to Malawi (as seen in Study 1 and Study 2) and other mobile initiatives created large amounts of data to be part of the infrastructure. Is this why the infrastructure has big data? A realist causal force would indeed claim so, but didn't that data exist forever in the paper registers at the facilities and in the health infrastructure. Its integration in the leHI with the use of mHealth applications is only an enactment of the architecture (praxis) that allowed the infrastructure to be shared with these new mobile phones sending data that existed only on paper. Thus Karma is only this perception that the mobile phones and the mHealth applications caused big data in the leHIs. Similar is the suggestion that new business models are coming up to build organizational capabilities. If the cloud computing providers did not provide these services would the new business models or models of deployment (IaaS) or data warehouse or HIEs (PaaS) services not used in the infrastructure? That again for the AGP models view on causality is irrelevant. These are only opportunities which Architecture of the tools that are deployed in the

infrastructure need to be cognisant about. Users will enact these visions in a praxis that needs to be studied for these claims of “success” to be verified and validated.

Blind spots in the AGP model

The AGP model as you might have understood by now, focuses on work that is done in establishing of an infrastructure. It focuses on agency of the actors, assemblage of the human and non-human components of infrastructure. It highlights the work done by the actors and how these are categorized into blocks of activities. Visibility of work is often a challenge for researchers to study, as has been well documented in the CSCW field (Bowers, 1994; Star & Strauss, 1999; Lewis & Simpson, 2013). Thus, the challenges of visibility of work apply to the AGP model. Particularly the researcher has to take care to not overlook the work in each of these blocks and how they were perceived to affect other work that happens in the other blocks. While I have talked about blind spots throughout the results and discussion of each paper, I’d like to summarize them here:

1. **Blind spot to micro-level methodologies and concepts** – This is particularly seen in Study 2 and Study 3 where they focus on methodological or theoretical notions, but the AGP model doesn’t seem to clearly articulate their necessity. While there is a salient prescription that praxis studies need a different kind of lens to be studied, there is no deeper articulation as to what these might be. One of the reasons for this is that the AGP taxonomy is placed at the genus rank and more research is needed for micro-level taxonomies in each of the block. It still is a blind spot that users of the AGP model need to consider. There is also considerable loss of understanding how
2. **Blind spot to notions beyond work** – There is a whole body of concepts for example structure, culture, politics, counter-networks that are not well understood as work. These become blind spots when using the AGP lens. There are other theories like structuration where structure is central theme and the interplay between actors and their agency in creating these structures is highlighting. These might have close correlations at a deeper taxonomical level to the AGP model, but at the genus level, I acknowledge them to be blind spots when using the AGP model. Structure, culture, politics are probably also forms of work that are done within and outside the infrastructure and that brings me to the last blind spot.
3. **Blind spot to external activities of the infrastructure** – This comes from the fact that infrastructure is not everything of everything. All agencies that act forces of

causality on components of the infrastructure cannot be considered part of the infrastructure. There are boundaries that help articulate what work is done within and for the establishing of the infrastructure. These are the ones of relevance for the AGP model. External activities that might have effects on activities in the A-G-P blocks. Although these are part of the context itself, they do not necessarily become part of the infrastructure. Consider the fuel shortage in Malawi due to politics, economy that was part of the context. Yet, these are not part of the leHI infrastructure, but rather part of the context. The praxis studies need to focus on these contextual realities, but the AGP model does not prescribe directly on the micro-level details of the context since they are not part of the work in establishing or destabilizing the infrastructure.

Reflections on the research context

Before concluding the thesis, I would like to highlight how the context has shaped this thesis. As with many action-research projects, partners in implementation sites are core to the learnings from the project. As mentioned that leHIs are in constant state of flux. The context of LMICs are also resource limited and hence juggling between resources adds to the increase in the flux. My research design has thus changed course a number of times. The course change has resulted in not evaluating the Punjab mHealth implementation or the implementation of an OpenMRS EMR after the methodology change. These were supposed to be done in India according to the initial plan of studying an leHI, but due to a number of changing factors, these are not part of my PhD thesis. The context is thus in some sense distributed across multiple geographies and studies. But yet the underlying Karma and activities in the A-G-P blocks reflect the coherence of how these come together in a quintessential leHI. The research context also is limited only to LMICs and the nature of leHI implementations in LMICs. There are already fairly large-scale HIEs for example in the US and other leHIs in Canada and Australia. These are by no chance perfect, but have differences that are unique to how they have evolved from praxis in these geographies. The appendix B paper in that sense reflects my desire to contribute to contexts where standardization is limited and electronic intelligence using semantic information can be used in leHIs in LMICs. LMICs are potential greenfield because there are limited electronic systems in use. This provides excellent opportunity for standardizing vocabulary and starting from a clean slate. How clean that slate really is, with existing paper practices and multiple languages in places like India is debatable. Yet, describing health information with semantic data and using this semantic data to integrate is an interesting challenge for the future of leHIs as they mature in LMICs.

Conclusion

The integration of electronic health information systems results in complexities that often result in implementation failure or huge cost overruns. Still leHIs are being implemented all around the world. This thesis concludes that the complexity of leHIs needs better understanding and organizing of infrastructuring work. The AGP model is a taxonomy that provides an organizing thought to infrastructuring work. The AGP model is a holistic view to the different categories in which activities to establish an infrastructure can be performed or studied. The AGP model can be applied post-facto to study the evolution of an infrastructure or as a way to organize teams and their work in the activities of establishing an infrastructure. It provides a more detailed and holistic vocabulary than “cultivation” with regard to II evolution.

The architectural block encompasses all the work that is related to design, and contextualization of the components of an infrastructure. This architectural block intersects with the other two blocks of governance and praxis, where the researcher using the AGP lens can clearly articulate a causality between the activities. The governance block covers all the work related to establishing policies, laws, and ways of working within the infrastructure, between the components of the system. The praxis block covers activities that are enactment of the envisioned ways of working between the components of the system. Components of the infrastructure should be understood as a socio-technical assemblage of users and technology. Thus, architectural block involves the designers and their designs that are assembled together, the governance block involves the laws and the ways in which the laws are made and the praxis block is how the technology with other technology and users interact with each other and enact the architecture and governance policies. Activities in each of the block creates a perceived causality or sentient causality called Karma, which is enacted upon when the activities in other blocks relate to those activities as causal. The concept of Karma explains why actors within the infrastructure consider some activities to have caused the death or expansion of the infrastructure. These activities from one of the AGP blocks, according to the actors resulted in effects on the other blocks and had negative or positive effects on the working of the infrastructure. The actors perceive these activities as good action or bad action, based on their sentience or ability to experience effects of infrastructuring work. In essence Karma gives a sense of motivation to the actors to stabilize or destabilize the infrastructure, but then II theory argues that there is never really a start or end to an infrastructure. Only an instance of what users perceive as new or old infrastructure that is bootstrapped on existing installed base of people, processes and technology.

The AGP model brings focus on infrastructuring work and allows the researcher to understand how the work can be organized. The complexities are managed through the

proxy of work done by the components of technology and people. The AGP model brings the focus on work, but ignores aspects such as counter-networks or external forces that result in changes to the infrastructure. Without expanding the boundaries of the infrastructure to cover these forces, the AGP model cannot explain how these forces had a causality on the change in one of the architectural blocks or the overall infrastructure. The AGP model is also in some sense macro-level view and hence is described as a genus-level taxonomy. There are higher and lower level taxonomies which could describe the work within architecture, governance and praxis categories. As an example, I highlight that the Action-design research methodology provides a more micro-level management principle to cover design of infrastructures. It also provides rules or processes that need to be followed to release a technology product that can be a component in the infrastructure. The AGP model in such aspects creates blind spots, particularly in micro-level methodological work process inside each of the infrastructuring work category. The AGP model provides a taxonomy that is not a guideline to follow to create a successful infrastructure. The infrastructuring work in each category still needs to follow its own unique methodology to relate to the work being done in the other categories.

Beyond the AGP model and the aspects that it highlights or the blind spots that it creates, the thesis makes practical contributions to a number of aspects of infrastructuring work. It suggests architectural strategy of matching mHealth to existing infrastructures for improved scaling. The thesis prescribes the use of appropriate research methods and theories to understand local context and needs. The thesis puts forth a tweaked agile methodology that can improve knowledge sharing in open-source communities. It also redefines big data in terms of organizational capabilities and suggests practical ways in which cloud computing models could enable sustainable use of analytics tools in LMICs.

The thesis also hopes to motivate the reader to expand this taxonomy to deeper levels of classification. Particularly interesting could be the suggestion of appropriate methods in each of the blocks of AGP model. Another interesting avenue could be to explore cladistics using this taxonomy, so that infrastructures could be classified to come from a particular pedigree or ancestry.

References

- Aanestad, M., & Jensen, T. B. (2011). Building nation-wide information infrastructures in healthcare through modular implementation strategies. *The Journal of Strategic Information Systems*, 20(2), 161–176. doi:10.1016/j.jsis.2011.03.006
- Abbate, J. (1999). *Inventing the Internet*. MIT Press.
- Alexander, C. (1979). *The Timeless Way of Building*. Oxford: Oxford Univ. Press.
- Asangansi, I., & Braa, K. (2010). The emergence of mobile-supported national health information systems in developing countries. *Studies in Health Technology and Informatics*, 160, 540.
- Attaran, A. (2005). An immeasurable crisis? A criticism of the millennium development goals and why they cannot be measured. *PLoS Medicine*, 2(10), e318.
- Bass, L. (2007). *Software architecture in practice*. Pearson Education India.
- Beck, K., Beedle, M., Van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., ... others. (2001). Manifesto for agile software development.
- Bevir, M. (2013). Legitimacy and the Administrative State: Ontology, History, and Democracy. *Public Administration Quarterly*, 37(4), 535.
- Boehm, B. (2002). Get ready for agile methods, with care. *Computer*, 35(1), 64–69. doi:10.1109/2.976920
- Booth, P., Matolcsy, Z., & Wieder, B. (2000). The Impacts of Enterprise Resource Planning Systems on Accounting Practice – The Australian Experience. *Australian Accounting Review*, 10(22), 4–18. doi:10.1111/j.1835-2561.2000.tb00066.
- Borgman, C. L. (n.d.). *From Gutenberg to the Global Information Infrastructure: Access to Information in the Networked World*. MIT Press.
- Bosch, J. (2004). Software Architecture: The Next Step. In F. Oquendo, B. C. Warboys, & R. Morrison (Eds.), *Software Architecture* (pp. 194–199). Springer Berlin Heidelberg. Retrieved from http://link.springer.com/chapter/10.1007/978-3-540-24769-2_14
- Bowker, G. C. (1994). *Science on the Run: Information Management and Industrial Geophysics at Schlumberger, 1920-1940*. MIT Press.
- Bowker, G. C., & Star, S. L. (2000). *Sorting Things Out: Classification and Its Consequences*. MIT Press.
- Bowker, G. C., Baker, K., Millerand, F., & Ribes, D. (2010). Toward Information Infrastructure Studies: Ways of Knowing in a Networked Environment. In J. Hunsinger, L. Klasttrup, & M. Allen (Eds.), *International Handbook of Internet Research* (pp. 97–117). Springer Netherlands.

- Braa, J., & Hedberg, C. (2002). The struggle for district-based health information systems in South Africa. *The Information Society*, 18(2), 113–127.
- Braa, J., & Sahay, S. (2012). *Integrated Health Information Architecture: Power to the Users : Design, Development, and Use*. Matrix Publishers.
- Braa, J., Hanseth, O., Heywood, A., Mohammed, W., & Shaw, V. (2007). Developing health information systems in developing countries: the flexible standards strategy. *MIS Quarterly*, 31(2), 381–402.
- Braa, J., Monteiro, E., & Sahay, S. (2004). Networks of Action: Sustainable Health Information Systems across Developing Countries. *MIS Quarterly*, 28(3), 337–362.
- Braa, K., & Purkayastha, S. (2010). Sustainable mobile information infrastructures in low resource settings. *Studies in Health Technology and Informatics*, 157, 127.
- Buyya, R., Yeo, C. S., Venugopal, S., Broberg, J., & Brandic, I. (2009). Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility. *Future Generation Computer Systems*, 25(6), 599–616. doi:10.1016/j.future.2008.12.001
- Bygstad, B. (2010). Generative mechanisms for innovation in information infrastructures. *Information and Organization*, 20(3–4), 156–168. doi:10.1016/j.infoandorg.2010.07.001
- Cahill, C. (2007). Doing Research with Young People: Participatory Research and the Rituals of Collective Work. *Children's Geographies*, 5(3), 297–312. doi:10.1080/14733280701445895
- Cao, L., Mohan, K., Xu, P., & Ramesh, B. (2009). A framework for adapting agile development methodologies. *European Journal of Information Systems*, 18(4), 332–343. doi:10.1057/ejis.2009.26
- Castells, M. (1996). *The Information Age: Economy, Society, and Culture. Volume I: The Rise of the Network Society*. Blackwell.
- Cavelty, M. D. (2007). Critical information infrastructure: vulnerabilities, threats and responses. In *Disarmament Forum* (Vol. 3, pp. 15–22).
- Chilundo, B., & Aanestad, M. (2005). Negotiating multiple rationalities in the process of integrating the information systems of disease specific health programmes. *EJISDC: The Electronic Journal on Information Systems in Developing Countries*, (20), 2.
- Ciborra, C. (1997). De profundis? Deconstructing the concept of strategic alignment. *Scandinavian Journal of Information Systems*, 9(1).
- Ciborra, C. U. (1992). From thinking to tinkering: The grassroots of strategic information systems. *The Information Society*, 8(4), 297–309. doi:10.1080/01972243.1992.9960124
- Ciborra, C. U., & Hanseth, O. (1998). From tool to Gestell: Agendas for managing the information infrastructure. *Information Technology & People*, 11(4), 305–327.

- Ciborra, C. U., Braa, K., Cordella, A., Dahlbom, B., Failla, A., Hanseth, O., ... Monteiro, E. (2000). *From Control to Drift: The Dynamics of Corporate Information Infrastructures*. Oxford: Oxford University Press.
- Conboy, K. (2009). Agility from First Principles: Reconstructing the Concept of Agility in Information Systems Development. *Information Systems Research*, 20(3), 329–354. doi:10.1287/isre.1090.0236
- Constantinides, P., & Barrett, M. (2006). Large-Scale ICT Innovation, Power, and Organizational Change The Case of a Regional Health Information Network. *The Journal of Applied Behavioral Science*, 42(1), 76–90. doi:10.1177/0021886305284291
- Crowston, K., Wei, K., Howison, J., & Wiggins, A. (2008). Free/Libre Open-source Software Development: What We Know and What We Do Not Know. *ACM Comput. Surv.*, 44(2), 7:1–7:35. doi:10.1145/2089125.2089127
- Dahlberg, L. (2001). Computer-Mediated Communication and The Public Sphere: A Critical Analysis. *Journal of Computer-Mediated Communication*, 7(1), 0–0. doi:10.1111/j.1083-6101.2001.tb00137.x
- David, P. A. (1987). Some new standards for the economics of standardization in the information age. In *Economic policy and technological performance*. Cambridge University Press. Retrieved from <http://dx.doi.org/10.1017/CBO9780511559938.010>
- Dedrick, J., & Kraemer, K. L. (1995). National technology policy and computer production in Asia-pacific countries. *The Information Society*, 11(1), 29–58. doi:10.1080/01972243.1995.9960178
- Detmer, D. E. (2003). Building the national health information infrastructure for personal health, health care services, public health, and research. *BMC Medical Informatics and Decision Making*, 3(1), 1.
- Dunn, M. (2005). The socio-political dimensions of critical information infrastructure protection (CIIP). *International Journal of Critical Infrastructures*, 1(2), 258–268.
- Earl, M. J. (1996). The chief information officer: Past, present, and future. *Information Management: The Organizational Dimension*, 456–484.
- Edwards, P., Bowker, G., Jackson, S., & Williams, R. (2009). Introduction: An Agenda for Infrastructure Studies. *Journal of the Association for Information Systems*, 10(5).
- El-Atawy, A. (2006). Survey on the use of formal languages/models for the specification, verification, and enforcement of network access-lists. *School of Computer Science, Telecommunication, and Information Systems, DePaul University, Chicago*
- Elden, M., & Chisholm, R. F. (1993). Emerging Varieties of Action Research: Introduction to the Special Issue. *Human Relations*, 46(2), 121–142. doi:10.1177/001872679304600201
- Ellingsen, G., & Monteiro, E. (2006). Seamless integration: Standardisation across multiple local settings. *Computer Supported Cooperative Work (CSCW)*, 15(5), 443–466.

- Ellingsen, G., Monteiro, E., & Røed, K. (2013). Integration as interdependent workaround. *International Journal of Medical Informatics*, 82(5), e161–e169. doi:10.1016/j.ijmedinf.2012.09.004
- Endres, A., & Rombach, H. D. (2003). *A handbook of software and systems engineering: empirical observations, laws and theories*. Pearson Education.
- Evans, J. D. (1999). Organizational and Technological Interoperability for Geographic Information Infrastructures. In M. Goodchild, M. Egenhofer, R. Fegeas, & C. Kottman (Eds.), *Interoperating Geographic Information Systems* (pp. 401–414). Springer US.
- Fitzgerald, B. (2006). The transformation of open source software. *Mis Quarterly*, 587–598.
- Fitzgerald, B., Hartnett, G., & Conboy, K. (2006). Customising agile methods to software practices at Intel Shannon. *Eur. J. Inf. Syst.*, 15(2), 200–213. doi:10.1057/palgrave.ejis.3000605
- Fleck, J. (1994). Learning by trying: the implementation of configurational technology. *Research Policy*, 23(6), 637–652. doi:10.1016/0048-7333(94)90014-0
- Fleetwood, S. (2009). The Ontology of Things, Properties and Powers. *Journal of Critical Realism*, 8(3), 343–366. doi:10.1558/jocr.v8i3.343
- Foster, I., & Kesselman, C. (2003). *The Grid 2: Blueprint for a New Computing Infrastructure*. Elsevier.
- Freire, P. (1985). *The Politics of Education: Culture, Power, and Liberation*. Greenwood Publishing Group.
- Gal, U., Lyytinen, K., & Yoo, Y. (2008). The dynamics of IT boundary objects, information infrastructures, and organisational identities: the introduction of 3D modelling technologies into the architecture, engineering, and construction industry. *European Journal of Information Systems*, 17(3), 290–304. doi:10.1057/ejis.2008.13
- Gandhi, M. K. (2001). The gospel of selfless action (M. Desai, Trans.) Ahmedabad. Navajivan Publishing House. (Original work published 1946)
- Grimson, J., Grimson, W., & Hasselbring, W. (2000). The SI challenge in health care. *Commun. ACM*, 43(6), 48–55. doi:10.1145/336460.336474
- Grindley, P. (1995). *Standards, Strategy, and Policy: Cases and Stories*. Oxford England ; New York: OUP Oxford.
- Grisot, M., Hanseth, O., & Thorseng, A. (2014). Innovation Of, In, On Infrastructures: Articulating the Role of Architecture in Information Infrastructure Evolution. *Journal of the Association for Information Systems*, 15(4).
- Halamka, J., Overhage, J. M., Ricciardi, L., Rishel, W., Shirky, C., & Diamond, C. (2005). Exchanging Health Information: Local Distribution, National Coordination. *Health Affairs*, 24(5), 1170–1179. doi:10.1377/hlthaff.24.5.1170

- Hanseth, O., & Aanestad, M. (2003). Design as bootstrapping. On the evolution of ICT networks in health care. *Methods of Information in Medicine*, 42(4), 385–391. doi:10.1267/METH03040385
- Hanseth, O., & Braa, K. (1998). Technology As Traitor: Emergent SAP Infrastructure in a Global Organization. In *Proceedings of the International Conference on Information Systems* (pp. 188–196). Atlanta, GA, USA: Association for Information Systems.
- Hanseth, O., & Ciborra, C. (2007). *Risk, complexity and ICT*. Edward Elgar Publishing.
- Hanseth, O., & Lyytinen, K. (2010). Design theory for dynamic complexity in information infrastructures: the case of building internet. *Journal of Information Technology*, 25(1), 1–19.
- Hanseth, O., & Monteiro, E. (1997). Inscribing behaviour in information infrastructure standards. *Accounting Management and Information Technologies*, 7, 183–212.
- Hanseth, O., & Monteiro, E. (2001). Understanding information infrastructure. (unfinished book manuscript), www.ifi.uio.no/~oleha/Publications/bok.html
- Hanseth, O., Jacucci, E., Grisot, M., & Aanestad, M. (2006). Reflexive standardization: side effects and complexity in standard making. *MIS Quarterly*, 30(1), 563–581.
- Hanseth, O., Jacucci, E., Grisot, M., & Aanestad, M. (2006). Reflexive standardization: side effects and complexity in standard making. *MIS Q.*, 30(1), 563–581.
- Hanseth, O., Monteiro, E., & Hatling, M. (1996). Developing information infrastructure: The tension between standardization and flexibility. *Science, Technology & Human Values*, 21(4), 407.
- Haux, R. (2006). Health information systems – past, present, future. *International Journal of Medical Informatics*, 75(3–4), 268–281. doi:10.1016/j.ijmedinf.2005.08.002
- Health Metrics Network, & World Health Organization. (2008). Framework and standards for country health information systems. Retrieved from <http://apps.who.int/iris/handle/10665/43872>
- Heeks, R. (2002). Information systems and developing countries: Failure, success, and local improvisations. *The Information Society*, 18(2), 101–112.
- Henfridsson, O., & Bygstad, B. (2013). The Generative Mechanisms of Digital Infrastructure Evolution. *Management Information Systems Quarterly*, 37(3), 896–931.
- Henfridsson, O., & Bygstad, B. (2013). The Generative Mechanisms of Digital Infrastructure Evolution. *Management Information Systems Quarterly*, 37(3), 896–931.
- Henningsson, S., & Hanseth, O. (2011). The Essential Dynamics of Information Infrastructures. *ICIS 2011 Proceedings*.
- Henningsson, S., & Hedman, J. (2014). Transformation of Digital Ecosystems: The Case of Digital Payments. In Linawati, M. S. Mahendra, E. J. Neuhold, A. M. Tjoa, & I. You (Eds.), *Information and Communication Technology* (pp. 46–55). Springer Berlin Heidelberg.

- Herbsleb, J. D. (2007). Global Software Engineering: The Future of Socio-technical Coordination. In *2007 Future of Software Engineering* (pp. 188–198). Washington, DC, USA: IEEE Computer Society. doi:10.1109/FOSE.2007.11
- Highsmith, J., & Cockburn, A. (2001). Agile software development: the business of innovation. *Computer*, 34(9), 120–127. doi:10.1109/2.947100
- Holmström, H., Fitzgerald, B., Ågerfalk, P. J., & Conchúir, E. Ó. (2006). Agile Practices Reduce Distance in Global Software Development. *Information Systems Management*, 23(3), 7–18. doi:10.1201/1078.10580530/46108.23.3.20060601/93703.2
- Hughes, T. P. (1989). The evolution of large technological systems. In W. E. Bijker, T. P. Hughes, & T. Pinch (Eds.), *The social construction of technological systems* (pp. 51–82). Cambridge, MA: MIT Press.
- Hughes, T. P. (1993). *Networks of Power: Electrification in Western Society, 1880–1930*. JHU Press.
- Imhoff, C. (2005). Streamlining Processes for Front-Line Workers: Adding Business Intelligence for Operations. *Operational Business Intelligence White Paper, Intelligent Solutions Inc.*
- Information Infrastructure Task Force (IITF), (1995). *Privacy and the National Information Infrastructure: Principles of providing and using personal information*. Retrieved from: http://www.iitf.nist.gov/documents/committee/infopol/niiprivprin_final.html
- J. Nawrocki and A. Wojciechowski, "Experimental Evaluation of Pair Programming," in European Software Control and Metrics (ESCOM 2001), London, England, 2001.
- Jalali, S., & Wohlin, C. (2012). Global software engineering and agile practices: a systematic review. *Journal of Software: Evolution and Process*, 24(6), 643–659. doi:10.1002/smr.561
- Jha, A. K., Doolan, D., Grandt, D., Scott, T., & Bates, D. W. (2008). The use of health information technology in seven nations. *International Journal of Medical Informatics*, 77(12), 848–854. doi:10.1016/j.ijmedinf.2008.06.007
- Joslyn, C., & Rocha, L. (2000). Towards semiotic agent-based models of socio-technical organizations. In *Proc. AI, Simulation and Planning in High Autonomy Systems (AIS 2000) Conference, Tucson, Arizona* (pp. 70–79).
- Kahin, B., & Abbate, J. (1995). *Standards Policy for Information Infrastructure*. MIT Press.
- Kahin, B., & Nesson, C. (Eds.). (1996). *Borders in Cyberspace: Information Policy and the Global Information Infrastructure* (1st ed.). Cambridge, MA, USA: MIT Press.
- Kapleau, P. (1989). *The Wheel of Life and Death: A Practical and Spiritual Guide*. Doubleday.
- Karasti, H., & Baker, K. S. (2004). Infrastructuring for the long-term: ecological information management. In *Proceedings of the 37th Annual Hawaii International Conference on System Sciences, 2004* (p. 10 pp.–). doi:10.1109/HICSS.2004.1265077

- Kaushal, R., Blumenthal, D., Poon, E. G., Jha, A. K., Franz, C., Middleton, B., ... Fernandopulle, R. (2005). The Cost of a National Health Information Network. *Annals of Internal Medicine*, 143(3), 165.
- Kemmis, S. (2001). Exploring the relevance of critical theory for action research: Emancipatory action research in the footsteps of Jurgen Habermas. *Handbook of Action Research: Participative Inquiry and Practice*, 91–102.
- Keny, P., & Chemburkar, A. (2006). Trends in Operational BI. *DM REVIEW*, 16(7), 20.
- Kling, R. (1980). Social Analyses of Computing: Theoretical Perspectives in Recent Empirical Research. *ACM Comput. Surv.*, 12(1), 61–110. doi:10.1145/356802.356806
- Kobielus, J. (2009). Mighty mashups: do-it-yourself business intelligence for the new economy. *Forrester Research*.
- Koch, S. (2004). Agile Principles and Open Source Software Development: A Theoretical and Empirical Discussion. In J. Eckstein & H. Baumeister (Eds.), *Extreme Programming and Agile Processes in Software Engineering* (Vol. 3092, pp. 85–93). Springer Berlin / Heidelberg.
- Krishnamurthy, S., Ou, S., & Tripathi, A. K. (2014). Acceptance of monetary rewards in open source software development. *Research Policy*, 43(4), 632–644. doi:10.1016/j.respol.2013.10.007
- Lakhani, K., & Wolf, R. (2003). Why hackers do what they do: Understanding motivation and effort in free/open source software projects.
- Latour, B. (1996). On actor-network theory: A few clarifications. *Soziale Welt*, 47(4), 369–381.
- Latour, B. (1997). Socrates' and Callicles' Settlement--or, The Invention of the Impossible Body Politic. *Configurations*, 5(2), 189–240. doi:10.1353/con.1997.0011
- Latour, B. (2007). Turning around Politics: A Note on Gerard de Vries' Paper. *Social Studies of Science*, 37(5), 811–820.
- Laurin, M. (2010). Assessment of the Relative Merits of a Few Methods to Detect Evolutionary Trends. *Systematic Biology*, 59(6), 689–704. doi:10.1093/sysbio/syq059
- Law, J. (1992). Notes on the theory of the actor-network: Ordering, strategy, and heterogeneity. *Systems Practice*, 5(4), 379–393. doi:10.1007/BF01059830
- Lee, G., DeLone, W., & Espinosa, J. A. (2006). Ambidextrous Coping Strategies in Globally Distributed Software Development Projects. *Commun. ACM*, 49(10), 35–40. doi:10.1145/1164394.1164417
- Lungu, I., Bâra, A., & Fodor, A. (2006). Business Intelligence tools for building the Executive Information Systems. In *5thRoEduNet International Conference, Lucian Blaga University, Sibiu*.

- Lyytinen, K., & Damsgaard, J. (2001). What's wrong with the Diffusion of Innovation Theory? In M. A. Ardis & B. L. Marcolin (Eds.), *Diffusing Software Product and Process Innovations* (pp. 173–190). Springer US.
- Manda, T. D., & Sanner, T. A. (2014). The Mobile is Part of a Whole: Implementing and Evaluating mHealth from an Information Infrastructure Perspective. *International Journal of User-Driven Healthcare*, 4(1), 1–16. doi:10.4018/ijudh.2014010101
- Mangalaraj, G., Mahapatra, R., & Nerur, S. (2009). Acceptance of software process innovations – the case of extreme programming. *European Journal of Information Systems*, 18(4), 344–354. doi:10.1057/ejis.2009.23
- Manovich, L. (2011). Trending: The Promises and the Challenges of Big Social Data. *Debates in the Digital Humanities*.
- Markus, M. L. (1983). Power, Politics, and MIS Implementation. *Commun. ACM*, 26(6), 430–444. doi:10.1145/358141.358148
- Matavire, R., & Manda, T. D. (2014). Intervention Breakdowns as Occasions for Articulating Mobile Health Information Infrastructures. *The Electronic Journal of Information Systems in Developing Countries*, 0(0).
- Maxwell, J. A. (2012). *Qualitative Research Design: An Interactive Approach: An Interactive Approach*. SAGE.
- Mayer-Schonberger, V., & Foster, T. E. (1996). A Regulatory Web: Free Speech and the Global Information Infrastructure. *Michigan Telecommunications and Technology Law Review*, 3, 45.
- McGarty, T. P. (1992). Alternative networking architectures: Pricing, policy, and competition. *Building Information Infrastructure. McGraw-Hill Primis*.
- McLuhan, Marshall (1967) *Understanding Media: The Extensions of Man*, 3rd edn. London: Routledge and Kegan Paul.
- Mengiste, S. A. (2010). Analysing the Challenges of IS implementation in public health institutions of a developing country: the need for flexible strategies. *Journal of Health Informatics in Developing Countries*, 4(1).
- Meso, P., Musa, P., Straub, D., & Mbarika, V. (2009). Information infrastructure, governance, and socio-economic development in developing countries. *European Journal of Information Systems*, 18(1), 52–65. doi:10.1057/ejis.2008.56
- Miles, M. B., & Huberman, A. M. (1994). *Qualitative Data Analysis: An Expanded Sourcebook*. SAGE.
- Miscione, G., & Staring, K. (2009). Shifting Ground for Health Information Systems: Local Embeddedness, Global Fields, and Legitimation. *International Journal of Sociotechnology and Knowledge Development (IJSKD)*, 1(4), 1–12.

- Mockus, A. (2009). Succession: Measuring transfer of code and developer productivity. In *IEEE 31st International Conference on Software Engineering, 2009. ICSE 2009* (pp. 67–77). doi:10.1109/ICSE.2009.5070509
- Mockus, A., Fielding, R. T., & Herbsleb, J. D. (2002). Two case studies of open source software development: Apache and Mozilla. *ACM Trans. Softw. Eng. Methodol.*, 11(3), 309–346. doi:10.1145/567793.567795
- Moe, N. B., Dingsøyr, T., & Dybå, T. (2010). A teamwork model for understanding an agile team: A case study of a Scrum project. *Information and Software Technology*, 52(5), 480–491. doi:10.1016/j.infsof.2009.11.004
- Monteiro, E. (1998). *Diffusion of Infrastructure: Mobilization And Improvisation*.
- Monteiro, E. (2000). Actor-network theory and information infrastructure. *From Control to Drift*, 71–83.
- Monteiro, E. (2000). Actor-network theory and information infrastructure. *From Control to Drift*, 71–83.
- Monteiro, E. (2003). Integrating health information systems: a critical appraisal. *Methods of Information in Medicine*, 42(4), 428–432. doi:10.1267/METH03040428
- Moodley, D., Pillay, A. W., & Seebregts, C. J. (2012). Position Paper: Researching and Developing Open Architectures for National Health Information Systems in Developing African Countries. In Z. Liu & A. Wassing (Eds.), *Foundations of Health Informatics Engineering and Systems* (Vol. 7151, pp. 129–139). Berlin, Heidelberg: Springer Berlin Heidelberg.
- Mosse, E. L., & Sahay, S. (2003). Counter Networks, Communication and Health Information Systems. In M. Korpela, R. Montealegre, & A. Poulymenakou (Eds.), *Organizational Information Systems in the Context of Globalization* (pp. 35–51). Springer US.
- Mukherjee, A., Purkayastha, S., & Sahay, S. (2010). Exploring the potential and challenges of using mobile based technology in strengthening health information systems: Experiences from a pilot study. *AMCIS 2010 Proceedings*, 263.
- Mulla, Z. R., & Krishnan, V. R. (2008). Karma-Yoga, the Indian Work Ideal, and its Relationship with Empathy. *Psychology & Developing Societies*, 20(1), 27–49. doi:10.1177/097133360702000102
- Murray, C. J. L., Laakso, T., Shibuya, K., Hill, K., & Lopez, A. D. (2007). Can we achieve Millennium Development Goal 4? New analysis of country trends and forecasts of under-5 mortality to 2015. *Lancet*, 370(9592), 1040–1054.
- Nakicenovic, N., & Grübler, A. (Eds.). (1991). *Diffusion of Technologies and Social Behavior* (Softcover reprint of the original 1st ed. 1991 edition.). Berlin, Heidelberg: Springer.
- Nerur, S., Mahapatra, R., & Mangalaraj, G. (2005). Challenges of Migrating to Agile Methodologies. *Communications of ACM*, 48(5), 72–78. doi:10.1145/1060710.1060712

- O'Brien, R. C. (1991). Brief case: EIS and strategic control. *Long Range Planning*, 24(5), 125–127.
- Ohlhorst, F. J. (2012). *Big data analytics: turning big data into big money*. John Wiley & Sons.
- Okuonzi, S. A., & Macrae, J. (1995). Whose policy is it anyway? International and national influences on health policy development in Uganda. *Health Policy and Planning*, 10(2), 122–132. doi:10.1093/heapol/10.2.122
- Ole Hanseth, Margunn Aanestad, & Marc Berg. (2004). Guest editors' introduction: Actor-network theory and information systems. What's so special? *Information Technology & People*, 17(2), 116–123. doi:10.1108/09593840410542466
- Orlikowski, W. J. (1991). Integrated information environment or matrix of control? The contradictory implications of information technology. *Accounting, Management and Information Technologies*, 1(1), 9–42. doi:10.1016/0959-8022(91)90011-3
- Orlikowski, W. J. (1992). The Duality of Technology: Rethinking the Concept of Technology in Organizations. *Organization Science*, 3(3), 398–427. doi:10.1287/orsc.3.3.398
- Orlikowski, W. J., & Robey, D. (1991). Information technology and the structuring of organizations. *Working Paper (Sloan School of Management)*; 3284-91.
- Orlikowski, W., & Hoffman, D. (1997). An Improvisational Model for Change Management: The Case of Groupware Technologies. *Inventing the Organizations of the 21st Century*, MIT, Boston, MA, 265–282.
- Paasivaara, M., & Lassenius, C. (2006). Could Global Software Development Benefit from Agile Methods? In *International Conference on Global Software Engineering, 2006. ICGSE '06* (pp. 109–113). doi:10.1109/ICGSE.2006.261222
- Perritt, H. H. J. (1996). Property and Innovation in the Global Information Infrastructure. *University of Chicago Legal Forum*, 1996, 261.
- Pipek, V., & Wulf, V. (2009). Infrastructuring: Toward an Integrated Perspective on the Design and Use of Information Technology. *Journal of the Association for Information Systems*, 10(5).
- Pironti, J. (2006). Key Elements of a Threat and Vulnerability Management Program. *ISACA Journal*, 3(2006).
- Porter-O'Grady, T., Hawkins, M. A., & MBA.), M. L. P. (RN. (1997). *Whole-Systems Shared Governance: Architecture for Integration*. Jones & Bartlett Learning.
- Prakash, A., & De', R. (2007). Importance of development context in ICT4D projects: A study of computerization of land records in India. *Information Technology & People*, 20(3), 262–281. doi:10.1108/09593840710822868
- Puri, S. K., Sahay, S., & Lewis, J. (2009). Building participatory HIS networks: A case study from Kerala, India. *Information and Organization*, 19(2), 63–83. doi:10.1016/j.infoandorg.2008.06.002

- Purkayastha, S. (2011). Towards a contextual insecurity framework: How contextual development leads to security problems in information systems. In *Proceedings of IRIS 2011* (pp. 654–666). Turku Centre for Computer Science.
- Purkayastha, S. (2012). HIXEn: An integration engine for multi-vocabulary health information using REST & semantic metadata mapping. In *2012 World Congress on Information and Communication Technologies (WICT)* (pp. 679–684). doi:10.1109/WICT.2012.6409162
- Purkayastha, S., & Braa, J. (2013). Big Data Analytics for developing countries – Using the Cloud for Operational BI in Health. *The Electronic Journal of Information Systems in Developing Countries*, 59(0).
- Purkayastha, S., Manda, T. D., & Sanner, T. A. (2013). A Post-development Perspective on mHealth – An Implementation Initiative in Malawi. In *2013 46th Hawaii International Conference on System Sciences (HICSS)* (pp. 4217–4225). doi:10.1109/HICSS.2013.53
- Ravitch, S. M., & Riggan, M. (2011). *Reason & Rigor: How Conceptual Frameworks Guide Research*. SAGE.
- Rhodes, R. A. W. (1997). *Understanding governance: policy networks, governance, reflexivity and accountability*. Open University Press.
- Ribes, D. (2014). The Kernel of a Research Infrastructure. In *Proceedings of the 17th ACM Conference on Computer Supported Cooperative Work & Social Computing* (pp. 574–587). New York, NY, USA: ACM. doi:10.1145/2531602.2531700
- Ribes, D., & Finholt, T. (2009). The Long Now of Technology Infrastructure: Articulating Tensions in Development. *Journal of the Association for Information Systems*, 10(5). Retrieved from <http://aisel.aisnet.org/jais/vol10/iss5/5>
- Robey, D., & Markus, M. L. (1984). Rituals in Information System Design. *MIS Quarterly*, 8(1), 5–15. doi:10.2307/249240
- Robson, C. (1993). *Real world research: A resource for social scientists and practitioner-researchers*. Blackwell Oxford.
- Rolland, K. H., & Monteiro, E. (2002). Balancing the local and the global in infrastructural information systems. *The Information Society*, 18(2), 87–100.
- Rolland, K., & Aanestad, M. (2003). The techno-political dynamics of information infrastructure development: Interpreting two cases of puzzling evidence. In *26th Information Systems Research Seminar in Scandinavia (IRIS) Porvoo, Finland*.
- Rosenberg, N. (1994). *Exploring the Black Box: Technology, Economics, and History*. Cambridge University Press.
- Roth, W.-M., & Lee, S. (2004). Science education as/for participation in the community. *Science Education*, 88(2), 263–291. doi:10.1002/sce.10113
- Rothkopf, D. (1997). In Praise of Cultural Imperialism? *Foreign Policy*, (107), 38–53. doi:10.2307/1149331

- Russo, B., Scotto, M., Sillitti, A., & Succi, G. (2009). *Agile Technologies in Open Source Development*. Hershey, PA: Information Science Reference - Imprint of: IGI Publishing.
- Sæbø, J. I., Kossi, E. K., Titlestad, O. H., Tohouri, R. R., & Braa, J. (2011). Comparing strategies to integrate health information systems following a data warehouse approach in four countries. *Information Technology for Development*, 17(1), 42–60.
- Sahay, S., Monteiro, E., & Aanestad, M. (2007). Towards a Political Perspective of Integration in IS Research: the case of Health Information Systems in India. In *9th International Conference on Social Implementation of Computers in Developing Countries*. Sao Paulo, Brazil.
- Sanner, T., Manda, T., & Nielsen, P. (2014). Grafting: Balancing Control and Cultivation in Information Infrastructure Innovation. *Journal of the Association for Information Systems*, 15(4).
- Sayer, R. A. (1992). *Method in Social Science: A Realist Approach*. Psychology Press.
- Scacchi, W. (2007). Free/open source software development: recent research results and emerging opportunities. In *The 6th Joint Meeting on European software engineering conference and the ACM SIGSOFT symposium on the foundations of software engineering: companion papers* (pp. 459–468). New York, NY, USA: ACM. doi:10.1145/1295014.1295019
- Schrag, C. O. (1999). *The Self after Postmodernity* (New edition edition.). New Haven; London: Yale University Press.
- Schulz, M. S. (1998). Collective Action across Borders: Opportunity Structures, Network Capacities, and Communicative Praxis in the Age of Advanced Globalization. *Sociological Perspectives*, 41(3), 587–616. doi:10.2307/1389565
- Schwartz, R. (1992). Software industry entry strategies for developing countries: A “walking on two legs” proposition. *World Development*, 20(2), 143–164. doi:10.1016/0305-750X(92)90096-E
- Schweiger, A., Sunyaev, A., Leimeister, J. M., & Krcmar, H. (2007). Information Systems and Healthcare XX: Toward Seamless Healthcare with Software Agents. *Communications of the Association for Information Systems*, 19(1).
- Scott, J. C. (1998). *Seeing Like a State: How Certain Schemes to Improve the Human Condition Have Failed*. Yale University Press.
- Scott, S. V., & Zachariadis, M. (2013). *SWIFT: Cooperative Governance for Network Innovation, Standards, and Community*. Routledge.
- Scott, W. R. (2001). *Institutions and organizations*. Sage Publications.
- Sein, M. K., Henfridsson, O., Purao, S., Rossi, M., & Lindgren, R. (2011). Action Design Research. *MIS Q.*, 35(1), 37–56.
- Seo, M.-G., & Creed, W. E. D. (2002). Institutional Contradictions, Praxis, and Institutional Change: A Dialectical Perspective. *Academy of Management Review*, 27(2), 222–247. doi:10.5465/AMR.2002.6588004

- Shaw, M., & Garlan, D. (1996). *Software Architecture: Perspectives on an Emerging Discipline*. Upper Saddle River, N.J: Prentice Hall.
- Singh, P. V. (2010). The small-world effect: The influence of macro-level properties of developer collaboration networks on open-source project success. *ACM Trans. Softw. Eng. Methodol.*, 20(2), 6:1–6:27. doi:10.1145/1824760.1824763
- Sørensen, E. (2005). the democratic problems and potentials of network governance. *European Political Science*, 4(3), 348–357. doi:10.1057/palgrave.eps.2210033
- Spender, J.-C. (1996). Making knowledge the basis of a dynamic theory of the firm. *Strategic Management Journal*, 17(S2), 45–62. doi:10.1002/smj.4250171106
- Stansfield, S., Orobato, N., Lubinski, D., Uggowitzer, S., & Mwanyika, H. (2008). The Case for a National Health Information System Architecture: a Missing Link to Guiding National Development and Implementation. *Making the eHealth Connection, Bellagio*.
- Star, S. L., & Griesemer, J. R. (1989). Institutional Ecology, 'Translations' and Boundary Objects: Amateurs and Professionals in Berkeley's Museum of Vertebrate Zoology, 1907-39. *Social Studies of Science*, 19(3), 387–420. doi:10.1177/030631289019003001
- Star, S. L., & Ruhleder, K. (1996). Steps Toward an Ecology of Infrastructure: Design and Access for Large Information Spaces. *Information Systems Research*, 7(1), 111–134. doi:10.1287/isre.7.1.111
- Stephany, F., Mens, T., & Gırba, T. (2009). Maispion: a tool for analysing and visualising open source software developer communities. In *Proceedings of the International Workshop on Smalltalk Technologies* (pp. 50–57). New York, NY, USA: ACM. doi:10.1145/1735935.1735944
- Stephen M. Mutula, & Pieter van Brakel. (2006). E-readiness of SMEs in the ICT sector in Botswana with respect to information access. *The Electronic Library*, 24(3), 402–417. doi:10.1108/02640470610671240
- Stoops, N., Williamson, L., Krishna, S., & Madon, S. (2003). Using health information for local action: facilitating organisational change in South Africa. *The Digital Challenge: Information Technology in the Development Context*. Gateshead: Athenaem Press Ltd.
- TCOJ (Telecommunications Council of Japan), 1994. *Reforms toward the Intellectually Creative Society of the 21st. Society*. Tokyo, Japan: Ministry of Posts and Telecommunications, May 31, 1994.
- Tilak, B. G. (2000). *Srimad Bhagavadgita-Rahasya* (B.S. Sukhantar, Trans.). Poona. Kesari Press. (Original work published 1915)
- Tilson, D., Lyytinen, K., & Sørensen, C. (2010). Research Commentary—Digital Infrastructures: The Missing IS Research Agenda. *Information Systems Research*, 21(4), 748–759. doi:10.1287/isre.1100.0318
- Titlestad, O. H., Staring, K., & Braa, J. (2009). Distributed development to enable user participation. *Scandinavian Journal of Information Systems*, 21(1), 27–50.

- Titlestad, O., Staring, K., & Braa, J. (2009). Distributed Development to Enable User Participation: Multilevel design in the HISP network. *Scandinavian Journal of Information Systems*, 21(1).
- Toyama, K., & Dias, M. B. (2008). Information and Communication Technologies for Development. *Computer*, 41(6), 22–25. doi:10.1109/MC.2008.193
- Tsiknakis, M., Katehakis, D. G., & Orphanoudakis, S. C. (2002). An open, component-based information infrastructure for integrated health information networks. *International Journal of Medical Informatics*, 68(1), 3–26.
- Ure, J., Procter, R., Lin, Y., Hartswood, M., Anderson, S., Lloyd, S., ... Ho, K. (2009). The Development of Data Infrastructures for eHealth: A Socio-Technical Perspective. *Journal of the Association for Information Systems*, 10(5).
- Urowitz, S., Wiljer, D., Apatu, E., Eysenbach, G., DeLenardo, C., Harth, T., ... Leonard, K. (2008). Is Canada ready for patient accessible electronic health records? A national scan. *BMC Medical Informatics and Decision Making*, 8(1), 33. doi:10.1186/1472-6947-8-33
- Walsham, G. (1993). *Interpreting Information Systems in Organizations* (1st ed.). New York, NY, USA: John Wiley & Sons, Inc.
- Walsham, G. (1995). Interpretive case studies in IS research: nature and method. *European Journal of Information Systems*, 4(2), 74–81.
- Walsham, G., & Sahay, S. (2006). Research on information systems in developing countries: Current landscape and future prospects. *Information Technology for Development*, 12(1), 7–24. doi:10.1002/itdj.20020
- Warsta, J., & Abrahamsson, P. (2003). Is open source software development essentially an agile method. In *Proceedings of the 3rd Workshop on Open Source Software Engineering* (pp. 143–147).
- Watson, H. J., & Wixom, B. H. (2007). The current state of business intelligence. *Computer*, 40(9), 96–99.
- Watson, H. J., Wixom, B. H., Hoffer, J. A., Anderson-Lehman, R., & Reynolds, A. M. (2006). Real-Time Business Intelligence: Best Practices at Continental Airlines. *Information Systems Management*, 23(1), 7–18. doi:10.1201/1078.10580530/45769.23.1.20061201/91768.2
- White, C. (2005). The next generation of business intelligence: operational BI. *DM Review Magazine*.
- WHO, & ITU. (2012). *National eHealth strategy toolkit*. International Telecommunication Union. Retrieved from <http://apps.who.int/iris/handle/10665/75211>
- Williams, L., & Cockburn, A. (2003). Guest Editors' Introduction: Agile Software Development: It's about Feedback and Change. *Computer*, 36(6), 39–43.

- Williamson, L., Stoops, N., & Heywood, A. (2001). Developing a district health information system in South Africa: a social process or technical solution? *Studies in Health Technology and Informatics*, (1), 773–777.
- Win, D. T. (2008). Kamma and Chaos theory (Complexity science). *ABAC Journal*, (2008).
- Wixom, B. H., & Watson, H. J. (2001). An empirical investigation of the factors affecting data warehousing success. *MIS Q.*, 25(1), 17–32. doi:10.2307/3250957
- Wong, J., & Hong, J. (2008). What do we “mashup” when we make mashups? In *Proceedings of the 4th international workshop on End-user software engineering* (pp. 35–39). New York, NY, USA: ACM. doi:10.1145/1370847.1370855
- Xu, Y., Sauquet, D., Zapletal, E., Lemaitre, D., & Degoulet, P. (2000). Integration of medical applications: the “mediator service” of the SynEx platform. *International Journal of Medical Informatics*, 58–59, 157–166. doi:10.1016/S1386-5056(00)00084-8
- Yasnoff, W. A., Humphreys, B. L., Marc Overhage, J., Detmer, D. E., Flatley Brennan, P., Morris, R. W., ... Fanning, J. P. (2004). A consensus action agenda for achieving the national health information infrastructure. *Journal of the American Medical Informatics Association*, 11(4), 332–338. doi:10.1197/jamia.M1616
- Zachariadis, M., & Scott, S. V. (2014). *The Society for Worldwide Interbank Financial Telecommunication (SWIFT): cooperative governance for network innovation, standards, and community*. London, UK: Routledge.

Appendix A: Selected papers

- P1: Design and Implementation of Mobile-Based Technology in Strengthening Health Information System: Aligning mHealth Solutions to Infrastructures**
- P2: A Post-development Perspective on mHealth - An Implementation Initiative in Malawi**
- P3: OpenScrum - A Scrum methodology to increase bus factor in an open-source community**
- P4: Towards a contextual insecurity framework: How contextual development leads to security problems in information systems**
- P5: Overview, not overwhelm – Operational BI tools for Big Data in health information systems**
- P6: Big Data Analytics for developing countries – Using the Cloud for Operational BI in Health**

P1: Design & Implementation of mobile-based technology in strengthening health information system: Aligning mHealth solutions to Infrastructures

Saptarshi Purkayastha

Norwegian University of Science & Technology (NTNU), Norway

ABSTRACT

In the context of developing countries, there is a mounting interest in the field of mHealth. This surge in interest can be traced to the evolution of several interrelated trends (VW Consulting, 2009). However, with numerous attempts to create mobile-based technology for health, we have seen too many experiments and projects that have not been able to scale or sustain. How do you design and implement scalable and sustainable mHealth applications in low resource settings and emerging markets? This book chapter intends to provide lessons from case studies of two successful and large scale implementations of mHealth solutions and the choices that were made in the design and implementation of those solutions. The chapter uses Information Infrastructure Theory as a theoretical lens to discuss reasons why these projects have been able to successfully scale.

INTRODUCTION

India is the fastest growing mobile market in the world (ITU, 2010). Mobile phones are accessible in remote geographies and have become an integral part of the fabric of society. India has more number of mobile phones than landline phones. Thus, mobile phones are the most common medium of communication for long distances in India (TRAI, 2010). Such deep penetration in social structure and technological capabilities, make mobile phones relevant Information & Communication Technology for Development (ICT4D). Mobile technology has been identified as an important tool to strengthening of health information systems (Ganapathy & Ravindra, 2008). Provisioning of health services through the use of mobile technology is called mHealth. mHealth applications range from data collection for health services using mobile devices, delivery of health related information to medical practitioners or researchers or patients, monitoring of patient vital signs through mobile sensors or mobile networks and even direct interventions (telemedicine) through the use of mobile technology. Using Sweden as an example of developed economy, we see that mobile phone penetration in developing countries had reached the same penetration levels as that of Sweden within just 10 years (ITU, 2009); while for infant mortality, the rate in developing countries in 2007 was at the level where Sweden was 72 years earlier. This shows the irony between the progress made in mobile phone acceptance and health indicators.

The excitement around mHealth can be seen through the increased interest in mHealth applications as summarized by VW Consulting (2009). Recent studies (Pyramid Research, 2010) show that mHealth applications will "increase three-fold in the next two years by 2012". Thus, a whole network of actors which includes mobile operators, handset manufacturers, application developers, health providers, patients, researchers have large stake in the field of mHealth. With this increased interest in mHealth, we have also seen numerous attempts that have not been able to scale or meet the needs of the health sector. An analysis by Anderson & Perin (2009) of the VW Consulting (2009) report shows that only 7 out of 51 projects have been able to scale, while 36 out of 51 have been stuck in proposal or just small pilots that have not continued. Some of these pilots have stopped because the funding agencies stopped the project and were not taken up by the community or the government. Few others projects have been surpassed by better technology availability and highlights the fact that infrastructure in the field of mHealth changes quickly. The argument holds true for mHealth that any new field of research meets with initial failures. Experiments should be considered a commonplace in development of new science. But for how long can the investments continue without extending our pool of knowledge is a question that researchers and stakeholders ask of mHealth.

In next section of the chapter we present opportunities and challenges for mHealth, by looking at some failed examples of mHealth projects. The section presents mobile phones as the Information & Communication Technology (ICT) of choice compared to other technologies. In the later section, we present the case of scalable Indian mHealth solution, “SCDRT” and its technology choices. Afterwards, we present the case of using plain-text SMS in Kenya to scale health services. In Section 5, we look at our case studies through the lenses of information infrastructure theory and discuss the reasons why these two projects have been able to scale well. In the later section, we discuss open-source solutions and their advantages in emerging markets and ICT4D projects. In the last section, we provide avenues for future research and give concluding remarks. This book chapter attempts to bring together successful cases of mHealth applications and theoretically explain why these projects have been able to successfully scale.

OPPORTUNITIES & CHALLENGES FOR mHEALTH IN EMERGING MARKETS

There are numerous examples of mHealth failures, which in itself would be enough to fill this entire book chapter. Following is a concise list of solutions that were hailed as revolutionary, but have not been able to scale (Anderson & Perin, 2009):

- TeleDoc – Jiva Healthcare by The Soros Foundation and Jiva Institute in 15 villages in Haryana. Doctors receive diagnostics on Java-enabled mobile phones and medicines are delivered to the homes. Won award in 2003 World Summit at Geneva, but hasn’t scaled.
- Tamilnadu Health Watch – by Voxiva was after 2004 Tsunami. Disease reporting by mobile calls, fixed-line and Internet. 300 PHCs trained, not in use now.
- Handhelds for health – IIM Bangalore & Encore Software for disease tracking & surveillance. Started with use of “Simputer” and now uses “Mobilis”. These products are comparative to smart phones, but haven’t scaled to be used across villages or state.
- AIIMS telemedicine project makes use of Windows Mobile based PDAs for capturing patient data and create an EMR (Electronic Medical Records). This project has been in development and test at Vallabhgarh in Haryana for couple of years, but has not scaled beyond its use in the village.
- AESSIMS (Acute Encephalitis Syndrome Surveillance Information Management System) by Voxiva in Andhra Pradesh to report cases of encephalitis has been piloted, but has not been adopted for scale.

Although mHealth encompasses all kinds of mobile computing devices - from wireless chip-based solutions to portable computers, we believe mobile phones are the most scalable, especially when considering emerging markets and low-resource contexts. Other than mobile phones, even PDAs, laptops, specialized telemedicine equipment or video conferencing devices rely on mobile phone networks in rural contexts. Thus, the underlying wireless network of mobile phones is crucial to all kinds of mHealth solutions.



Figure 1: Range of mHealth devices on increasing cost

Another important consideration in terms of emerging markets and low-resource contexts is that there is a big problem in the capacity of the health staff. The number of doctors, nurses or health facilities available in comparison to the population is very low. This means that for a health beneficiary (anyone who receives medical services) to reach a location where medical services are delivered is problematic. Many times the health beneficiary has to travel long distances to avail proper primary care and mHealth provides an opportunity to bridge the distances between the patient and health providers or between

skilled medical practitioners and less skilled ones. In emergency situations, a low-skilled health worker can easily contact a medical officer and deal with the situation appropriately by using mobile phones. Where travelling distances for a skilled medical practitioner might be tedious as well as costly, mHealth may prove to be a useful solution that can be both efficient and low cost.

From a health information system perspective, mHealth provides a unique opportunity to get data quickly from the field-level health workers. Getting data on a timely basis is currently a big problem with health systems from emerging markets. From examples of mHealth initiatives (Mukherjee & Purkayastha, 2010), we see that mHealth solutions also help improve the data coverage rate because data comes from lowest levels of the health system. There is also a hypothesis presented in the above mentioned paper that since data comes from the lowest levels, quality and accuracy of data is better than if they came only from the higher levels. Although there is not enough evidence to support this hypothesis, it provides good research direction and opportunities for research in the field of health information systems for practitioners.

Braa & Purkayastha (2010) show that mobile phones have some important characteristics that make them suited to large-scale deployment and used in low resource settings.

- **Greater Market Penetration:** Mobile phones are easily available and people are less intimidated by the technology of mobile phones. This market penetration is also called the "installed base" in infrastructure theory. The installed base provides a form of inertia to the technology artifact and hence has greater chance of success in the system
- **Small Learning Curve:** There are only limited ways in which a mobile phone can work compared to a computer/Smartphone where it can perform many operations at a time. Mobile phones also only have a keypad for input instead of keyboard, mouse etc. available in computers. This makes the learning curve for using mobile phones much smaller.
- **Low Power Consumption:** Mobile phones consume less power compared to other kinds of mobile computing devices. This is extremely critical in low resource contexts where stable and continuous power is a rarity. Approximately 1.6 billion people worldwide live without access to electricity, of which 25% live in India.
- **Wide-coverage area:** Mobile phone networks work in remotest of places in these emerging markets. In emerging markets, the governments also have an inclusive agenda where they want people from all places to get access to mobile services.
- **Low cost of device:** Mobile phones ought to be between Rs.2000 (\$40) to Rs.5000 (\$100) to be purchased in mass quantities. This makes applications on mobile phones to have a wider audience compared to other mobile computing devices.

But along with these opportunities, there are also challenges for using mobile phones as the devices for mHealth applications. Mobile phones have some inherent limitations that need to be highlighted compared to computers or other mobile devices. These are embedded in the artifact of mobile phones, but with time could also possibly change. There is a clear trend that we see mobile devices are getting more and more powerful. But in today's context, the following limitations of mobile phones need to be taken into consideration when designing mHealth solution (Braa & Purkayastha, 2010):

- **Limited Processing Power:** The cost of the device is generally directly proportional to the processing power of the device. So, cheaper the phones are their processing capabilities also keep on decreasing. Most of the processors in mobile phones do not support multi-tasking and cannot allow processing large amounts of data.
- **Small Screen Sizes:** The normal display sizes for cheap mobile phones are less than 3 inches with resolutions less than 220dpi. Although small screen sizes help improve the battery life and make the devices low-powered, they are a limitation on the usability of the devices. We understand that most of the health workers in India are middle-aged women and hence it is extremely important to keep the font-size large, but still not waste too much of screen area.
- **Limited Visualization:** As people with low technological skills are the target audience, it would be great to display as much visualization as possible. Sadly, mobile phones are limited by the visualizations that they can show. Thus, the designer of mHealth application needs to

intelligently place graphics and icons on the screen, so as to make the applications attractive and more usable, but considering how clear and easy to understand the images can be.

- **Limited Memory:** Compared to computers or other devices, mobile phones cannot store a lot of data. This makes storing all data entered by the health worker for many months or years pretty much impossible. Even complete patient records for a large number of patients cannot be stored on the phone. Thus, mHealth application designers need to create applications that can sync records with online servers through mobile phone networks
- **Weak mobile networks:** At many of these contexts, the mobile phone networks are unstable or have weak signal strength. This means that mHealth solutions need to be robust enough to be able to deal with situations where no wireless communication is possible.

These pros and cons need to be carefully balanced by designers of mHealth applications and governments or implementing organizations need to realize the advantages of mobile phones as devices for mHealth.

In figure 2 below, the graph is an interpretive understanding from the author of the chapter that can assist new mHealth designers in balancing the pros and cons of mobile phones as a device in mHealth. The graph shows the infrastructure factors that a designer of mHealth application needs to consider. From left to right, we see increasing limitations to designing mHealth applications and from bottom to top we see the increasing weight of factors that need to be considered. For e.g. although market penetration is an advantage of huge importance, equally important limitation is the limited intuitiveness/usability of mobile phones. Similarly, although the low-cost of mobile phones is an advantage over smartphones, equally important disadvantage is that mobile phones have limited memory. This is not an exhaustive list of factors that need to be considered. But it is an indication of the factors that were considered by the designers of the cases mentioned later in this book chapter. Thus, all these factors mentioned in the graph need to be considered when designing mHealth applications.

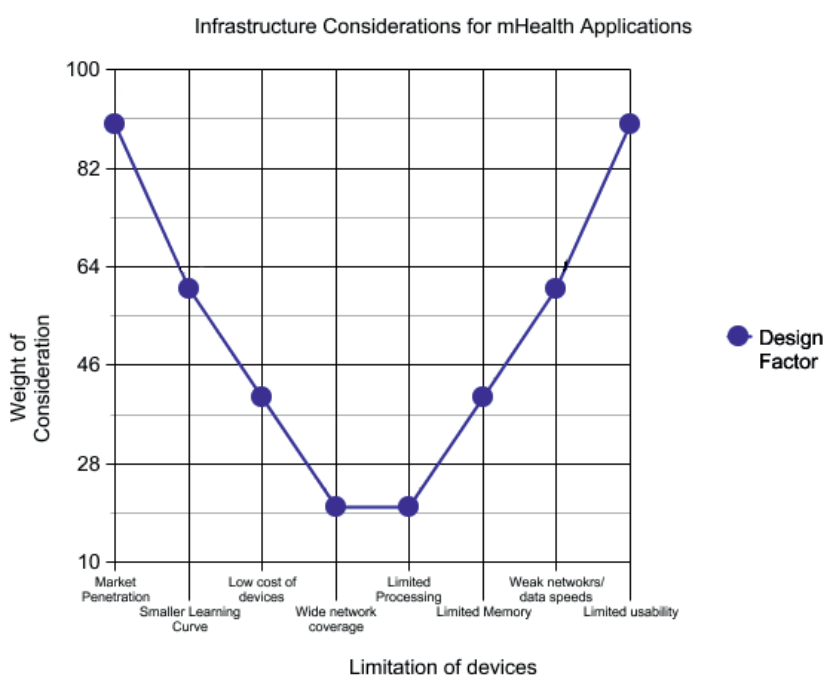


Figure 2: Infrastructure considerations for mHealth applications

To build a mobile health information infrastructure, a balance between the complexities of the context and technology needs to be analyzed. As mentioned by Hanseth and Ciborra (2007) simplicity of ICT

can help reduce risk. It is important that mHealth solutions be simple and manageable in their approach. As a first step to designing large-scale information systems, the chapter prescribes that simple technological solutions be tried first and then more complex solutions be developed and deployed as the designers as well as users become more and more used to the system. In the next section we discuss the importance of looking at mHealth applications as part of the infrastructure.

RESEARCH METHODOLOGY FOR CHAPTER

The *Indian mHealth case* described below in the chapter has been developed through Scandinavian action research tradition in IS development, such as user participation, evolutionary approaches and prototyping (Sandberg, 1985; Bjerknes et al., 1987; Greenbaum & Kyng, 1991). The research is part of a global network of action researchers and aims to generate knowledge by taking part in the full cycle of design, development, implementation, use and analysis. The above mentioned steps are done together with all the involved parties (government of India/state of Punjab, mobile phone operators, handset distributors and health workers) before the interventions are adjusted accordingly, and the next cycle begins again (Susman & Evered, 1978). The chapter is a result of the participation in this action research network for the last 3 years and provides insights from the central role in design of the system and participatory role in implementation and maintenance of the system after implementation. The research has been done within the framework of interpretive research (Walsham, 1995). Data was collected through group discussions, requirement meetings, developer discussions, feedback reports along with one-to-one interviews in health workers, health officers, doctors and ministers.

The first phase of the research for this project was started as part of the pilot and gathering requirements by involving the state health department and their officers through interviews and group discussions. After the requirements gathering phase, prototype demonstration were made to the national and state ministries of health working under the National Rural Health Mission (NRHM) programme. The feedbacks from these demos have been recorded in the meeting notes. These meeting notes have also helped in improving the software and served as further requirements for the next iterations. The next phases involved development of the software, which have been done using Agile Methodology of software development. Regular iterations and emails to the developer mailing lists have served as useful data to interpret the development of the software and project as a whole. The next phase of the research is the involvement in user training and after the user training, recording feedback on a set of questions from the health workers. The data collected has been analyzed and quantitatively represented in the report given to the ministries of each state, using which they have been able to participate in monitoring the progress and usefulness of the application. The research covers involvement in customization and training of the application and the data interpretation involves documents from implementations, health system manuals and being part of meetings. The research for Kenyan case has been done over a period of 6 months, by looking at the documents produced by the designers of the system and communicating with the implementation field workers for the project. The data interpretation for this case is through ex-post-facto observations of the implementation and documents made available by the implementers.

THE SCALABLE INDIAN MHEALTH SOLUTION

As governments seek to make health services more “patient centric”, there is increasing demand for the health service provider (called Auxiliary Nurse Midwife (ANM) in India) to be more mobile and be able to cover the catchment population in her jurisdiction (typically 5-7 villages or population of about 5000) for providing health services. Thus, the health worker's activities are mobile in nature and hence the tool required to assist her in service delivery should be mobile as well. mHealth solutions in this context are only going to be used by health workers when these solutions can provide support and assistance to the health workers in their day-to-day activities. If mHealth solutions do not do so, they are considered to be an additional burden for the health workers who are already covering a large population base and results in large scale protests and shutdown of the mHealth application.

Implementing software solutions at the lower levels of the Indian health system is a huge undertaking due to its enormous scale in terms of the vast number of installations, system maintenance and training activities. So when HISP India (a decade old NGO with global footprint) was approached in January

2009 by the National Health Systems Resource Center with an idea to empower health workers and get information from sub-centers, use of mobile phones became the obvious choice. The author of this book chapter being the Director of R&D of the project closely followed the project through design, development and implementation. The project was initiated as pilot in 5 states to represent varied geographies, varied user profiles, varied infrastructure and user base. The project was successfully evaluated a year later and has currently been implemented in full-scale in state of Punjab to over 5000 mobile phones and is under discussion for implementation in two other states. This is probably an example of one of the largest implementation of any mHealth solution around the world and reasons for that are simplicity, installed base and designing the solution by aligning to the available infrastructure.

The application in the pilot was called Sub Centre Data Registration and Transmission (SCDRT) application and had the following aims and outcomes:

1. To develop a very simple paper form-like data collection tool on mobile phones
2. Efficient transmission through SMS of the data from the sub-centre to the higher levels
3. Establish a basis for improved data quality and validation
4. Explore the potential of other value added mobile phone based applications, such as:
 - a. Providing feedback to the ANM on activity scheduling.
 - b. Strengthening processes of communication of the ANM with other functionaries (such as the medical doctor).
 - c. As a training tool, such as to help orient the ANM on new data elements.
5. Integrating the SCDRT information with the mainstream district based health information systems. In this case, it was the software application called District Health Information System (DHIS2 - a standard aggregate health information system used in many states in India) used in the pilot sites.

The details of the implementation of the pilot and its evaluation (Mukherjee & Purkayastha, 2010), highlights the different steps required for successful implementation of mHealth projects and also the challenges faced in the process of implementation. The paper highlights that there are enough challenges in the implementation of such a widely used mHealth solution and is a good example for implementers and policy makers. This book chapter on the other hand is geared towards mHealth application designers and what they need to consider when designing scalable mHealth solutions.

SCDRT enables the ANM to send monthly report via an SMS through the mobile phone to the next level(s). SCDRT was designed not as a stand-alone application for sending the report, but integrated with the mainstream district health information system for national level reporting. By design SCDRT was envisaged to be similar to the paper formats to maintain the familiarity and identification of the ANMs to the application. The figure below describes the SCDRT infrastructure and flow.

The flow is summarized:

1. ANM fills the Sub-Centre Monthly Dataset (monthly reporting form) on the mobile phones.
2. After completing the form, the report is sent as a text message (SMS) to desired phone numbers, which are located at the PHC/Block/District.
3. The SMS is compressed as a binary message. The compression is about 70%, resulting in more data being reported with lesser number of SMS messages.
4. These messages are received at the PHC/Block/District into the GSM Gateway Cards using a software application called SMSListener, and messages are imported into the state HMIS application (called DHIS2).
5. SMSListener is a utility application developed and configured to listen/receive SMS on a stand-alone system. When SMSListener receives the compressed message, it first decompresses it and writes complete information on xml files which then is easily imported into the DHIS2 application
6. The staff at PHC/Block/ District performs data quality checks using the built-in features called validation rules which are part of the DHIS2 application.

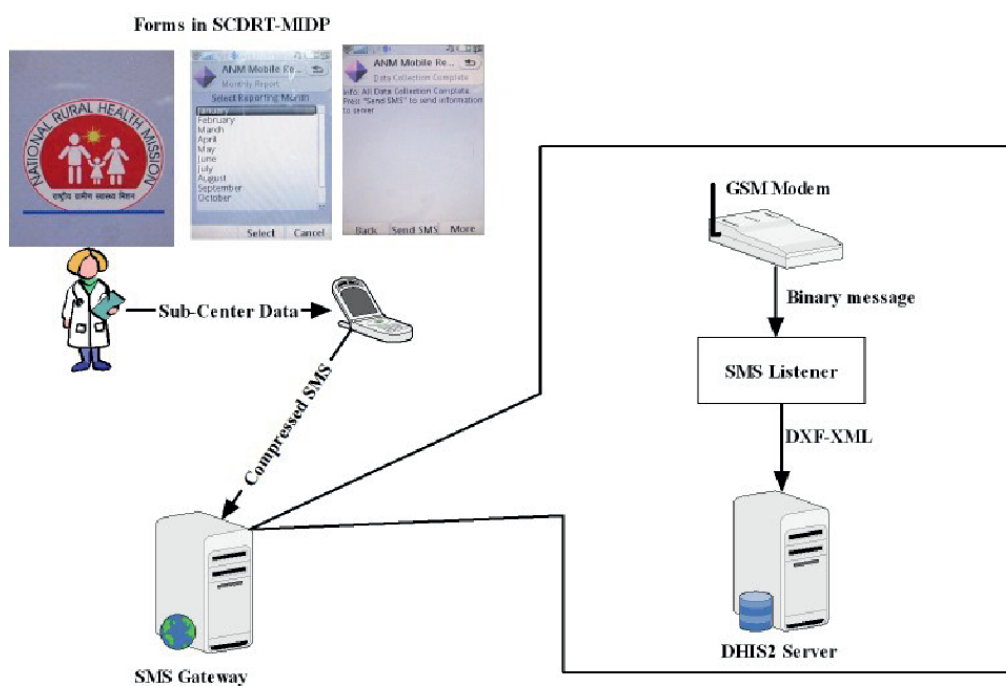


Figure 3: SCDRT Infrastructure & Flow

Managing Complexities in Design and Development

The SCDRT application and its evolution from the pilot can be described through the perspective of managing complexity in design, development and implementation. The SCDRT application uses very simple technology and depends in the background on the existing system of DHIS2 to store data on the server-end.

The application is developed using basic Mobile Information Device Profile (MIDP) 2.0 components that are part of all Java-enabled mobile phones and thus meant that the application could be run on basic Java phones. The specification required to run the application was the basic Java-enabled handset:

GENERAL	GSM 900 / 1800 / 1900
KEYPAD	5-way navigation key
DISPLAY	128 x 160 pixels (or more)
JAVA:	MIDP 2.0 (JSR 118)
	Connected, Limited Device Configuration (CLDC) 1.1 (JSR 139)
	Wireless Messaging API 2.0 (JSR 205)
	Scalable 2D Vector Graphics API (JSR 226) *
	FileConnection and PIM API (JSR 75) *

Table 5: Mobile phone specification

SCDRT Technology Requirements

1. Mobile Phones: Any Java-enabled phone; one for every ANM at the sub-centre.
2. GSM Modems: Each PHC/Block/District installation will receive SMS using these into DHIS2.
3. Software: Free & Open-source applications Mobile-SCDRT, SMSListener and DHIS2.
4. SIM cards: Every mobile phone and GSM modem requires a SIM card with a phone number.

Other than the handset at each health facility/health worker, there is not much infrastructure required to run the project. A GSM modem needed to be installed at the DHIS2 server. Any SIM card could be installed in the phones and GSM modem and they need not be the same. But to encourage communication and make the services cheaper, it is useful to establish a Closed-User Group (CUG) connection between all the mobile phones and GSM modems. The CUG connection from the operator makes it free to make calls within the health system and about 200 SMS are free in a month. It was seen that a CUG connection helps foster communication between health workers and also between the data managers and health officers. Besides the data coming and contributing to improving the health information system, the communication and improved efficiency at which the health system now communicates through mobile phone is a very important consequence. Most evaluations of this project suggests that health workers became more social with each other and the community in which they provide health services with the introduction of mobile phones. The CUG is thus a recommendation for any mHealth solution and is an important infrastructural consideration with the mobile phone operators.

The SCDRT-MIDP is a JavaME MIDP 2.0 application which uses plain JavaME components. It consists of Form, TextField, Choice, Command and built-in MIDP 2.0 layouts and CommandActions. A total of 9 screens are shown to the health worker based on the category of data reported by the sub-center. After the data collection is complete, a screen is displayed to notify the health worker of the fields that have not been filled. If acceptable, the "Send SMS" button is displayed and the data collected is sent as SMS to any (max 3) phone numbers, which are actually numbers of the SMS gateway. The data from the mobile devices is sent through SMS and the choice of SMS was to manage the complexity and align to the existing infrastructure in rural India where data speeds are slow and latency of the connections is huge. The SMS is created through a simple formatted string. Data from each TextField or Choice is collected and this is added as a pipe separated ("|") string. The month/week/date of reporting is separated from the rest of the string with a ("\$\$") dollar character. After the full string is created, it is sent to the Compressor class. The Compressor creates a byte[] array as output containing the data to be sent as an SMS. (sent as EMS binary message in 8-bit format).

The Compressor is a java class which compresses Java String to byte[] and decompresses a byte[] to a Java String. It uses range encoding for the compression and decompression. Range encoding is a simple algorithm for compression and does not require huge amounts of memory. This was an important technical decision when considering that the application was made for low-end mobile phones and should be usable for a large number of devices. The range encoding algorithm is also an open-standard and can be used by anyone easily. The compression ratios provided by range encoding are not superlative, but these are exactly the technical choices that the designer needs to make, when making solutions that can scalable and still be useful. The SMS sent is compressed through range encoding algorithm, which is simple yet elegant and gives an average compression rate of 67%. It does not include any salt and hence it would be incorrect to call it encryption. But when the data is sent, it is sent as a binary message and it cannot be understood if received on a normal phone, since it is binary data. On phones which send binary messages as only Nokia Picture Format (not as EMS), the decompression has some problems on the server side. On such phones, we have to disable compression before deploying. Most of these phones are outdated and not manufactured any more, but still it is one of the limitations of sending binary messages through the phones.

The SMS sent by the health worker is received by the GSM gateway, which according to the implementation model is located at 3 places (PHC, Block and District). The SMS Listener is kept running on the computer which acts as the GSM gateway. Whenever the SMS is received, the SMSListener first de-compresses the SMS and then converts it into an XML in a simplified DXF format. This XML is saved to a mi/pending folder in the DHIS2_HOME and is available to DHIS2 for importing. The "Mobile Importing" module in DHIS2 includes a start mobile importing button. When the process of importing to DHIS2 starts, it runs through the xml's located in the pending folder and imports the data in each of them. If the importing is successful the xml file is moved to the mi/completed folder. If some error occurs in the importing process, then the user is shown a message and the xml file is sent to the mi/bounced folder. The mobile importing module is an additional module that has to be included when building DHIS2 web application. It adds a "Mobile Importing" section to the Services

menu of DHIS2. Each mobile phone is registered to a single sub-center in the organization unit of DHIS2. The xml contains the phone number from which the SMS was received and when importing the data, the phone number acts as the information through which data is associated to a sub-center. Data after importing can be seen in the Data Entry screen within DHIS2. Thus, the mobile phone sends the same data that a person would do data-entry into DHIS2 through the keyboard.

On the mobile phone interface, all elements can be localized. In the pilot states, the application was localized to Gujarati, while in other states it was in English. In the state-wide implementation in Punjab, the application was localized to Punjabi using Gurmukhi text. The design of the application is simple and all localizations can be done through a single messages.properties text file. This text file is external to the application and can easily be modified by through a word processor without involving any change in code. This allows flexibility and ease to translate the application to multiple languages very easily. These localizations are lightweight and do not require a lot of processing power on the phones. The designers of SCDRT solution took into consideration that any localization required can be easily done by even non-developers and can be done with simple tools without requiring any programming knowledge.

The user-interface of the application deployed state-wide in Punjab is also extremely simplistic and easy for the user to understand. It has been designed to look similar to the paper forms and the pages are arranged similar to the paper forms.



Figure 4: Punjabi mHealth Application User Interface

The entire project including the DHIS2 server components have been developed through open-source collaboration. The projects have been publically hosted and contributions have been made by developers from at least 4 different countries.

At the start of development of the project, other alternative open-source software solutions were evaluated. JavaRosa is a popular open-source framework for developing mobile applications. JavaRosa is the Java implementation of the OpenRosa standard created by the OpenRosa consortium (Klungsoyr et al., 2008). JavaRosa renders Xforms on any mobile device that supports execution of Java on phones. Xforms are XML-based forms which separate the data being collected from the markup of the controls collecting the individual values (Boyer, 2007). The OpenRosa mobile consortium has defined some standard tags for Xforms on mobile devices, so that these can be commonly represented across devices.

Thus, JavaRosa was an implementation of the OpenRosa standard on Java-based mobile phones. The JavaRosa platform is very popular for mHealth projects and has been used by more than 15 different mHealth applications around the world (Klungsoyr et al., 2008). So ideally a solution based on JavaRosa would require the developer to create an Xform for whatever data one wants to collect on a server. The JavaRosa J2ME application would then send a request to the server and download the Xform to the mobile device. The user would then be able to see the form for data collection on the mobile phone and fill the data. After the data is filled in, a new Xform with data contents is submitted to the server and the server understand the data and extracts it out for processing. There were 3 major issues that cropped up when using JavaRosa:

- **Data Services/XML data:** One of the characteristics of low resource settings was the weak networks. These places have very slow connectivity for data services from mobile operators. Even the data services are not very robust in such settings. JavaRosa internally tried to solve this with fail through checks and resends, but that was an overhead for low resource settings. Additionally the cost of data services is quite expensive compared to SMS, especially when we are dealing with sending of monthly reports which consume only 1-2 SMS per report for 1 month.
- **Required a Powerful Phone:** Since the JavaRosa client was trying to read XML and render the forms on the fly, it required good amount of memory and processing capabilities on the mobile phone. This made the application display form elements slowly and would not work on the low-end phones in the earlier mentioned price range.
- **Lack of server-side integration:** OpenRosa Xforms were not linked or integrated with the DHIS in any way. This meant that additional development effort would have to be made on DHIS to be able to understand and parse Xforms sent through JavaRosa client. Thus, JavaRosa required much more efforts and polishing so that it can become an end-to-end solution for an already established health information system based on DHIS2.

Thus, to manage the complexities of GPRS, XForms, separate libraries for UI elements etc. a simplified application platform was chosen instead of JavaRosa. Another project which we looked at was Kiwanja's FrontlineSMS. It is a powerful community supported project which was designed for the purposes of low resource communities. It had proved to be successful in 2007 during the monitoring of the Nigerian elections (Banks, 2007). FrontlineSMS was a simple, yet powerful solution because it used Short Messaging Service (SMS) for sending information from mobile phones to the server. FrontlineSMS could be setup with only a GSM modem/phone connected to a computer and receive SMS. The SMS could then be put into a database or mapped to create an XML. Although FrontlineSMS seemed like an appropriate solution, the usability of the solution was in question. A similar approach to FrontlineSMS is used by the case presented below in Kenya, but the same solution was not considered appropriate for the Indian context. The Indian context wanted people to see the forms similar to what they would see in the paper forms and not type messages from the phone's SMS system. It was considered too tedious for middle aged health workers from India to be able to type messages correctly with the correct keywords and message formats

The software development for SCDRT did not take more than a couple of weeks and simplicity was the basic design goal of the project. After the software had been developed for the pilot, continuous feedback was received from the health workers through refresher trainings. From these training it was observed that the health workers needed better feedback and wanted to store more information on the mobile phones for their own use.

Thus, development was done in an iterative fashion with regular updates of the applications being installed on the phones. The advantage of the software being open-source was that other developers who were working on similar projects contributed to the applications and improved the usability and flexibility of the application over a period of time. There were projects in other countries in Africa that used the same framework, customized the forms and deployed the application to the health workers.

The regular updates to the application on the deployed handsets were done and immediate improvements in the user experience could be seen.

All these factors of managing risk and complexity ensured that the mHealth application could be scaled and implemented successfully to over 5000 phones and is continues to be scaled.

USING SMS IN KENYA FOR TO FIGHT MALNUTRITION AND MALARIA

The Millennium Villages Project (MVP) in Sauri, Kenya has worked on a pilot project to fight malnutrition and malaria in children under the age of 5 years through the use of SMS and the most basic mobile phones. This mHealth solution is called ChildCount and has been developed by the Earth Institute at Columbia University. Community Health Workers (CHW) are empowered with mobile phones and are trained to create SMS to report malnutrition cases. The project used a popular open-source platform known as RapidSMS as the back-end system and data is sent as formatted SMS from the phones of health workers.

Unlike the Indian mHealth application (SCDRT), this case makes the CHW to use the phone's SMS functionality and does not provide any form-like user interface. This makes the learning curve for creating the SMS longer, but highlights the issue of aligning solutions to infrastructures. The infrastructure of the health system in Kenya did not allow for buying new handsets and distributing it to health workers. Thus, basic mobile phones which did not have Java in them had to be made usable. Even the infrastructure for data services is unavailable and costly in Kenya. Thus, SMS was the most obvious infrastructural choice for such an mHealth solution.

The system provides the following functionality:

- Ability to the CHWs to register themselves by sending an SMS and identifying themselves with a phone number for sending reports.
- Ability of the CHW to register new cases of children with malnutrition and generate a unique ID number for each child.
- Ability to report different observations and measures of a child's case through SMS
- Automated notification and alerts to health workers for follow-ups and treatment of children whom they have registered
- A web interface through which data for individuals can be monitored as well as an aggregate view of indicators generated from individual records.

The following is an example of a formatted SMS that is sent to register a child into the system:

NEW LAST FIRST GENDER(M/F) Date Of Birth(DDMMYY) Parent
new sumi john m 010609 Joanna

After this SMS is received into the RapidSMS servers, there is a routine run to see if the child is already registered in the system. If the child is not registered in the system, a new patient record linking the child to the system is generated and sent back to the health worker as an SMS.

PATIENT REGISTERED> 121 SUMI, John. M/13M.
Mother: Joanna, Kangaba Village.

There also other features like **user-to-user messaging**:

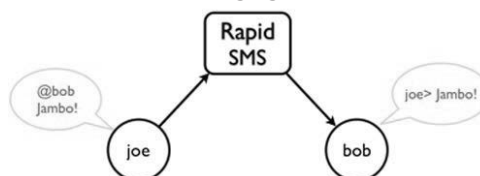


Figure 5: User-to-user messaging

For example:

@bob Jambo!

Would result in @yndour receiving:

joe> Jambo!

Group Messaging:

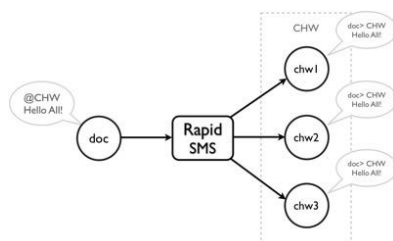
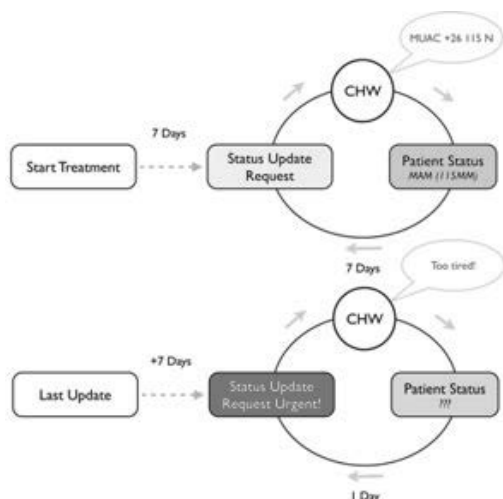


Figure 6: Group user messaging

Reminders and alerts:



Core to ChildCount is an automated reminder system to help ensure that no patient falls between the cracks. When certain events occur, like when a patient enters into a home-based Supplemental Feeding treatment program with Community based management of acute malnutrition (CMAM), a follow-up alert can be assigned to the caregiver assigned to that patient. When the time for a follow-up visit nears, in this case seven-days, a message will go out to the patient's health care provider requesting a follow-up malnutrition monitoring report. After one day, if ChildCount does not receive an update for that patient, the alert status for the patient will be elevated to urgent. At this point a reminder will be sent out on a daily basis not only to the health care to their manager and team mates for follow-up.

As experienced by the health staff and data managers, one of the unintended benefits of the system was the ability for the managers to provide better feedback to the health workers. The data coming in allows the health workers to work more optimally than before. Prior to ChildCount implementation, the health managers had a tough time to monitor the activities of the health workers. While the system is not encouraged to be used a monitoring system, there have been reports created to compare the performance of CHWs.

The project from the start looked at the proposed solution through an infrastructure perspective. The technology choices of using RapidSMS, formatted SMS, SMS as a delivery medium were all because of the existing infrastructure available. Also partnerships were established between different players of the ecosystem. Ericsson as a partner helped provide handsets of Sony Ericsson to each health worker, but to scale the system, existing handsets owned by health workers could also be used. Zain, the local GSM mobile operator provided a toll-free SMS number that allows CHWs to send messages to the system for free. The health facilities and the health officers in the pilot area were involved in training and consulting during the implementation of the project.

As seen in the report (Berg et. al., 2009), there have been over 20000 reports that have been sent by CHWs. There have been over 9500 child registrations and 7,646 nutrition screening reports, 839 Rapid Diagnostics Tests results and registrations of 7803 measles vaccinations

Along with the malnutrition and malaria, the solution is also being adapted for other programs such as immunization, PMTCT (Preventing Mother to Child Transmission of HIV). It is also being scaled at multiple MVP sites in different countries and also in a lot of countries in Africa. The fast scale at which the solution has grown and evolved highlights the importance of considering and aligning to the existing infrastructure when designing and developing mHealth solutions.

AN INFRASTRUCTURE PERSPECTIVE

The research approach is action-oriented and interpretative and characterized as a ‘network of action’ methodology. The network of action approach is based on the principle of creating learning and innovation through multiple sites of action and use, and sharing these experiences vertically and horizontally in the network. It is premised on collective action where connected research units are able to share experiences and learning. The case presented here is derived from units (or nodes) within the Health Information System Programme (HISP) network of action. HISP is an international research network doing open source development and implementation of District Health Information Systems (DHIS) in over 10 countries in Africa and Asia (Braa & Humberto, 2007).

Along with being simple and manageable, designers of mHealth solutions need to think of the infrastructure in emerging markets. As emphasized by Monteiro and Hanseth (1995), there exists the social shaping of infrastructure. We need to think of infrastructure as a socio-technical network and consider the technical aspects of design, but also the social capacities of individuals who are going to be part of the mHealth infrastructure. Mobile phones, unlike smartphones, PDAs, portable computers are already socially relevant. They signify empowerment of health workers who are majority women and the use of mobile phones as a communication tool is equally significant as much as a data collection and reporting tool. The ease of use of mobile phones and an existing installed base weighs a lot more than technological prowess of other ICT tools for the scalability and success of mobile phones.

It is important not to look at the mobile phone as a standalone device, but with a systems perspective including various other kinds of infrastructure – such as the paper registers at the sub-centers, the computers at the district levels, the mobile phone networks, internet access, the servers at the state level, and also the basic infrastructure required to support the mobile phone use (charging facilities, support centers from mobile operators and handset manufacturers, network coverage etc). To understand all of the factors of an mHealth infrastructure, the chapter suggests looking at Information Infrastructure theory as a theoretical lens.

Information Infrastructures have been identified by Hanseth & Monteiro (2001) to have the following characteristics:

1. Infrastructures have a supporting or *enabling* function.
2. An infrastructure is *shared* by a larger community (or collection of users and user groups).
3. Infrastructures are *open*.
4. Infrastructures are more than "pure" technology; they are rather *socio-technical networks*.
5. Infrastructures are connected and interrelated, constituting *ecologies of networks*.
6. Infrastructures develop through extending and improving the *installed base*.

From the above cases of mHealth applications, we look at the following aspects of Information Infrastructure:

1. **Enabling:** An mHealth infrastructure should be designed to support a wide range of activities, not especially tailored to one. It is enabling in the sense that it is a technology intended to open up a field of new activities, not just improving or automating something existing. Rather than just capturing data from health workers, mHealth applications should become part of the work culture of the health workers and assist them in their day-to-day activities. In the above cases, we see this as one of the establishing principle. Although both the applications were designed for data collection and reporting, the infrastructure created due to the mHealth applications enabled other forms of use. These include peer-to-peer communication, communication between community and the health workers, requests for leave and information exchange about

the new developments in the community. Health workers in the Indian case were given mobile phones which had a digital camera in them. This made it possible for health workers to capture images of mosquito pits or skin rash in patients, so that further action can be taken on these. When these images were shown during monthly meetings at the district-level, officers responded in a better way and health workers felt more empowered. Thus, the mHealth infrastructure created by these applications acted as enablers for new ways of working in the health system. This enabler of change may sometimes be expected by the designers of the mHealth applications or sometimes there may be unintended consequences.

2. **Shared:** The mHealth infrastructure should be able to bring in more partners and users. Mobile phones assist in communication between the medical officers, state health & data officers and the field-level health workers. The infrastructure allows shared information between members of the health structure. In the above cases, we see that the information received through mobile phones is shared between different health staff and officers. In the Kenyan case we see that different programs have been integrated with the advent of this mHealth application. Since the same health worker provides data for different health programs like HIV/AIDS, PMTCT, Malnutrition and Immunization, it makes sense that the same application can report for all this programs. Thus, it avoids duplication of work for the health workers and increases the efficiency and interest of health workers.
3. **Open:** The mHealth infrastructure should be made open in the sense that there are no limits for number of user, stakeholders, vendors involved, nodes in the network and other technological components, application areas or network operators. There are already different standards that allow open-access to technology in mobile networks. GSM, CDMA, SMS are standards that allow communication between different mobile handsets. There are also standards like XForms that allow standard communication of data and data collection forms on multiple devices. When you look at the Kenyan case, use of plain-text SMS allows different kinds of keywords that can be created by the administrators of the system and change the way in which they report data. This openness in the infrastructure allowed the system to be spread to other programs and health workers. The Kenyan mHealth application also allows new health workers to register themselves and start reporting data. This allows inclusion of many more people into the system and training for the usage of the application is done by one health worker to the other through the word of mouth. In the Indian case, we see that the back-end system of DHIS2 allows creating many flexible datasets and this can be used with mobile phones for reporting. In both the cases, we see that the solutions are open for use with any mobile operator, any mobile handset and open standards.
4. **Socio-Technical Networks:** mHealth is not just a technology piece in the health system, but rather involves the socio-technical network. Although ICT and mobile phones are part of the infrastructure, they are to be symmetrically seen with the people and processes involved in the system. Infrastructures are heterogeneous concerning the qualities of their constituencies. They encompass technological components, humans, organizations, and institutions. We see from our cases that the designers of the mHealth solutions have given importance to how their users are going to use the system and made quick changes based on the user's feedback. The human aspects of the system like user capacity, health system hierarchy, processes of reporting were considered in design of the systems. The designers of the system did not look at the mHealth solutions as a technology tool in standalone, but as an assemblage of the health worker and the mobile phone + application. Also the assemblage is between this entity and different elements of the health system that needs to be considered by the designers of the mHealth application. The use of CUG in the above cases is an example of how social networks were tapped and better use of the infrastructure was done to foster communication and the humanness in all of this.
5. **Ecologies:** Infrastructures are layered upon each other just as software components are layered upon each other in all kinds of information systems. This is an important aspect of

infrastructures, but one that is easily grasped as it is so well known. In mHealth, there is an ecological dependence between mobile operators, handset manufacturers, health workers, health information systems. All these players act together to make a mHealth system work and finally result in better health services delivery. From the Indian case, we see that the implementers first negotiated with the handset manufacturers and mobile phone operators for best deals for the solution. The low price and Java-enabled compatibility of the handsets were important components of the ecology that enabled the solution to scale well. The mobile operators providing CUG connection at low prices was also important factor of the ecology. The mHealth solution in both the cases provided a win-win situation for all the parties in the mobile ecosystem.

6. **Installed base:** There is already an installed base of mobile phones, mobile networks, health data capture forms, health information systems. There is inertia within an infrastructure and designers of mHealth application should realize this inertia. For e.g. Some areas have better mobile networks from certain mobile operator. People from some areas are better used to certain languages and certain handsets. This inertia of the installed base is hard to change and mHealth applications need to align themselves to the installed base. In the Indian case, we see that the DHIS2 system was an already existing system which is widely by the different states for management of the health system. It was important for the mHealth application to make use of this installed base and is one of the important factors for its success. Similar installed base of a large number of Java-enabled mobile phones being available in the market improves the success rate of this solution.

USE OF OPEN-SOURCE COLLABORATION

In both the cases we see a common use of open-source software and collaboration between different developers and players. We consider this to be an important reason for success of these mHealth applications. Open-source software allowed the designer and developers of the solution to quickly improve the software through the help of other programmers who had the knowledge of similar projects. The same mHealth applications could be easily adapted to other contexts due to the open-source nature of the solutions. These mHealth applications are also being implemented at new locations and customized according to those contexts much more quickly due to the open-source nature of these applications.

Raymond (2001) discusses that open-source became a phenomenon through an accidental revolution. But when designing mHealth applications, we need to consider open-source not by accident, but by looking at the pros and cons that open-source software development brings to the table. Today we see a billion dollar industry for free and open-source software with headline products such as Linux and Apache (Feller & Fitzgerald, 2002), the Free & Open-Source Software (FOSS) community and business of FOSS in health care or mHealth is yet to emerge. The same rules of success or failures of FOSS that have been researched in technology artifacts like operating system and web servers can be applied to mHealth applications is still an open question. We still need more evidence to see if mHealth applications can use the open-source model and be successful and prove to be as wide-spread as the other mentioned technology artifacts. But one underlying concept of "Freedom" is something that FOSS in mHealth provides is beyond doubt. Let us discuss what freedoms we observe in our cases of mHealth solutions.

Freedom of pricing - The cases of mHealth applications that are presented in this chapter were created by a number of developers through open-source collaboration. Although most of the developers are involved in other activities, their contributions have enabled in creation of these applications. The overall cost of development of the mHealth applications is low because these were primarily contributions of a group of global developers. In the case of the Indian mHealth solution, the collaboration and the number of developers participating initially were less, but later after the pilot finished and lessons were learnt, a larger group of developers found interest and took up the development to forefront. Due to the open-source development model, any new implementation of this

mHealth solution can directly use the application "as is" or make changes to the application freely. This makes the cost of entry and development for the new implementation very less compared to if they would have to start developing their own solution from scratch. It is also evident that after the pilot, the state of Punjab did not have to pay anything in development cost for the mHealth solution. The software was quickly translated by the state's own developers within a day and the dataset for reporting was also easily customized by looking at the existing source code of the pilot applications. Similarly the ChildCount application from Kenya is being used at many other sites. Some of these are independent of the support of MVP or the original developers of the software. Other developers and implementers are able to pick the software and start using and customizing it without any paying any cost to the original software developers. The original developers are not getting paid and that could be seen as their loss, but since they created the software initially for themselves only and now it is being used at other places, proves to be beneficial for the community of mHealth initiatives on the whole. Also the original developers of the tool have the opportunity of being requested by other implementers for customization and support, which can be a useful business model, as seen in other FOSS software.

Freedom of choice - Along with freedom of pricing and cost, the health system has the freedom to choose which organization to use for supporting the development or maintenance of the mHealth solution. The state where the solution is implemented also gets the freedom to modify or change the solution without having to pay or ask for permission to the original authors of the system. Sen (1999) refers to such freedom of choice as a social opportunity where markets provide people with liberty, helps foster efficiency and build democratic institutions. With open-source mHealth applications, there can be democratic processes for development of solutions. It also gives local development opportunity of international and global solutions for customization to local contexts. This freedom of choice is important for mHealth applications because that allows for continuous improvements and change. It also allows software to be flexible and change according to the changes in the health system. This makes the ICT tool to be more in sync with the actual processes of the health system.

Freedom of interoperability - Integration has been considered as the holy-grail of Management Information Systems (Kumar and van Hillegersberg, 2000). People would like to see data from different sources for comparison and management of the available resources. This is more important for the health system in developing countries where resources are limited and these have to be effectively managed. Thus, when we talk about integrated view of data for management, it is important to have inter-operability between different software systems that are used in the health system. Open-source software as a principle allows better interoperability. Since the different algorithms for storing data, formats for data exchange and ways to process and manipulate them is open and accessible, these systems are easy to interoperate with. The cases of mHealth applications in this book chapter allow interoperability through very simple technology. The Indian case of SCDRT, allows any application that can send data through compressed SMS can communicate with the DHIS2 server. The compression algorithm is open and so is the XML format for data exchange. Any other mobile application can be used with the server end of the system and use to send data to DHIS2. Similarly, the Kenyan case uses HL7 as a standard for data exchange after data has been received by the server. The different keywords that are setup in the server side for reporting of data can be exported to other systems. The underlying platform of RapidSMS has a open-format for configuring SMS structure and this makes it easy for other people to build upon. Anyone who wants to create another application can interface with the RapidSMS through its API and easily create new solutions similar to ChildCount.

Freedom to collaborate - The community around these mHealth solutions encourages the process of sharing knowledge. At many times during implementer or developer debates, the teams would put the question to the community at large and the community members who faced similar challenges would respond back. Similarly, other organizations with similar requirements for such solutions share use-cases which helps build a more generic application. Since the community is already sharing code, it also encourages people to share stories of success, failures, challenges and politics. Thus, other than just code being shared, open-source enables sharing of ideas and helps in the times of challenges.

As with proprietary and closed system, exchanging data with these systems or changing them involves many unfreedoms. The unfreedoms are inherently part of the business model of closed-source or proprietary software. Some of these are not purely technical choices, but also organizational or business choices.

- Intellectual Property (IP) challenges related to proprietary ownership - When a designer or a developer of an mHealth solution does not want their software to be changed by others or wants to share the revenue of modification open-source isn't a good choice. Proprietary software is more useful when we want to establish an Intellectual Property on the mHealth solution and want to give this technology to someone at a price tag.
- Restricted access to way in which information is stored and managed. Restrict the flexibility to exchange information - Proprietary software generally provides these restrictions. These restrictions hamper the inter-operability of the software with other systems. Software developers would be able to charge extra fees for modifying the software for inter-operability with other products.
- Restrictions to modification of existing hardware/software platforms - Some people do not want their solutions to be modified or there can be a business logic to charge for such modifications. Either ways, open-source software is not good if you want to imply such restrictions. Proprietary software creates unfreedoms such as these and implies restrictions on hardware/software changes to mHealth solutions.
- Cost of initial ownership as well as recurring cost of modifications to the system - as mentioned earlier and in the example of our cases, due to open-source software the cost of ownership or entry for a new implementer is very low. Also people can modify and customize to their free will. This is not possible with proprietary software or people have to pay more or get special access to make such customizations.
- Non-compliance to standards of data exchange - Although you can have proprietary software systems to be complaint with standards, it is much harder in proprietary systems to understand the way these store and manipulate data. Thus it becomes much harder for people to inter-operate with closed-source / proprietary software systems.

With Free and Open-Source Software (FOSS) there is choice to be able to replace these unfreedoms. Sen (1999) concludes through his observations and reasoning that increasing freedom to individuals or a social group leads to development. Not only that, but removing unfreedoms is an important criteria to development. With FOSS initiatives we suggest that stakeholders get freedoms and this in turn would lead to better chances in developmental projects. Thus if mHealth initiatives are being implemented with a social development focus, then there is a strong suggestion to use free and open-source software.

An important prescription of this chapter is to show how can mobile applications be sensitively designed and introduced in order to support the development of an integrated mobile based health information infrastructure. Information infrastructure is used in the broader sense, meaning the technological and human components, networks, systems, and processes that contribute to the functioning of the health information system. Strengthening and reinforcing the social network of health workers by creating communication possibilities across hierarchy and among peers, as well as creating feedback channels for the lowest level will be crucial. Existing research into the topic of how sustainable mobile health information can be effectively deployed and scaled is limited, and hence this topic lies in the frontiers of health information systems research (Donner, 2008).

CONCLUSION & FUTURE RESEARCH DIRECTIONS

In the chapter, it is evident that considering the available infrastructure is critical to the success of implementation of mHealth solutions. Through the cases of two large-scale, successful mHealth applications, it can be seen that from the planning and design phases of mHealth projects the socio-technical networks needs to be studied as the infrastructure. The inertia of existing work practices and

systems need to be dealt with and integration with existing systems is crucial for scaling mHealth solutions. We also see that minimizing risks and managing complexity correctly is another step in building scalable mHealth solutions.

Today with the mounting interest in mobile applications in health (mHealth) in emerging markets it is an interesting field of research for a number of reasons:

1. **Evolving Infrastructures:** Mobile phones represent the fastest growing ICT in emerging markets. Mobile phones reach lower-levels of the system and show promise to bridge the digital divide (Warschauer, 2004). As time goes by, we see more and more computing power being built into mobile devices and the prices of mobile devices coming down greatly. Smartphones with more powerful operating systems and those that enable hosting more powerful applications are more commonplace and these devices become cheaper and cheaper (Raento et al., 2009). Most emerging markets are also deploying 3G services that enable high-speed data and internet on mobile phones. Thus, with time data over IP packets would make more sense as data transmission medium instead of SMS. People also will gain more skills over time with the use of mobile phones. Thus, mHealth applications should be adapted to this improved user skills.

As the infrastructure improves, how these scaled and established base of mHealth applications evolve will be an interesting challenge for the implementers of these solutions. There is also a need for research to be able to highlight a typology of applications that suit a given infrastructure. Given that the mHealth applications are customized and are being used in different contexts, it is interesting to study the evolution and changes in the solutions and how they are being used in the different contexts. How technology from one place can be directly adapted for use in another context is a challenge and has potential for further research on these mHealth solutions.

2. **Changing Mobile Standards:** Mobile standards are also changing rapidly. Recent rulings on mobile net neutrality (FCC, 2010) show that mobile telephony could slowly be moving towards more operator driven access to internet from an open-access internet. Mobile computing represents the future of access to the internet, but a change in the open nature of this infrastructure could result in a different evolution of the mobile world. The change of power to the mobile operators could result in different network configurations. There is a continuous change in the power struggle between mobile operators, handset manufacturers and government regulators. As can be seen in the 3G auctions in India, suddenly the cost of operations in India for mobile operators have increased to 10-folds. If the same pricing for mobile services are available after this change is an interesting trend to study. Any change in these factors need to be considered by developers of mHealth applications. Also with 3G, the opportunity to do a lot of data transmission increases and the solutions that can use these fast speed of data exchange will be interesting to research.
3. **Quality Studies:** As shown by Akter et. al. (2010), there is a need to study the quality of mHealth applications and how quality affects the growth of the field of mHealth. Along with perceptions of the people who use these systems, it is also important to see the improvements in the health service delivery after the continuous use of mHealth applications. Call for quality studies should result in research in fields of CSCW (Computer Supported Co-operative Work), HCI (Human Computer Interaction) and Anthropology to study the behavior of health workers with mobiles. There are some very limited conclusions that have been made by studies on quality of the mHealth applications by independent researchers and organizations. What methods to use, what kind of studies will be required for evaluating the quality of mHealth applications is an interesting and emerging area of research.

REFERENCES

- Akter, S., D'Ambra, J., & Ray, P. (2011). Trustworthiness in mHealth information services: An assessment of a hierarchical model with mediating and moderating effects using

- partial least squares (PLS). *Journal of the American Society for Information Science and Technology*.
- Anderson & Perin (2009). *Case studies from the Vital Wave mHealth Report*, Retrieved November 10, 2010 from http://www.cs.washington.edu/homes/anderson/docs/2009/mHealthAnalysis_v1.pdf
- Banks, K. (2007). then came the nigerian elections: The story of frontlinesms. *SAUTI: The Stanford Journal of African Studies*, (Spring/Fall), 1–4.
- Berg, M., Wariero, J., Modi, V. (2009). *Every Child Counts - The use of SMS in Kenya to support community based management of acute malnutrition and malaria in children under five*, Retrieved December 25th, 2010 from http://www.childcount.org/reports/ChildCount_Kenya_InitialReport.pdf
- Bjerknes, G., Ehn, P., Kyng, M., & Nygaard, K. (1987). *Computers and democracy: A Scandinavian challenge*. Gower Pub Co.
- Boyer J (2007) XForms 1.0 (Third Edition). W3C. Retrieved December 25th, 2010 from: <http://www.w3.org/TR/xforms/>
- Braa, K., & Purkayastha, S. (2010). Sustainable mobile information infrastructures in low resource settings. *Studies in health technology and informatics*, 157, 127.
- Braa, J., & Humberto, M. (2007). Building collaborative networks in Africa on health information systems and open source software development–Experiences from the HISP/BEANISH network. *BEANISH network*.
- Donner, J. (2008). Research approaches to mobile use in the developing world: A review of the literature. *The Information Society*, 24(3), 140–159.
- Federal Communications Commission (FCC) (2010). *Report and Order on Open Internet Rules* (Federal Communications Commission, December 2010, Washington D.C) Retrieved from http://www.fcc.gov/Daily_Releases/Daily_Business/2010/db1223/FCC-10-201A1.pdf
- Feller, J., & Fitzgerald, B. (2002). *Understanding open source software development*. Addison-Wesley Longman Publishing Co., Inc. Boston, MA, USA.
- Ganapathy, K., & Ravindra, A. (2008). *mHealth: A potential tool for health care delivery in India*. Rockefeller foundation.
- Greenbaum, J. M., & Kyng, M. (1991). *Design at work: Cooperative design of computer systems*. CRC.
- Hanseth, O., & Ciborra, C. (2007). *Risk, complexity and ICT*. Edward Elgar Publishing.
- Hanseth, O., & Monteiro, E. (2001). Understanding information infrastructure. *Manuscript available online: http://www.ifi.uio.no/*oleha/Publications/bok.html*, Download in December.

- ITU (International Telecommunication Union) (2009). *World Telecommunication/ICT Indicators Database 2009*(Geneva, Switzerland: International Telecommunication Union (ITU), www.itu.int/ITU-D/ict, 13th edn).
- ITU (International Telecommunication Union) (2010). *World Telecommunication/ICT Indicators Database 2010*(Geneva, Switzerland: International Telecommunication Union (ITU), www.itu.int/ITU-D/ict, 14th edn).
- Klungsoyr, J., Wakholi, P., Macleod, B., Escudero-Pascual, A., & Lesh, N. (2008). Open-ROSA, JavaROSA, GloballyMobile—collaborations around open standards for mobile applications. *International Conference on M4D Mobile Communication Technology for Development, Karlstad University, Sweden*.
- Kumar, K and van Hillegersberg, J. (2000), “ERP Experiences and Evolution”. *Communications of the ACM*, 2000, 43(4), pp. 22-26.
- Monteiro, E., & Hanseth, O. (1995). Social shaping of information infrastructure: on being specific about the technology. *Information technology and changes in organizational work. Proceedings of the IFIP WG8. 2 working conference on information technology and changes in organizational work, December 1995* (p. 325–343).
- Mukherjee, A., & Purkayastha, S. (2010). Exploring the potential and challenges of using mobile based technology in strengthening health information systems: Experiences from a pilot study. *AMCIS 2010 Proceedings*, 263.
- Pyramid Research (2010). *Health Check: Key Players in Mobile Healthcare* (Pyramid Research, <http://www.pyramidresearch.com/store/RPMHEALTH.htm>)
- Raento, M., Oulasvirta, A., & Eagle, N. (2009). Smartphones. *Sociological methods & research*, 37(3), 426.
- Raymond, E. S. (2001). *The cathedral and the bazaar: musings on Linux and open source by an accidental revolutionary*. O'Reilly & Associates, Inc. Sebastopol, CA, USA.
- Sandberg, A. (1985). Socio-technical design, trade union strategies and action research. *Research Methods in Information Systems, North-Holland, Amsterdam*, 79–92.
- Sen, A. (1999). *Development as freedom*. Oxford University Press.
- Susman, G. I., & Evered, R. D. (1978). An assessment of the scientific merits of action research. *Administrative science quarterly*, 23(4), 582–603.
- TRAI (Telecom Regulatory Authority of India) (2010). *Annual Telecom Report* (Ministry of Telecom; India).
- VW Consulting (2009). mHealth for development: the opportunity of mobile technology for healthcare in the developing world. *Washington, DC and Berkshire, UK: UN Foundation-Vodafone Foundation Partnership*.
- Walsham, G. (1995). Interpretive case studies in IS research: nature and method. *European Journal of information systems*, 4(2), 74–81.

Warschauer, M. (2004). *Technology and social inclusion: Rethinking the digital divide*. the MIT Press.

ADDITIONAL READING SECTION

Asangansi, I., & Braa, K. (2010). The emergence of mobile-supported national health information systems in developing countries. *Studies in health technology and informatics*, 160, 540.

Braa, J., Hanseth, O., Heywood, A., Mohammed, W., & Shaw, V. (2007). Developing health information systems in developing countries: the flexible standards strategy. *MIS Quarterly*, 31(2), 381–402.

Braa, J., Kanter, A. S., Lesh, N., Crichton, R., Jolliffe, B., Sæbø, J., Kossi, E., et al. (2010). Comprehensive Yet Scalable Health Information Systems for Low Resource Settings: A Collaborative Effort in Sierra Leone, 2010, 372-376.

Braa, J., Monteiro, E., & Sahay, S. (2004). Networks of Action: Sustainable Health Information Systems across Developing Countries. *MIS Quarterly*, 28(3), 337-362.

Bults, R., Wac, K., Van Halteren, A., Nicola, V., & Konstantas, D. (2005). Goodput Analysis of 3G wireless networks supporting m-health services. *8th International Conference on Telecommunications (Con TEL05)*.

Ciborra, C. U., & Hanseth, O. (1998). From tool to: Agendas for managing the information infrastructure. *Information Technology & People*, 11(4), 305–327.

Dick, M. H. (2010). Weaving the “Mobile Web” in the Context of ICT4D: A Preliminary Exploration of the State of the Art. *Proceedings of the American Society for Information Science and Technology*, 47(1), 1–7.

Donner, J. (2004). Innovations in mobile-based public health information systems in the developing world: An example from Rwanda. *Retrieved November, 18, 2008*.

Donner, J., Verclas, K., & Toyama, K. (2008). Reflections on MobileActive08 and the M4D Landscape. *Proceedings of the First International Conference on M4D* (p. 73–83).

Fraser, H. S. F., & Blaya, J. (2010). Implementing medical information systems in developing countries, what works and what doesn't. *AMIA Annual Symposium Proceedings* (Vol. 2010, p. 232).

Hanseth, O., & Aanestad, M. (2003). Bootstrapping networks, communities and infrastructures. On the evolution of ICT solutions in health care. *Methods of Information in Medicine*, 42(4), 385–391.

Hanseth, O., & Monteiro, E. (1997). Inscribing behaviour in information infrastructure standards. *Accounting Management and Information Technologies*, 7, 183–212.

Hanseth, O., Monteiro, E., & Hatling, M. (1996). Developing information infrastructure: The tension between standardization and flexibility. *Science, technology & human values*, 21(4), 407.

- Haux, R. (2006). Health information systems-past, present, future. *International Journal of Medical Informatics*, 75(3-4), 268–281.
- Heeks, R. (2002). Information systems and developing countries: Failure, success, and local improvisations. *The Information Society*, 18(2), 101–112.
- Hoe, N. S. (2006). *Breaking barriers: the potential of free and open source software for sustainable human development; a compilation of case studies from across the world*. Elsevier, New Delhi, IN.
- Istepanian, R. S. H., & Pattichis, C. S. (2006). *M-health: Emerging mobile health systems*. Springer-Verlag New York Inc.
- Istepanian, R. S. H., Jovanov, E., & Zhang, Y. T. (2004). Guest editorial introduction to the special section on m-health: Beyond seamless mobility and global wireless health-care connectivity. *Information Technology in Biomedicine, IEEE Transactions on*, 8(4), 405–414.
- Ling, R. S. (2004). *The mobile connection: The cell phone's impact on society*. Morgan Kaufmann Pub.
- Mansell, R. (2002). From digital divides to digital entitlements in knowledge societies. *Current Sociology*, 50(3), 407.
- Mishra, S., & Singh, I. P. (2008). mHealth: A developing country perspective. *Making the eHealth connection. Bellagio, Italy*.
- Monteiro, E. (2000). Actor-network theory and information infrastructure. *From Control to Drift*, 71–83.
- Sahay, S., & Walsham, G. (2006). Scaling of health information systems in India: Challenges and approaches. *Information Technology for Development*, 12(3), 185–200.
- Sarker, S., & Wells, J. D. (2003). Understanding mobile handheld device use and adoption. *Communications of the ACM*, 46(12), 35–40.
- Tongia, R., & Subrahmanian, E. (2007). Information and Communications Technology for Development (ICT4D)-A Design Challenge? *Information and Communication Technologies and Development, 2006. ICTD'06. International Conference on* (p. 243–255).

KEYWORDS & DEFINITIONS

Health Information Systems: are information systems that capture data related to health of an individual, community or an entire nation and allow meaningful use as information.

mHealth (or m-Health): Provisioning of health services through the use of mobile technology is called mHealth. mHealth has also been referred to as Mobile Health Information Systems.

Information Infrastructure: A structure of the people, processes, procedures, tools, facilities, and technology which supports the creation, use, transport, storage, and destruction of information.

Free & Open-source Software: is software that is liberally licensed to grant the right of users to use, study, change, and improve its design through the availability of its source code

SMS: Short Messaging System is the text communication standard in phone or mobile systems allowing non-voice or non-data communication between different users of telephone networks.

Scalability: is the ability of a system, network, or process, to handle growing amounts of work in a graceful manner or its ability to be enlarged to accommodate that growth

Complexity: It is the measure of the number of linkages to properties in an object. This collection of properties is known also known as state.

Auxilliary Nurse Midwife (ANM): is the peripheral health worker who provides care to childbearing women during pregnancy, labour and birth, and during the postpartum period. They also care for the newborn and assist the mother with breastfeeding as well as provide family planning knowledge and counselling.

P2: A post-development perspective on mHealth – An implementation initiative in Malawi

Abstract

While the number of mHealth projects is on the rise, global health administration has tried to put the focus of evaluation of these projects by the measure of health outcomes. Through our implementation in Malawi we argue that only measuring improvements in health interventions is a top-down view about the developmental impact of mHealth. We critique that looking at improvements in health indicators or scaling in terms of breadth (number of users) or scaling in terms of depth (use across organizational hierarchy) is incomplete view of the developmental impacts of mHealth. Through our action-research project of facility-based reporting of health data through mobile phones, we conclude that developmental impacts of mHealth are local and each locale has a different developmental impact depending on the context of use and available resources. Through the discussion that is part of the paper, we create a comparison of priorities between global development and local priorities with respect to mHealth projects (and probably eHealth projects).

1. Introduction

mHealth or Mobile-based Health Information System projects have shown the promise of change in health care in developing countries [1]. Although the number of mHealth initiatives is staggering, recent reviews have shown that most initiatives have failed to scale beyond pilots [2]. Other researches [3] have suggested that to see the benefits of “ICT for development” (ICT4D) interventions in primary healthcare, these need to be scaled to a level, where they can inform decision making and resource allocation for whole administrative regions. Even at the global level or country-level, there is a push towards suggesting that the real impacts of mHealth initiatives can be experienced when these solutions can scale in breadth (number of users) or scale in terms of depth (across organizational hierarchy). There is also a large number of mHealth initiatives that consider developmental impact of mHealth projects primarily through impacting health care interventions [4][5][6] or indicators [7]. Recent review of mHealth projects shows that while mHealth initiatives have focused on treatment compliance, data collection & disease surveillance, point-of-care support for health workers, disease prevention, health promotion, emergency response [8], very few projects have tried to study the organizational impact of mHealth. We argue with a post-development perspective [9] that developmental impacts of mHealth are first to be analyzed through the local organizational changes and later through impact at the macro-levels of administrative regions or impact on indicators. We also question the idea of how power and visibility [10] for local change might be taken away from the hands of community health workers (or patients), who are the direct users of mHealth applications and moved over to managers and administrators at the higher levels within the health system. Sometimes this power and visibility focus is on foreign agencies that are funding or implementing mHealth solutions.

Rest of the paper is organized as follows. In section 2, we put forth the concepts from post-development theory that guides our analysis. In section 3, we describe our research approach of Critical Action Research. In section 4, we describe our empirical case of mHealth initiative and detail the development and implementation of the mHealth project. In section 5, we provide analysis and discuss the developmental impacts of mHealth projects at grassroots level. We conclude the paper by providing a comparison of priorities between global development and local priorities with respect to mHealth projects (and probably eHealth projects) that are designed, developed and implemented just from a top-down world-view, but rather put the locale at the fore-front of all mHealth implementations.

2. Post-development theory

Post-development theory puts forth the idea that the notion of development has led to a mental structure that has resulted in a hierarchy of developed countries and under-developed countries [11]. Such hierarchy leads to the idea that under-developed countries depend on help from developed countries to reach a stage of lifestyle of these so-called developed countries. ICT projects are very true in such cases because technology promises to be the bridge to bring development into contexts where technological knowledge and resources are inadequate. The existing development ideology suggests that external agencies from developed countries provide the technology expertise and help solve social challenges in the under-developed countries through the use of technology. Instead of this indicator-determinist world-view, post-development school of thought is interested to study local culture and knowledge, critical stance towards established scientific discourse and defense and promotion of localized, pluralistic grassroots movement [12].

Among the concepts of Post-development theory, we use the following concepts:

1. Cultural relativism as an opposition to Ethnocentrism

2. Grassroots movements
3. Non-universalism

Ethnocentrism is behavior of an individual or groups of individuals to judge another culture solely on the values of one's own culture [13]. This bias is also termed as egotistic because of the pride that one experiences and views other culture as lacking in something. This behavior leads to the condition where culture from the so-called "South" or "under-developed" cultures are considered to be backward and need support. In the case of mHealth/eHealth, this means that support for technology change in health systems in under-developed countries needs to come from developed countries. Thus due to ethnocentrism, use of technology is considered self-evident as an improvement to health care systems independent of how the culture is accustomed in its forms of communication within its existing cultural practices and norms. In contrast, post-developmental theory looks at cultural relativism as the opposite idea of ethnocentrism. Through cultural relativism, a researcher or in our case implementer of mHealth system is able to accept the practices and customs of existing culture where the system needs to be implemented. The implementation is able to use principles of existing culture and apply them to mHealth or other forms of technology intervention. Cultural relativism does not mean that one's views are incorrect, but it does mean that claiming one's view as self-evident is inappropriate [14]. Thus, dismissing the notion of technology benefits because it was useful in the so-called "developed" world is incorrect, but questioning the principles of technologists, along with the view of the culture where the technology is to be implemented is highlighted by the concept of Cultural relativism.

Post-development theory uses the concept of grassroots movement to highlight the fact that change that is done from outside the system is generally resisted to by individuals on whom the change is being thrust upon [12]. The concept of grassroots movement highlights the fact that change is often most appropriate when coming from within the system and helps improve the system in a more sustainable way. This has been highlighted by other researchers [2] reviewing different mHealth projects and encourage the fact that there should be "South-to-South" collaborations to enhance mHealth. Most mHealth projects that we've reviewed do not look at the grassroots effect or movement as an impact of mHealth implementations. Evaluation of mHealth projects also lack any form of studying the grassroots movements that have occurred due to the interventions made by technology.

Non-universalism is the concept in post-development theory which suggests that there is no common process of development that can be followed in all contexts. Technology or any change that is supposed to result in development cannot be the same across different contexts. Non-universalism suggests that approaches to development are context-specific. This is to say that change needs to be examined through the eyes of the contextual principles that are embedded in the culture of the society.

3. Research Approach

The 3 authors of this paper are part of an international research network named Health Information Systems Project (HISP). The main activities of the HISP network consists of developing free and open-source software systems (FOSS), implementing them in conjunction with local partners. The HISP network does research through action-research methods by involving local agencies like health ministries and has implemented the District Health Information System (DHIS2) in more than 15 countries in Africa and Asia. The mHealth project (DHIS-Mobile) is tightly linked to the global DHIS2 project and aims to share learning between the different nodes of the network.

The 3 authors of this paper are particularly involved in the implementation of mHealth applications in partnership with the Ministry of Health, Malawi and are currently piloting the mHealth solution in 2 health areas in Lilongwe, Malawi. The research is conducted as Critical Action Research as described by researchers based on Habermas [15] and developed further by researchers like Kemmis [16] and Carr and Kemmis [17]. Our research is also guided by the *network of action* approach [18]. This approach is an addendum to the idea that local Health Information Systems research can be made more robust and sustainable by being part of a larger network and sharing experiences between the different nodes of the network.

Key informants for our study include medical personnel, health surveillance assistants, and statistical clerks, from all 17 health facilities that are part of the pilots. Training sessions we conducted for would-be users on the solutions under pilot involving three stages. Firstly, we conducted focus group discussions, with participants, covering topics such as existing paper-centric routine health data collection and reporting practices, data use at health facility level. We also discussed what sort of feedback health facilities get from the District Health Office, if any, on monthly reports they submit. Secondly, we had hands-on training on the DHIS Mobile solutions under pilot. The third part of the training was a feedback session on all matters covered during the training. This was done through another round of discussions and completion of pre-designed feedback forms. Data for this paper comes from close involvement with different partners who are involved in the project and its implementation. We've conducted focused group discussions and interviews with 22 community health workers, 2 health facility managers and 2 district-level health department officials. Every iteration of the mHealth application involves feedback and critical analysis of the data collected through the interviews and changes are made to the application based on the feedback.

For the purpose of understanding the context, we've collaborated with researchers working in other health information-related projects in Malawi, local master students from the Chancellor College of the University of Malawi. One of the authors of the paper has also been involved in review of mHealth projects in Malawi and understands the local culture, context and language of Malawi.

4. The Empirical Case of Facility-based Reporting mHealth Application

The global HISP network has been involved in the design, development and implementation of a suite of tools that can allow reporting health information from health facilities to district health offices. The application that we are implementing in Malawi is an evolution of a previous application that has been implemented starting in India [19] and then to other places in Tanzania, Gambia, Zambia and other countries. So based on the experiences and lessons learnt from other nodes in the network, we evolve the existing mHealth solution and work towards matching the mHealth solution to the context of Malawi.

To enhance our understanding and perspective of the context, we are piloting two types of applications through which community health workers and health facility administrators can report data to the ministry of health. One is a mobile web-browser based system that allows the user to open a website, fill forms and submit data to the ministry of health. The other system is a JavaME application that can store information on the mobile device and based on the user's input will send the information to the ministry of health's centralized servers. Both these applications contain the same forms that are to be reported by the health workers. We chose to implement only 2 health programs forms in our pilot phase because we wanted to compare the experiences of the community health workers with the change from paper forms to mobile-based forms. The Integrated Disease Surveillance and Response (IDSR) form is supposed to be submitted on a weekly basis and the HMIS-15 form is supposed to be submitted on a monthly basis. A total of 17 health facilities distributed across two health areas (Kabudula and Area 25) have been trained to use the application and have been reporting data to the ministry of health.

The project took off in the second half of 2011, through discussions with the Ministry of Health's Central Monitoring and Evaluation Division (CMED) and the Lilongwe District Health Office, on goals and scope of the project. Through the discussions, it was agreed that we run a pilot of the mobile phone-based reporting solutions in all health facilities under Lilongwe DHO. Lilongwe was chosen as a pilot district because it was the first district to implement efforts towards shifting from DHIS 1.3, which is a desktop-based solution, to a DHIS2, a server-based solution. The majority of districts in Malawi still run DHIS 1.3. Although Lilongwe DHO had started shifting to DHIS2, these efforts were put on hold and the office has gone back to using DHIS 1.3, due to technical support related issues that are beyond the scope of this paper. Nevertheless, the DHIS2 pilot is being run partly to revive the efforts and ensure that Lilongwe DHO shifts from DHIS 1.3 to DHIS2. At various points, we have also provided technical advice to the Ministry of Health on how to manage migration from DHIS 1.3 to DHIS2. Following this, we acquired 20 Nokia C2-00 handsets from India, for the first part of our implementation. We opted to get the phones from India, where they cost \$50 compared to around \$85 in Malawi.

To enable all health facilities send in monthly reports, we provide them with a monthly credit of MWK 1500 (~\$9), for voice calls. Thus far, we have not capped monthly Internet traffic to allow health facilities submit reports even when they have exhausted the allocated MWK 1500, allocated for voice calls.

4.1 Mobile Services and Internet Connectivity

Over time, we have also had problems relating to mobile service delivery. For example, it has taken about 5 months for our mobile service provider to cap voice calls for all the post-paid numbers we issued to health facilities, despite this being the agreement before we rolled out our solutions. Data reporting by health facilities has also, at times, been affected by intermittent GPRS/EDGE services. For example, at some point, we advised one health facility under Kabudula Health area to use a different service provider, from the one we are using.

At the start of our pilots, the HMIS and IDSR officers at Lilongwe DHO had no internet access. This meant that they could not ably access data health facilities had submitted using their mobile phones, to the online DHIS2 server hosted at the Malawi College of Medicine, in Blantyre. The two health area offices taking part also had no dedicated Internet access or access to the above mentioned DHIS2 Server. After noting these problems, we provided the HMIS and IDSR officers at Lilongwe DHO with Internet dongles, to enable them access the online DHIS2 server. We also provided Kabudula Health Area Office with an Internet dongle and oriented staff on how to monitor monthly data reporting, by health facilities under their jurisdiction. We were unable to get Area 25 Health Area Office connected, because their computers had been taken to the district health office for repairs, when we visited the office.

Getting the health area offices connected to the Internet and able to access the DHIS2 server, to which health facilities are submitting reports, is an attempt at creating an opportunity for the health areas to have access to tools for automated data analysis. This way, health area offices can be encouraged to utilize data much more, in decision making. Furthermore, they can also provide much needed guidance and feedback to health facilities, on various health service performance indicators

4.2 Multi-stakeholder Involvement and Systems Development Support

This pilot cuts across multiple organizations and geographical boundaries, which provides interesting opportunities to synergize competencies and efforts. Conversely, there are also various challenges in trying to align the interests of multiple stakeholders. Participating in this effort is the Ministry of health in Malawi, the University of Oslo, and Chancellor College (through the Mathematical Sciences Department). The pilot is also part of an ever changing landscape of mHealth in Malawi, where there is an ever increasing number of players and solutions being piloted and scaled. Realizing this, we are now part of mHealth-Malawi, a grouping of organizations implementing mHealth solutions in Malawi. The grouping is chaired by the Ministry of Health and is focused on harmonizing efforts on mHealth in the country, as well as developing guidelines on mHealth solution implementations in Malawi.

4.3 The grassroots movement of using mHealth

Our empirical findings have suggested some important advantages that make mobile phone-based submission of reports useful for the health workers and health facilities. For e.g., staff from health facilities indicated that when they have to physically travel to the district health office to deliver reports their travel costs are neither refunded nor subsidized. This results in the problem that they do not prioritize report submissions, but only submit reports when they are going to town to get their salaries. This is supported by the quote below:

“For the reports to be delivered well, we sacrifice ourselves...going to district office...using our own cash...so it’s a big challenge” (an officer from one of the health facilities taking part in our pilots)

When asked what they do if they have no money. Some officers mentioned that they don’t submit the reports or send the reports through some colleagues who might be going to town.

In addition, personnel from most health facilities also indicated that they are unavailable for service delivery to clients, at their health facilities, for a complete day, when they have to submit reports at the district health office. Most of the roads in rural areas are muddy during the rainy season and commuting to the district health office is extremely difficult. Since we also travelled to some health facilities during the rainy season, we experienced that without a 4x4 vehicle, it was practically impossible to reach the health facilities from the town center.

In addition, it is a known fact that health facilities in Malawi, especially those in rural areas are understaffed. Thus, we were requested by the health officers and district health office personnel to have solutions in place that allow medical personnel to submit reports on time, without adversely affecting their service delivery to clients. The use of mobile phones for data reporting is a promising route to take. Now after the pilot has been running for a couple of months in one area, we have evidence to show that mobile phone-based reporting can help address some of these challenges, as is indicated in the quote below:

“Previously we had problems with transport and stationery...now [reporting] has been simplified with the phones” (an officer from one of the health facilities taking part in our pilots)

Earlier when using paper forms, health facilities would send reports to the district, using ambulance drivers. Both personnel from health facilities and the district health office agreed that this option is riddled with any problems. Many a times, the ambulance drivers do not deliver the reports at the district health office. An example was given of the discovery of reports for three months, in an ambulance that was involved in a car accident. In addition to this, health facilities hardly get any feedback on the successful delivery of their reports.

The quotes below substantiate these claims:

“It is just unfortunate that this [the ambulance] is probably the best means of sending reports to the district, but we send [through] people who do not know the importance of the reports...I remember last time when one of the drivers had an accident people discovered that he had a pile of reports from various health facilities, not being delivered to the DHO (district health office) for months.” (IDSR officer, Lilongwe District Health Office).

We have held meetings with people working at health facility, district health office, and ministry levels indicate that meeting for data analysis are seldom held at health facility level. Previously, such meetings were common at health facility level, with support from a World Bank programme. The meetings died out when the program folded, as is indicated in the quote below:

“Since this was a project, the money was there, but when the project came to an end...they [health facilities and district health offices] had a problem to sustain it” (official, Ministry of Health)

5. Discussion and Analysis

From our conversations and feedback with field-level health workers, we have seen that mobile-based health information systems are useful for them. There are different factors on why the solution has proved useful for health workers in our pilot areas in Malawi. The challenges that we discovered as part of our research gives us enough evidence to justify that projects need to prioritize the problems they attempt to solve based on the requirements of the locale.

As mentioned in the beginning of the paper, that evaluation of mHealth projects is a global focus. We see through our action-research experience that current view of evaluation is simplistic. For example in evaluating Text4baby case study, researchers study the change in behavior of the patients through Randomized Clinical Trials (RCT) [20]. Here they associate patient behavior purely to the mHealth intervention, whereas there may be other socio-technical aspects that are reasons outside the Medical Center that cause such behavioral change. Fairly similar is the approach from researchers in New Zealand evaluating based on RCT [21]. Kaplan & Maxwell [22] suggest qualitative methods to evaluate Computer-based medical systems and critique the limited approach to evaluation such as RCTs and experimental designs. We support the idea of qualitative methods to evaluate, but would like to go a step further that without considering the nuances of the locale and without local needs being a priority, even qualitative evaluation methods are going to prove inadequate. We also find that some mHealth evaluations [23] where researchers have customized population surveillance applications to context and learnt lessons through implementations. But when looking at cost benefit of comparing to other devices, the researchers do not highlight the lack of power, maintenance of devices, use beyond data collection, changes in social structure due to devices etc. Using their approach of basing the evaluation on survey might be to only evaluate their application, but it does not evaluate the mHealth system holistically.

5.1 Information Delivery Efficiency

Transportation is a huge problem in our pilot health areas. Earlier when paper reports had to be delivered to the district health office using vehicles; there were a number of problems. The cost of transportation, risk of travel in bad weather and bad road infrastructure, misplacement of reports when sent through ambulance drivers are serious issues that decrease the sense of satisfaction of the health officers. Although the officers realize the importance of delivering reports to the district office and care about information and management done through information, their problems of submitting of reports through physical travel to the district health office is a bigger hindrance than their weighing of data reporting as priority. A Developmental change in the context of health officers is being able to improve their lives by making things simpler and decreasing the hassle in delivering reports. Thus, we discovered that the improvement of health interventions or disease surveillance was much less important to the health officer community than we had initially planned before starting out on the project. As researchers we realized that the cultural relativism of the context demanded that we understand these problems and cater our mHealth solution to these. In post-modernization of health systems, just as we had seen during modernization in enterprises, that technology enhanced efficiency of the system. Thus, not all mHealth projects need to focus on improving healthcare interventions. Some projects can also just focus on improving organizational efficiency and still achieve useful developmental impact to health systems.

5.2 Cost benefits

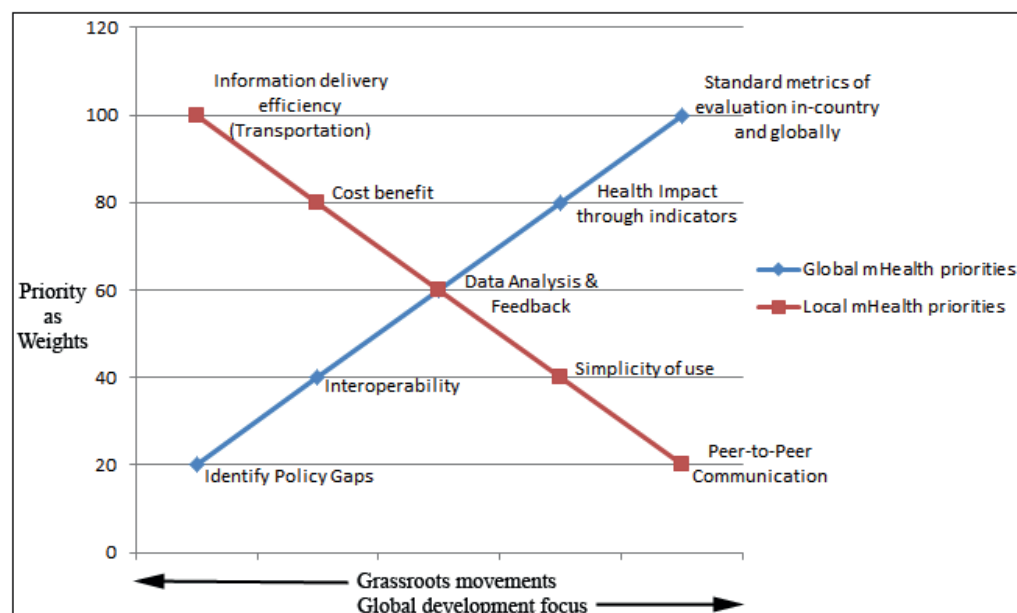
Introduction of technology (mobile handsets and mobile data services in this case) is an expensive affair. There needs to be careful analysis if the local context is able to sustain the cost of running an mHealth system. In our case, we observed that health officers needed to spend approx. MWK 1500 for a monthly trip to the district office. There are additional problems of stationary that the health facilities or officers have to bear out of their own pockets. The ministry of health has not been able to provide enough forms to the facilities for reporting. This makes us wonder if the ministry of health will be able to provide handsets or atleast subsidize handsets or internet costs on handsets for the health workers. Thankfully, we see that the locale of Malawi and the people already own mobile phones. Unlike any other communication technology, mobile phones are available with all health officers where we piloted the project and hence if we can use the mobile phones of the health workers, we make an important developmental impact to the health system, by reducing the cost of reporting health data from facilities to district or national level.

5.3 Data Analysis and Feedback

Beyond transportation there were other problems that were identified in our conversations. This was related to not getting adequate feedback for the reports that were submitted to the district health office. The district health offices complained to us that the reports were not delivered on time and hence they were not able to give feedback. They also suggested lack of training or resources to analyze the data that was received at the district health office. We have realized that visibility and power struggles are in place in the context and even if individuals at higher levels of health systems hierarchy were not able to analyze what was being sent, the lower level health workers still needed to submit the reports. In the post-development literature we see that this challenge of visibility of work is not just between so-called “developed” cultures and so-called “under-developed” cultures, but also between individuals because of the perspective of some people being more developed than others. Although we haven’t piloted a solution for the same, we’ve realized that simple automated delivery notification that can be sent from the servers will help build confidence within the health workers at the lower levels and will help motivate them to send reports more regularly.

5.4 Simplicity and Ease of Use

We have seen that health workers are burdened by the amount of work that is need for reporting data. It is important that the mHealth application should be simple and easy to use. From the design and development of the application, we have tried approach simplicity through two different solutions. We assumed that the browser based



solution would be harder to use compared to a custom JavaME application and hence wanted to compare the usability of the two applications. After our research, we realize that the health workers in Malawi were able to use the browser forms as easily as the application. The health officers highlighted the fact that they liked the simplicity of the mHealth applications because they replicate their existing expertise of paper forms. Both the application, as well as the browser form is similar to the paper forms that the health officers were used to from earlier. The fact that they asked for simplicity as a priority for mHealth application is not highlighted in the reviews of mHealth application that we've mentioned at the start of this paper. Thus, we see that although from a global development level, simplicity of mHealth is not a priority, at the grassroots level, it is important for mHealth to be simple if it has to make developmental impacts.

5.5 Peer-to-Peer Communication

In some of our communication with health workers, we have seen that they would like mHealth applications to allow improved peer-to-peer communication. This is interesting to note in the light of how social networks because of the use of technology have skyrocketed in terms of use. The grassroots health workers want to communicate more between their peers to be able to discuss and solve issues in the health system on their own. This highlights a difference in perspective from the global development level where improvements in health systems are primarily derived from communication between top-down and bottom-up levels of hierarchy. Here we observed that horizontal communication in the health system is a priority as much as vertical communication. If mHealth solutions have to make developmental impact, we suggest that solutions should prioritize horizontal communication in the health system and not just focus on vertical communication.

5.5 Comparing Global Development Priorities to Local Priorities in mHealth

In figure 1, we do a comparison between the weights of how local priorities are different from global development priorities. The figure is an interpretative weighted graph for the priorities. We see that local priorities are focused on information delivery efficiency, cost benefits, data analysis and feedback, simplicity of use and peer-to-peer communication. Although the global development priorities also intersect with the data analysis and

Figure 13: Comparing grassroots and global development priorities

feedback through a policy level focus, the other priorities are far different from the local priorities. These are only local priorities of our case and thus might be different from local priorities in other locales. Global development focus seems to be on standard metrics for comparison between countries or in-country health facilities, Health impact measurable through indicators, inter-operability between different mHealth solutions and identifying policy gaps. These global development priorities highlights that these focus on top-down design of mHealth systems. We have interpreted the global development focus from articles referenced in the introduction section. Although in [8], the researchers make similar recommendations and considerations for mHealth projects as we have discovered in our implementation initiative.

5. Conclusion

We conclude our paper by urging mHealth solutions implementers to respect the locale and prioritize solutions based on the needs of the grassroots. We also suggest to aid agencies and global health organizations to evaluate projects not just through the lens of global structures, but to consider the effects on the locale and changes that happen in the locale due to mHealth implementation. Thus, even if our project evaluation does not take into consideration how the health information helps take informed decision at the district or national level, we realize the problem of the locale has been solved through mobile communication that now makes the lives of grassroots health officers much better. This in their own and our opinion is appropriate development.

10. References

- [1] Consulting, V. W. (2009). mHealth for development: the opportunity of mobile technology for healthcare in the developing world. Washington, DC and Berkshire, UK: UN Foundation-Vodafone Foundation Partnership.
- [2] Curioso, W. H., & Michael, P. N. (2010). Enhancing “M-Health” With South-To-South Collaborations. *Health Affairs*, 29(2), 264–267. doi:10.1377/hlthaff.2009.1057
- [3] Sahay, S., & Walsham, G. (2006). Scaling of health information systems in India: Challenges and approaches. *Information Technology for Development*, 12, no. 3: 185–200
- [4] Kaplan, W. A. (2006). Can the ubiquitous power of mobile phones be used to improve health outcomes in developing countries? *Globalization and Health*, 2, no. 1: 9.
- [5] Rashid, A. T., & Elder, L. (2009). Mobile Phones and Development: An Analysis of IDRC-Supported Projects. *The Electronic Journal of Information Systems in Developing Countries*, 36(0).
- [6] Michael, P., Nemser, B., Cosmaciuc, R., Cole-Lewis, H., Ohemeng-Dapaah, S., Dusabe, S., Kaonga, N. N., et al. (2012). Capitalizing on the Characteristics of mHealth to Evaluate Its Impact. *Journal of Health Communication*, 17(sup1), 62–66. doi:10.1080/10810730.2012.679847
- [7] Tamrat, T., & Kachnowski, S. (2011). Special delivery: an analysis of mHealth in maternal and newborn health programs and their outcomes around the world. *Maternal and Child Health Journal*, 1–10.
- [8] Michael, P., Batavia, H., Kaonga, N., Searle, S., Kwan, A., Goldberger, A., Fu, L., et al. (2010). Barriers and gaps affecting mHealth in low and middle income countries: Policy white paper. Columbia university. Earth institute. Center for global health and economic development (CGHED): with mHealth alliance.
- [9] Escobar, A. (2011). *Encountering Development: The Making and Unmaking of the Third World*. Princeton University Press. 212-225
- [10] Escobar, A. (1988). Power and visibility: Development and the invention and management of the Third World. *Cultural Anthropology*, 3(4), 428–443.
- [11] Sachs, W. (1997). *The Development Dictionary: A Guide to Knowledge as Power*. Orient Blackswan.
- [12] Escobar, A. (1992). Reflections on “development”: Grassroots approaches and alternative politics in the Third World. *Futures*, 24(5), 411–436. doi:10.1016/0016-3287(92)90014-7
- [13] Omohundro, J. (2007). *Thinking Like an Anthropologist: A Practical Introduction to Cultural Anthropology*. McGraw Hill.
- [14] Zechenter, E. M. (1997). In the name of culture: Cultural relativism and the abuse of the individual. *Journal of Anthropological Research*, 319–347.
- [15] Habermas, J. (1987). *The theory of communicative action: Lifeworld and system: A critique of functionalist reason* (Vol. 2). Beacon Pr.

- [16] Kemmis, S. (2001). Exploring the relevance of critical theory for action research: Emancipatory action research in the footsteps of Jurgen Habermas. *Handbook of action research: Participative inquiry and practice*, 91–102.
- [17] Carr, W., & Kemmis, S. (2005). Staying critical. *Educational Action Research*, 13(3), 347–358.
- [18] Braa, J., Monteiro, E., & Sahay, S. (2004). Networks of Action: Sustainable Health Information Systems across Developing Countries. *MIS Quarterly*, 28(3), 337–362.
- [19] Braa, K., & Purkayastha, S. (2010). Sustainable mobile information infrastructures in low resource settings. *Studies in health technology and informatics*, 157, 127.
- [20] W. D. Evans, L. C. Abroms, R. Poropatich, P. E. Nielsen, and J. L. Wallace, “Mobile Health Evaluation Methods: The Text4baby Case Study,” *Journal of Health Communication*, vol. 17, no. sup1, pp. 22–29, 2012.
- [21] R. Whittaker, S. Merry, E. Dorey, and R. Maddison, “A Development and Evaluation Process for mHealth Interventions: Examples From New Zealand,” *Journal of Health Communication*, vol. 17, no. sup1, pp. 11–21, 2012.
- [22] B. Kaplan and J. Maxwell, “Qualitative Research Methods for Evaluating Computer Information Systems,” in *Evaluating the Organizational Impact of Healthcare Information Systems*, J. Anderson and C. Aydin, Eds. Springer New York, 2005, pp. 30–55.
- [24] Z. A. Rajput, S. Mbugua, D. Amadi, V. Chepng’eno, J. J. Saleem, Y. Anokwa, C. Hartung, G. Borriello, B. W. Mamlin, S. K. Ndege, and M. C. Were, “Evaluation of an Android-based mHealth system for population surveillance in developing countries,” *J Am Med Inform Assoc*, Feb. 2012.

P3: OpenScrum: Scrum methodology to improve shared understanding in an open-source community

Saptarshi Purkayastha

Department of Computer and Information Science,
Norwegian University of Science and Technology,
Trondheim, Norway
saptarsp@idi.ntnu.no

Abstract: While we continue to see rise in the adoption of agile methods for software development, there has been a call to study the appropriateness of agile methods in open-source and other emerging contexts. This paper examines Scrum methodology adopted by a large, globally distributed team which builds an open-source electronic medical records platform called OpenMRS. The research uses a mixed method approach, by doing quantitative analysis of source-code, issue tracker as well as community activity (IRC logs, Mailing lists, wiki) in pre and post Scrum adoption, covering a period of 4 years. Later we conducted semi-structured interviews with core developers and followed it up with group discussions to discuss the analysis of the quantitative data and get their views on our findings. Since the project is "domain heavy", contributors (developers and implementers) need to have certain health informatics understanding before making significant contributions. This puts knowledge-sharing and "bus factor" as critical points of management for the community. The paper presents ideas about a tailored Scrum methodology that might better suited for open-source communities to improve knowledge-sharing and community participation, instead of just agility

Highlights:

- Quantitative analysis of pre and post Scrum methodology adopted by the OpenMRS project over a period of 4 years.
- Interviews with core developers and focused group discussions with core and community developers to discuss the quantitative analysis and their views on findings of that data.
- Results indicate less agile, but improved shared understanding of design and code-base.
- More active participation in the community and developers feel more community focused.
- A modified scrum methodology is recommended that might be more suited to “domain-heavy” and community-driven open-source projects

Keywords: Agile software development; Scrum; Bus factor; Open-source software; Software Engineering; OpenScrum; OpenMRS;

1. INTRODUCTION

Appropriateness of agile methods for emerging contexts (open-source software (OSS), software as a service etc.), ranked first among the top 10 research agenda in the ISR special issue on Flexible and Distributed IS development [1]. Yet, we have seen limited research on agile methods within open-source communities. A recent review by Jalali and Wohlin [2] highlights that Global Software Engineering (GSE) projects with Agile methods are extremely rare. This might be primarily attributed to lack of clear mention that an open-source community is following a certain agile methodology. Some researchers have asked if open-source software

development is essentially an agile method [3]. But, Koch [4] mentions similarities, but also points out differences between agile software development (ASD) and OSS development. Thus, until the software development method being used by a community can be evidently clarified to follow one of the fairly well understood agile methods, they cannot be claimed to be the same.

Early work on ASD focused on defining agile methods [5] [6] [7], adoption of agile methods [8] [9], efficiency of agile methods [10] and then more recently focus on empirical studies about post-adoption issues of agile methods [11] [12] and team management [13]. While improved software quality is an observed output, the above researchers highlight “agility” as the most important criteria for adoption of ASD methods. “Agility” in such cases has been used to describe the ability to rapidly and flexibly create and respond to change in the business and technical domains. “Agility” is achieved by having minimal formal processes. Often used concepts to describe “Agility” include nimbleness, quickness, dexterity, suppleness or alertness. These ideas suggest a methodology that promotes maneuverability and speed of response [14].

On the other hand, OSS communities are generally seen as a collaboration of individuals or organizations that participate in software development without contractual bindings, but rather enjoyment-based intrinsic motivation [15]. Some researchers have suggested change in practices (like OSS 2.0) [16] [17], where OSS development is moving towards commercial participation. There is also more recent suggestion that OSS is still largely a combination of commercial ventures and volunteer contributions [18]. Sustainability is often an issue in open-source communities where volunteer contributors “come-and-go” or choose their own tasks [19] [20]. Sustainability of OSS is often described by using the term “truck factor” or “bus factor” i.e. the total number of key developers that would, if incapacitated (e.g., by getting hit by a bus), lead to a major disruption of the project [21]. Another challenge that we see in open-source communities is to gather contributors in projects that are for a vertical domain (health-care, finance, human-resources, etc.) [15]. In many cases, by strategic planning, paid developers will be assigned to work on open source products in vertical domains [16]. If the revenue model for such planning falls short, the developers are moved to other projects.

In this paper, we look at OpenMRS (Open Medical Records System), an open-source electronic medical records platform, which has adopted a tweaked ASD methodology. Since, the project is in a vertical domain of health, it is hard to find skilled volunteers who continue for long periods. Maintaining high bus factor is important for the project’s sustainability.

The paper attempts to answer the following research questions:

RQ1. How does adoption of agile methods in OSS change community participation?

RQ2. Can agile methods increase sustainability in OSS communities?

Beyond the above research questions, through the case, we hope that the paper responds to the calls for research of agile methods in OSS development. The case also highlights the challenges and opportunities of switching to Scrum methodology. The case is an avenue for reflection on open-source communities towards metrics of community participation. This will help them understand how their community is at present and what can be done to improve community participation.

The paper is organized as follows. In the next section we look at some of the concepts from software engineering research and agile methodology that have framed the formative and reflexive parts in the research design. In section 3, we describe the mixed method research used for this paper. In section 4, we analyze the tailored use of Scrum sprints in the OpenMRS community and detail out some effects on the project due to use of Scrum. In the discussion section of the paper, we highlight that Agile Methods can be used for knowledge management in open-source projects, instead of focusing on only the agility aspects. Here, we also suggest a tailored Scrum method, we refer to as OpenScrum that might be suited to open-source

communities. The last section of the paper concludes by suggesting evolution of agile methods in open-source communities.

2. CONCEPTUAL FRAMEWORK

More than a decade ago the Agile Manifesto clarified about the values of agile software development and put forth principles that can be adopted to meet those values. While much of the practices around agile software development have been promoted by practitioners and consultants, there has been a growing need to conceptualize “Agility” [22]. Here, Conboy suggests that Agility comes from two concepts of Flexibility and Leanness. Although used interchangeably, there are conceptual differences between Flexibility and Agility and also between Leanness and Agility. Thus, to be considered agile, the methodology should contribute to creation of change, proaction in advance of change, reaction to change or learning from change. It should also contribute and not detract from perceived economy, perceived quality and perceived simplicity. These allow producing software which is continually ready, with minimum time and cost required to be put into use (*ibid.*).

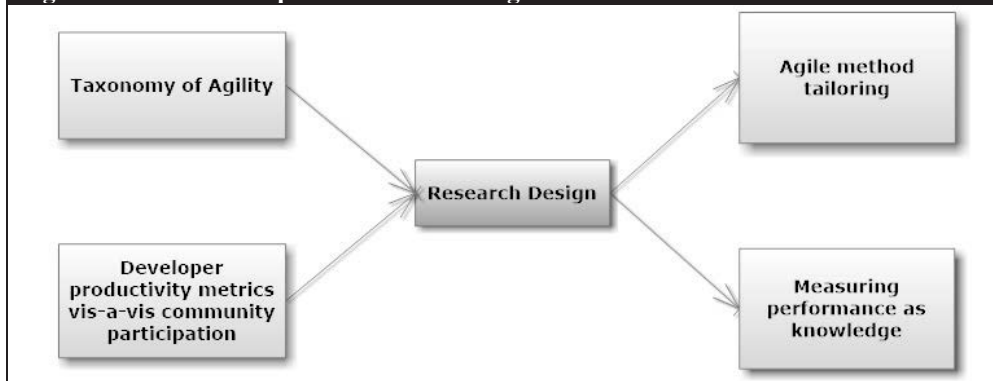
While Agility in such terms is an overall measure of the organizational performance to deliver a software product, one should also consider how individual developer productivity is affected by the practice of agile development. While developer productivity has been a hotly debated topic, the 1993 IEEE standard for software productivity metrics defined it as “the ratio of output to the input effort that produced it”. Jones [23] identified 250 factors affecting developer productivity, while more simplistic summary still lists 15 factors [24]. So instead of co-relating multiple factors that affect productivity, it is common to measure output such as in Changes in Lines of Code (CLOC) or Non-Commentary Source Lines (NCSL) [25]. Another measure of developer productivity through interactive participation has been suggested – mainly through the use of code reviews, comments on other people’s code, number of forks, network analysis of contributors [26].

We’ve seen case studies which suggest that communication, co-ordination and control problems in GSE have reduced due to use of agile methods such as Scrum and eXtreme Programming [27]. This and similar research [28] [29] [30] suggests that distributed teams indeed benefit from using agile methods. From all of these cases, we see that there is some level of tweaking done to agile methodology to be relevant to the organization. Tailoring of methods has been observed to play an important role in benefits like reduction of code defects density, delivery ahead of schedule and accurate planning for future projects [31].

While this need for tweaking has been well documented, very little has been written about tweaking agile development to open-source projects. OSS projects might simply be GSE projects in the public domain. Yet, as highlighted in the introduction, sustainability of OSS projects that are managed through community contributions or those that involve multiple stakeholders with different interests, highlight the need for a different kind of tweaking. In all of the above research on GSE, and most research on agile methods [32], we see that management control for changing software development practices could be done by a limited number of stakeholders and all these stakeholders were either organizationally or contractually bound. Even in earlier mentioned GSE literature review [2], “Agile-Open” projects were largely centrally governed. In antithesis to these contexts, consider the Linux project, which in 2011 had 7800 developers for 800 different organizations and 75% of these developers are paid by companies to work on the Linux kernel. Is moving to an agile methodology possible for such a project? What kind of tweaking of agile method will be needed for such a community is an unknown proposition. May be using Linux as a poster-boy for open-source projects is not a useful exercise. But even in small open-source projects which have multiple stakeholders,

working with the organizational challenges of adopting agile methods is highly relevant. Figure 1, shows the research design, where we attempt to study agility, as factors of change as described earlier by Conboy and attempt to discuss developer productivity in terms of community participation. The resulting tailored OpenScrum is an output of the research along with measures for community participation.

Figure 1: Use of Concepts in Research Design



The previously introduced concept of “bus-factor” can be understood as a function of knowledge distribution. The more widely distributed knowledge in a community; higher is the “bus-factor”. One needs to look at bus-factor also through decision making capacity. If a project has one person making all decisions, the bus-factor is 1 and if this person gets hit by a bus, the project is in jeopardy. Getting hit by a bus, should not be taken literally. It refers to any event that can lead to the unavailability of an individual to the organization. Thus, the processes of KM to increase bus-factor in a software development organization would result in spreading information and decision-making capacity [33]. Infact, at this point in the paper, it is important to mention that the term “Scrum” was coined by Takeuchi & Nonaka [34], which is the agile development method used in the paper’s case. Important concepts from their research highlighted, “multi-learning” and organizational transfer of learning. Multi-learning highlights the fact that learning by doing manifests itself along two dimensions – multiple levels (individual, group and corporate) and multiple functions. Knowledge is also transmitted in the organization by converting project activities to standard practice [35]. Thus, OSS project can be understood to follow an agile methodology when individuals as well as the community as a whole, implements agile principles and processes.

3. RESEARCH METHODOLOGY

The research followed a case study methodology [36] to understand the effects of agile methods in its natural context. Case study method is also useful to study post-facto effects, where theory and research are in their formative stages [37]. The research employs a mixed method approach [38] with initially taking an interpretive approach with quantitative methods and later interpretive approach with qualitative methods [39]. Data collection was done from the issue tracking system (JIRA). Individual work units are henceforth referred to as tickets. We analyzed emails from mailing lists (developer [n=18318]; implementer [n=8316], announcement) and source-code; covering the period from Jan 2009 to Jan 2013. Over 3000 tickets were analyzed for factors such assignee, reporter, priority, creation time to resolution time, linkage to source-code and linkage to a sprint or software release. This was done through the use of JQL queries that allow retrieving issues based on selective options from JIRA. Source-code was analyzed in co-relation to the tickets and measured according to the changes in lines of code per developer, number of commits, refactoring of existing code, unit tests and

code comments. The research covers code from OpenMRS svn¹ repository for distributed modules, as well as code from git² for the OpenMRS core, migration to which happened in August 2012. An Ohloh.net project was created for code analysis by listing various code locations. Additionally, a tool called Fisheye from Atlassian Inc. was used for analyzing activity by developer in terms of code commits and code reviews. Nabble.com was used to get aggregate information about individual contributors on the mailing list. Text mining was not done on the contents of the mailing list, but analysis was done only on the name, email and known organization from the sender's list. Documents on wiki pages which describe design, development and use were analyzed through an interpretive perspective. The wiki is used to collect summary information about discussions and often as a knowledge base about design decisions taken by the community. IRC logs were analyzed for the number of active participants in the IRC, as well as the number of lines of communication was collected to measure the activity in the IRC, similar to that of the mailing list. The mailing list was used to differentiate between developers and implementers. Individuals who have more than 10 emails to the dev list are identified as developers, where as individuals who have more than 5 emails to the impl list are identified as implementers.

This quantitative data was interpreted in relation to the different concepts of agility as presented in the previous sections. This analysis was then shared with each individual core developer through a set of semi-structured interviews which last about 45min to 1hr. A total of 25hrs of interviews were done and 3 group discussions were organized with the core developers. The interviews were transcribed and entered into Nvivo, qualitative data analysis software. Then performed coding based on concepts of “learning”, “agility”, “knowledge”, “release cycle”, “participation” and performed thematic synthesis [40]. The resulting themes from the analysis were matched against quantifying words like “more”, “less”, “increase”, “decrease” to verify that the interviewees described the concepts across the interview in the same increasing or decreasing order. Beyond discussing interpretations of quantitative data, opinions were asked on a wide variety of topics such as community participation, developer workload, project management and software development methods in the OpenMRS community. This resulted in deeper understanding of the phenomenon and allowed drawing upon interpretations of core developers. These discussions helped meet the principles of interpretive research [41] such as - principle of contextualization; principle of interaction between researcher and subjects; principle of dialogical reasoning; principle of multiple interpretations - each of which helps bring rigor and validity to the findings.

As in any research approach, case study has its strengths and weaknesses [42] [43]. Case research is important for this type of research, as it allows for study of a large number of variables in a given setting, while these variables do not have to be previously defined [44]. The weakness of such case research is that it is hard to make generalizations or be able to draw conclusions that can be claimed to be valid for all open-source projects. But I take the view that OpenMRS is indeed representative of many similar open-source software communities that work in a vertical domain and have a similar governance and participation model. The OpenMRS governance model is community-driven. Issues are created by community members, weekly developer meetings, weekly implementer meetings, design discussions are on public mailing list or during the weekly meetings. Code review happens in public, voting is used to prioritize features etc. There is a newly formed OpenMRS Foundation with an executive board and community members vote to put a member on the board of directors. Most day-to-day decisions are not taken by the board, but instead through community discussions.

¹ Svn or Subversion is a centralized version control system where source code is stored and versioned

² Git is a distributed version control system, which allows developers to fork code and work separately on same parts of the source-code.

The leadership of the OpenMRS community has tried to model itself similar to Mozilla, including having the ex-CEO of Mozilla on the OpenMRS board to get a better understanding of governance principles.

3.1 CONTEXT AND RESEARCHER ROLE

In the paper, I attempt to contextualize ASD in OpenMRS as much as possible. Krutchen [45] highlighted the importance of contextualizing. However, due to length constraints, we don't describe the context of ASD using the full "frog and octopus" model [46], but make maximum attempt to describe all areas, although not as separate sections. OpenMRS is a software platform and a reference application which enables design of a customized medical records system with no programming knowledge (although medical and systems analysis knowledge is required). It has a modular design, where modules are add-ons that extend the functional scope of the system. There are 76 modules installable from the OpenMRS module repository, 125 modules have their source-code in OpenMRS svn. While there are close to 220 OpenMRS modules that are openly available from different sources (github, bitbucket, sourceforge), yet this is only a rough estimate of available modules. Most modules are developed by developers who are not part of the core team. These modules cover broad range of functionality and there is a clear separation of openmrs-core, which has distinct software development lifecycle from modules. While the focus of this research is openmrs-core, we include some modules which are distributed along with the reference application called core & bundled modules. These include FormEntry, HTMLFormEntry, Logic, XForms, DataEntryStatistics, SerializationXStream, Reporting, ReportingCompatibility, HTMLWidgets and PatientFlags. Unless mentioned otherwise, the paper refers to OpenMRS as the "core + distributed modules". I have been involved in the project as an independent developer for about 6 years, without direct funding from any organization to be part of the software development process. I have also spent a summer internship through Google Inc. at OpenMRS in 2008, through which closer engagement in the community had started. I have been identified as a contributor to the core for many years and have been actively engaged in different roles – as developer, implementer and consultant at for-profit and not-for-profit entities that use the OpenMRS platform. I have participated in many design discussions, roadmap decisions and overall community management discussions before this research. Over the years, I have developed few open-source modules that are used by implementations all over the world as well as proprietary modules that are used by for-profit and not-for-profit global organizations. All of this highlights that I already had a deep understanding of the community and its practices (implicit and explicit) including roles of core developers and other community members. Walsham [47] classifies such style of involvement as "involved researcher" while doing interpretive research. Yet, the motivation and the decision-making process of changing to agile method, (specifically a customized Scrum method) from a global, distributed software development model were not known to me clearly before this research. This is because the decision was taken by the OpenMRS leadership group and was announced to the community through the developer mailing list. More on the motivation and decision-making process for adoption of agile methodology is covered in the next section.

4. THE EVOLUTION OF SOFTWARE DEV METHODOLOGY IN OPENMRS

OpenMRS is inter-twined software and community, similar to many open-source projects, where the software application development community describes itself as product developers as well as clients of the product. This intertwining is fairly evident in how developers are implementers and implementing organizations contribute developer resources. Yet, some community members are purely implementers who do not have developer resources, while some OpenMRS core developers are allocated only to "*core OpenMRS tasks*" and not to

implementation-specific requirements. As we have found out in the study, this is a fairly hard task to balance.

4.1. The globally distributed open-source development model

OpenMRS software development before the move to ASD was like most other community-driven open-source projects. Certain developers are maintainers of specific modules or parts of the system. These developers work for different organizations and have their organizational interests or personal interests. The project started in 2004 as collaboration between two health care organizations and quickly expanded into a global, open-source software community [48]. The project at the time of research had 218 code contributing developers distributed across the globe. Only 41 of these developers are from organizations that implement or support OpenMRS installation. The independent contributors are those who have “come-and-gone” into the project from time to time. The unknown developers are possibly individuals who want an internship position or give a shot at contributing to OpenMRS, but do not actively contribute. In Table 1, we see activity of developer in the community with reference to emails responses, code commits, code reviews. A 30-day period of no-activity makes the developer dormant and continuous activity count is reset.

Table 1: Types of OpenMRS developers and active developers

	No.	Average days active (out of 1460 days)
No. of OpenMRS core developers (C)	13	411 days
No. of organization-backed developers (OD)	41	63 days
No. of intern students (I)	84	71 days
No. of independent contributors (ID)	12	95 days
No. of unknown developers (UD)	68	12 days

This shows that although the core developers are moving forward with development, there is active participation from a larger community of developers, who participate in all types of activities ranging from engaging in discussions to doing code reviews. The role of unknown developers, independent contributors and large number of interns is different and unique to open-source organizations.

In OpenMRS a major percentage of the new developers to the project are student interns, who contribute for a summer or limited period of time. Nearly 75% of the developers fall into category of “community developers” and make up for largest opportunity for the organization to increase commits from these developers from the current 23% levels. This means that without any additional direct cost to the organization, the number of commits can be increased, if the potential of these contributors can be tapped into. On the other hand, OpenMRS has approached the challenge of growth by hiring intern students as Full-time Engineers (FTE), either as core developers or into support organizations. While other open-source projects, mainly run by for-profit organizations have introduced the concept of “bounty”, either monetary or “in-kind” [18] to developers who submit code or find specific bugs, OpenMRS has not done anything similar. Thus, the motivation for code submissions to OpenMRS has generally been intrinsic.

As highlighted in the introduction section, volunteer contribution gives rise to a sustainability challenge. OpenMRS also does not make concerted effort to reach out to dormant developers, as highlighted by a core developer. *“We would like to see developers return, but student developers generally remained active only due to Google’s funding over the summer and go away to other paying jobs after graduating”*. When asked why, there is no effort to reach out to dormant developers, a general consensus was lack of analytics to know, when and why people drop out of the project. One interviewee mentioned, *“We are experimenting with a CRM*

system that will be used to track contributors and developers. This will allow us analytics and look at these contributors as leads”.

While analytics and tracking of developers till date has been somewhat easy, OpenMRS leadership realizes that as they’ve moved to distributed version control and also grown in the number of participating organizations, it continues to become harder to trace code changes across the community. Multiple forks of modules as well as core have been done in recent months by different organizations and developers. But much of the features from these, experimental forks has not returned to the mainline of development. One core developer highlighted, *“We are less concerned about forks. We don’t know whether to encourage or discourage forks, when we don’t have a mechanism to lure those forks to submit pull requests. We have moved to github and forks are inevitable, but at least easy to track”*. While, this suggests the general approach in open-source, that “free will” of the partnering organization or contributing developers be respected, it is important to realize that governance and management need active action, instead of passive observation.

In the past, OpenMRS developers have been responsible to maintain and contribute to their own modules. Core developers would generally choose tickets on parts of the system that they are familiar with, and new developers have a set of introductory tickets that have been identified for attempt by new contributors. The developers come from different time zones and are expected to work on the tickets over their estimated time. Before the scrum methodology was adopted in March 2011, developers would generally not have a scheduled meeting, but would hang out in the IRC room. There would be discussions around problems, if a developer had trouble resolving the issue on their own. But there was no formal group communication through which one developer could communicate with all other developers. Also since most developers were themselves the maintainers of the modules, they would rarely get useful responses to resolve an issue. As one developer quoted, *“My module was considered to be black magic that just worked. No one else looked at it and if anything broke, I would be the only person to know where to quickly find it. Instead of showing someone else how to fix it, I’d go and fix it myself”*. While most developers were comfortable doing this, if a developer went to vacation or was helping out an implementation, tickets belonging to that developer’s module would be ignored. Bugs remained unfixed and changes took longer to release. Bus-factor in such cases was 1 for many modules and more than 83% of the modules did not receive updates, even though feature requests were made by implementers.

OpenMRS also maintains 2 previous versions, after a new version of core is released. The process of releasing maintenance versions of previous releases is done through the use of what is commonly referred to as backporting of a fix. New features are rarely added to maintenance releases, but sometimes highly voted and relevant features are indeed backported. But in general, maintenance releases only contain bug fixes. While this is useful for implementations to use only the stable functionality, the process of backporting is a time consuming process and there is some lack of clarity in what is allowed to be backported (bug fixes), but sometimes new releases of external libraries, data model changes have also been included in maintenance releases. In Table 2, you can see the release cycles for major version releases. Maintenance releases have not been included for the core, but the bundled modules have their maintenance release distributed and those have been included in the table. Only tickets on which work started and finished between the release period have been counted and long pending or worked on tickets have been ignored. A detailed analysis between of maintenance releases of core and modules is done later in the paper, when we compare the change in performance after adopting Scrum methodology in section 4.3 below.

Table 2: Releases timeframe			
Version	Time to release	Tickets resolved = Core + modules	No. of contributors
1.5.0	116 days	177 = 114 + 63	35

1.6.0	202 days	$322 = 293 + 29$	40
1.7.0	233 days	$263 = 167 + 96$	50
1.8.0	275 days	$320 = 189 + 131$	49
1.9.0	352 days	$636 = 451 + 185$	71

While it takes months before large OpenMRS implementations move to a new release, maintenance releases allow implementations to fix bugs and get important performance benefits. As one implementation-support developer quoted, *“The 1.8.0 release was a paradigm shift in how we were fixing issues to deal with implementations. There were 2 quick maintenance releases made because of performance improvements. Supporting large implementations is about running the right modules, with the right core”*. In similar light, a core developer mentioned, *“Implementations differ from each other because of the modules that they use. So, when implementation-support developers commit code, it is to the modules that they use in production. They care less about the core, unless something is breaking a module”*.

While the realization that core was getting less relevant for the implementations, it is also understood by the developers that modules needed to be developed at a separate pace from the core. OpenMRS started “Sprinting” using the scrum methodology between the 1.7.1 and 1.8.0 release. The scrum methodology as adopted by OpenMRS community is described in the next section.

4.2. Adopting a tailored Scrum methodology

Scrum is a popular agile software development methodology that focuses on project management in situations where it is difficult to plan ahead; where feedback loops constitute the core element [49]. The core aspect of scrum is the “time-boxed” effort called sprint to complete a set of tasks known as sprint backlog, which have been selected from a larger product backlog [50]. Instead of discussing general aspects of Scrum methodology, which is well understood and can be read elsewhere, we focus on the use of scrum in OpenMRS. From here on, this is referred to as OpenScrum.

OpenMRS sprints are designed in a way that all participating developers – core, organization-backed as well as community developers work together in sprints of 1 or 2 weeks depending on the module or task at hand. The sprint duration is generally suggested by one developer (who knows the module) or project leader based on their guess of the complexity at hand and from roadmap requirements from implementer meetings. Sprints are also sometimes proposed by developers from the community, implementers or core developers. A sprint schedule is advertised 2 weeks in advance through the developer mailing list and planning starts by nominating or volunteering a Sprint leader. The Sprint leader should be a developer, with adequate knowledge about the module. This developer decides the sprint backlog, by creating new tickets or allocating existing tickets from the product backlog for the sprint. This process is somewhat different from the general role played by Stakeholders and Product Owners in textbook Scrum method. The OpenMRS sprints have the community as the Product Owner and the community as a whole decides what tickets to prioritize by voting on tickets. Although, this has been contentious within the implementer community because core developers have higher chances to influence voting. Also implementations which do not have developer resources are under-represented in such voting schemes. The list of developers who are participating in the Sprint is continuously updated in the sprint schedule. The Sprint leader has to monitor the list of participants so that the tasks that need to be completed in the sprint do not exceed the amount of time that would be required to complete the product backlog. These estimates are fairly

complex to make since time estimate of independent developers cannot be accurately made by the Sprint leader. Sometimes, although independent developers self-nominate to participate in Sprints, they do not actively engage and do not spend adequate time during the Sprints.

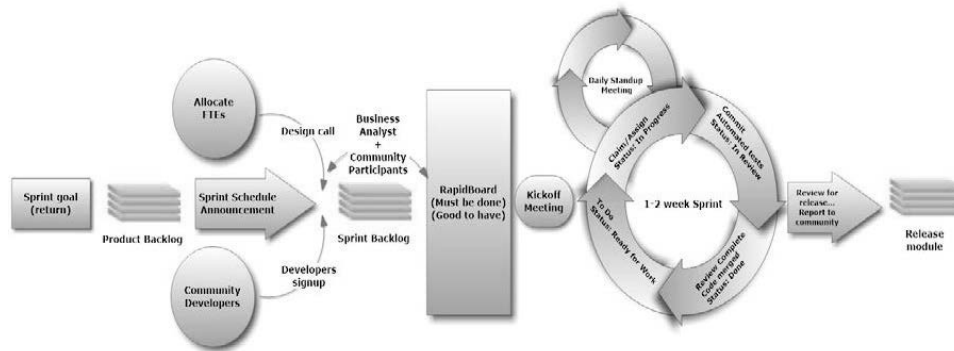


Figure 2: A high-level view of the OpenMRS Scrum

During the planning phase, the design calls play a vital role to translate requirements into working tasks as tickets and to make a guess about what tasks can be completed. OpenMRS involves community members along with a Business Analyst role to complete this process. Implementers are ideally useful for this process, but lack of participation from them, has made the role of Business Analyst to be more important. Community developers sign up for the sprint during this period and FTEs from OpenMRS are allocated to the project before or during the design calls. The output of the design call is the creation of a RapidBoard, which lists out all the activities that need to be completed in the sprint along with their priorities. Figure 3, shows an example RapidBoard, which is representation of the Sprint backlog.

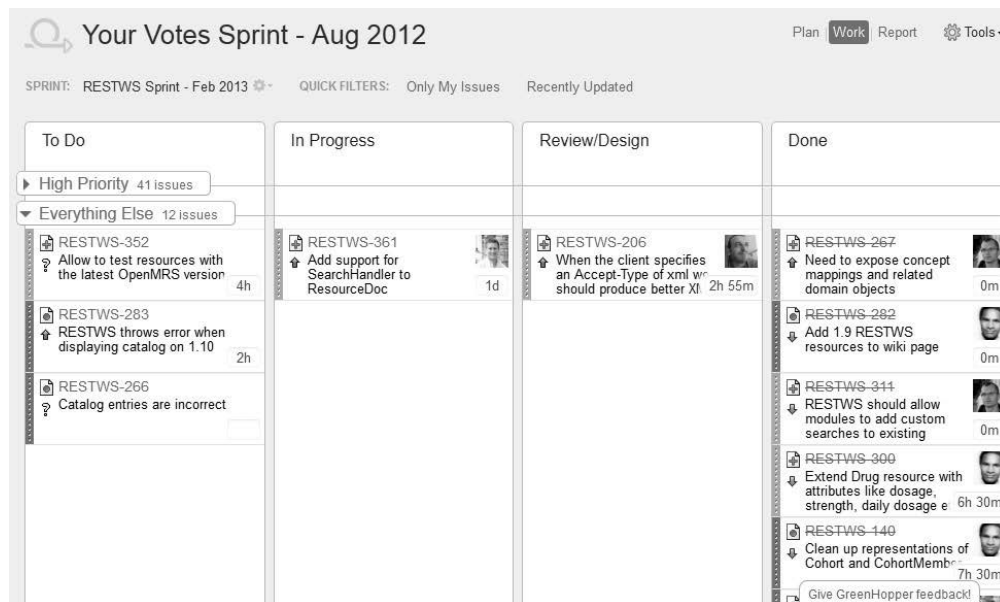


Figure 3: The OpenMRS Scrum RapidBoard

The RapidBoard is essentially an information radiator [14] like a pinup board that is updated automatically and shows the status of a sprint. The OpenMRS sprint starts with a kickoff meeting, which is generally at the IRC. The goals of the sprint, what steps to follow, how to commit code, how to review code, how to merge, what unit tests to write etc. is discussed in this meeting. The prioritized tickets are described in the meeting and community developers introduce themselves to each other during this kickoff meeting. The appropriate wiki pages are highlighted in the meeting; so that new developers can become well versed with some design decisions as well as coding standards to be followed for the sprint.

The OpenMRS sprints also do not have a clear role for a ScrumMaster. A project co-ordinator has recently played the role of the ScrumMaster, yet it is unclear how this role has been used to enforce rules. In textbook Scrum, the ScrumMaster is intended to protect the team from distracting influences, ensure that the rules of sprint are followed and help in gathering resources for sprints. When moving to Scrum methodology, OpenMRS did not have the role of a ScrumMaster. Since July 2012, a ScrumMaster role has been defined, yet activities of the role are vague in OpenMRS. There has been training through OpenMRS University conference calls to explain to the community, the different processes that are followed in OpenMRS scrum. These university calls have had limited participation and there is lack of clarity in the conceptual terms that are part of the followed methodology. In one of the group discussions of community developers highlighted this, *“We would like to have implementers to be ScrumMaster”*. Another developer disagreed, *“May be Scrum leader needs to be a developer, but Product Owner should be a single individual instead of the community, so that this person can tell us that the module is ready to be released”*. This shows some conceptual lack of understanding among the developers regarding the scrum methodology in practice and textbook definitions of roles in Scrum.

The OpenMRS implementers mailing list (n=8316) receives about half the amount of traffic compared to the developers list (n=18318). More than 65% of the responses even on the implementers mailing list is from developers. At least from the people who interact openly, we can infer that the OpenMRS community is largely developer driven. This highlights the problem of getting an active role for Stakeholders as well as Product Owner. For instance, when there was an announcement made to organize roadmap meetings which would enable community members to prioritize issues, to create a product backlog that will be used in sprints, after 4 months of attempts to meet, there was no active participation in these calls. Thus, the role of Stakeholders in most sprints has been largely absent. The Product Owner is also different for each sprint, depending on the module being developed in the sprint.

As the sprints continue, there is a daily standup meeting in the IRC, where developers mention what they've been working on and highlight if they have any blockers. Blockers are attempted to be resolved by advice from the core developers soon after the standup meeting. The RapidBoard keeps changing but does not include any time for preparing for release. Merging pull requests, documenting changes, prepare documentation for release are supposed to be done by the Scrum leader, but the OpenMRS scrum model does not have separate time allotted for these kinds of work. The Scrum leader announces at the end of a sprint the results in terms of tickets completed, burndown chart etc. Figure 4 is an example:

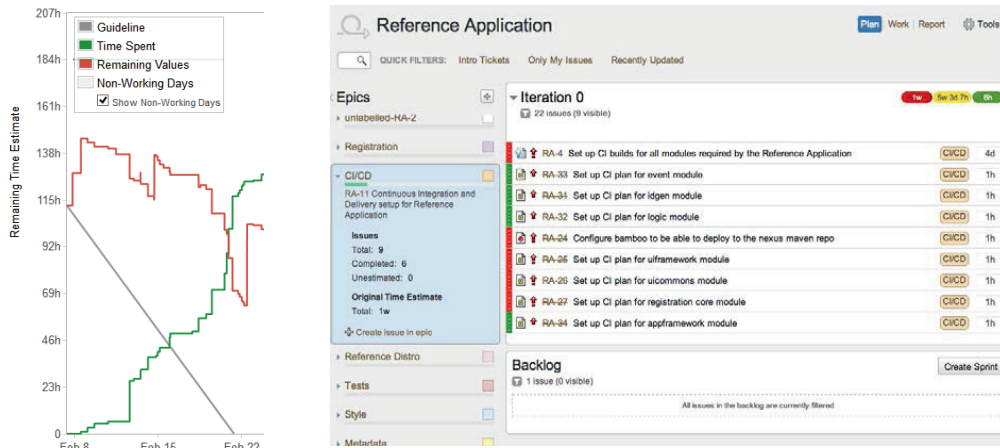


Figure 5: Planning board from Epic to tickets

Figure 4: Burndown chart

4.3. Findings and Analysis of the OpenMRS Methodology

While it is evident and somewhat expected that any change in development methodology will result in initial slowdown, we do see that some degree of inefficiency in the way OpenMRS moved to the new methodology. Let us look at the OpenMRS methodology in terms of the Taxonomy of Agility described earlier. It is a combined interpretation of Flexibility and

Table 3: Agility parameters for OpenMRS			
Agility	Interpretation	Leanness	Interpretation
Creation of change →	Good increase	Perceived economy →	Unclear (NA)
Proaction before change →	Little increase	Perceived quality →	Good Increase
Reaction to change →	Moderate increase	Perceived simplicity →	Slight decrease
Learning from change →	Good increase		

Leanness. Table 3, summarizes the findings after analyzing the parameters.

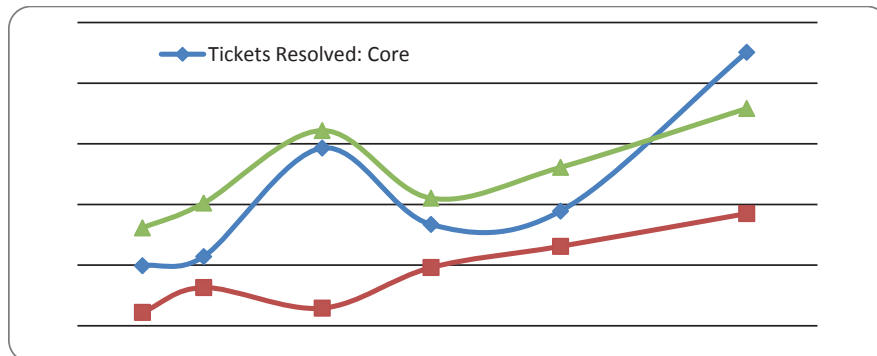


Figure 6: Tickets resolved in each release

Creation of change

Creation of change in a software artifact can be seen from a few different angles. Releases are one way to look at it. We looked earlier that the number of days to release had increased, but that's only the core releases. There was a concerted transition to focus from core to releasing modules quickly. In Figure 6, we see that although there is a clear increase in the tickets resolved over the releases, the number of participating contributors have also increased. Thus, the average contributions (normalized to 40 contributors) in terms of resolving tickets have not really drastically increased. Though it is interesting to observe that after shifting to OpenScrum, there has been stability in the avg contributions from the community. Another interesting fact is that although the focus has indeed shifted to modules, the bundled modules are not the ones that have shown dramatically higher signs of extra work. They seem to be getting the same amount of work done as earlier.

As mentioned in the research design, contributor productivity is another factor of study. Figure 7 shows the CLOC for the top 5 code contributors every month. These are not the same developers, but the top contributors in terms of CLOC to the project for each month. We see that soon after adopting the Scrum methodology the CLOC of the developers went down, but over time the top 5 developers have been committing equal amount of code to the project. This shows that individual developer productivity has become less disparate since adopting OpenScrum.

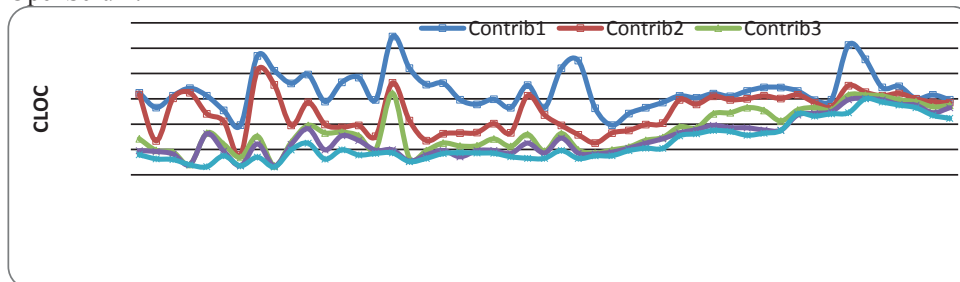


Figure 7: CLOC for top 5 developers

Proaction in advance of change & reaction to change

Lean organizations are often considered to be proactive to change and visionary leaders have the wisdom to see the incoming change in market conditions (Takeuchi & Nonaka, 2011). Out of the total of 48 sprints between Mar 2011 and Jan 2013, there have been 25 sprints that have been on Core (n=19) and Bundled modules (n=6). Others have been on experimental features (referred to as spikes i.e. sprint by 1-2 FTEs) or popular community modules. Yet, as one lead developer commented, *"We wait till something is upon us and then we start experimenting in spikes"*. Another developer added, *"We've needed 2 or 3 spikes before anything experimental"*.

has been converted to usable modules. Look at RESTWS or OCC". This adheres to OpenMRS's vision of using the "Story of Floss" – "*Whenever possible, start with the floss (sic: dental floss). See the solution through end-to-end, since this is often the best way to understand the problem and informs the next pass at the solution. In the end, it is rare that we fully understand the problem until the third iteration of the solution. Be agile, open to corrections, and iterate on your solutions. But, most importantly, take action*". From these we might infer that due to sprints and spike-style development, experimentation and chances for innovation have increased. For example, 9 sprints in this period have happened on experimental features that aren't considered to be traditional EMR platform features. These sprints do not directly benefit the large OpenMRS community. Also, decision to run such sprints have not been debated in the OpenMRS community. The leadership of the OpenMRS due to political or with motivations to increase OpenMRS adoption have allocated such sprints. On the other hand, one could also argue that experimental spikes results in lower quality, but better chances of innovation. Quality might be improved in subsequent iterations, if an innovative solution has been found. Thus, we could summarize that we see only moderate to low levels of proaction in advance of change and moderate increase in reaction to change due to adopting this methodology.

Learning from change

There have been 6 sprints on the RESTWS module and each time there was decreasing activity in the number of issues resolved as well as CLOC. This might point to stabilization of code, while moving from experimental spike to quality improvement sprints on which all the devs work together. The number of reported bugs against RESTWS has decreased with each release of the module. Another vital aspect has been that all developers now work on same part of the system and learn similar lessons. As mentioned earlier, developers now learn more and look at varied pieces of code that they would earlier not work on. This includes writing code, as well as reviewing code submitted by other developers. This allows the developers to learn much more and adapt to the different coding styles of developers.

Perceived economy, quality and simplicity

As part of this research, we've not analyzed factors that look at the economy or cost of the development process. During the interviews the managers did not have any kind of cost-estimates before or after the change in the development methodology. Thus no direct or indirect observations could be made on the economic parameter.

The perceived quality of the code has been accepted to be better by the developers as well as the community in general. In 2010, a practice of continuous integration was adopted, which ran unit tests as soon as any code was committed. Along with this, a practice that 2 core developers will need to review code before any ticket gets closed. Although this ensured quality, it was often only the 2 lead developers who would review code. Other developers including community developers would hardly comment or do formal code reviews. With the change to OpenScrum, more community developers are reviewing code and commenting on each other's code. This is because the developers are watching each other's changes more closely. As one core developer highlights, "*Now missing javadoc comments like @since for newly introduced methods is suddenly more obvious to reviewers. Also writing unit tests is a necessity because it's the reviewer's first comment*".

OpenScrum hasn't necessarily made things simple. Earlier, each developer would fix their own modules, work on parts that they were comfortable with. New modules would be based on requirements from an implementation that would have their developers working on it. The OpenScrum development processes - checks and balances have made things more complex. The GSE method was simple, in the sense that people could work independently, with less effort in co-ordination. The down-side was that with more and more people writing their own

modules and with growth of the community without any governance processes, it was becoming harder to ensure quality. So, it is probably worth discussing that for open-source projects although co-ordination is a difficult task, it is worth pursuing, to ensure better quality.

Challenges with Time-boxing and Community Participation

One of the main goals of sprints is that the team focuses on finishing planned set of features in a time-boxed manner. Short cycles emphasize that quick progress needs to be made, even though things can be improved over the next iterations. Another focus of Scrum is that the sprint backlog is completed in a manner that a product is ready to be delivered.

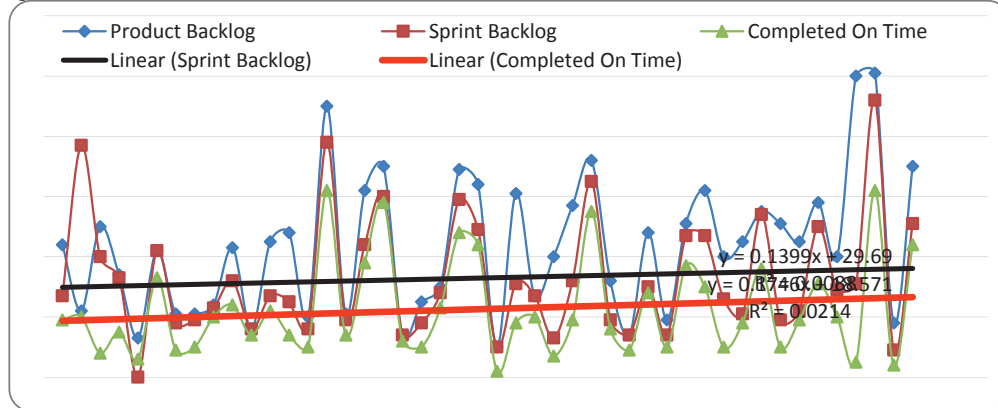


Figure 8: Issues Resolved Vs Planned

From Figure 8, we can see that there has been a constant challenge with the OpenMRS sprints to be able to meet the expected goal in time. Software estimation is generally accepted to be challenging [51], but in open-source communities this estimation becomes nearly impossible because one can never correctly estimate the commitment and time spent by community contributors. The trendline for Sprint backlog Vs Completed on Time shows a constant difference that doesn't seem to have improved over time. Since, not being able to meet the estimated goal with every recurring sprint, the product remains not ready for release most of the time after every sprint. The OpenMRS community in general acknowledges this problem, yet does not have a solution to deal with it.

A clear change since moving to OpenScrum has been increase in community activity. There is much more activity in the IRC, probably due to the daily standup meetings in which the developers communicate about their activities and blockers. There is also a marked increase in the number of comments that are made on tickets. These comments are made by developers as well as implementers. This shows that people are more actively monitoring and helping each other during the sprints compared to earlier. This is a healthy sign of increased communication in the community and the timing seems to point to the fact that change in the development methodology has resulted in this increased communication in the community.

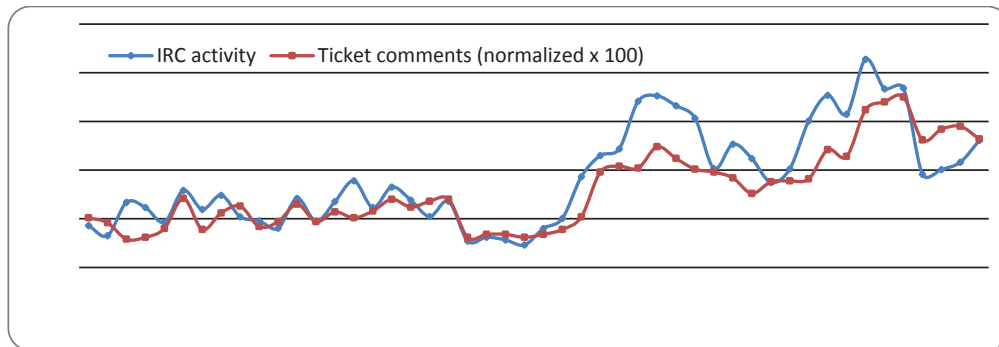


Figure 9: Community activity in IRC and ticket comments

A small note on community participation and priority was highlighted in all the interviews with the developers. While it is clear that sprinting on the most needed aspects is important for the community, it is somewhat difficult to build consensus on what is most needed. For example, requirements for those implementations often get prioritized, that have allocated developer resources. As one core developer put it, *“As we have limited developer resources, supporting implementations of our fellow developers is vital. But sometimes, the loudest voices in the implementer community only get prioritized”*. While the goal of communities should also be to listen to the shrillest voice and provide adequate assistance to them, the interviews highlighted the fact that since moving to OpenScrum, it has become harder to organize sprints that are edge-case requirements. Feature requests that could benefit the community at large or request for new modules, might not get enough votes compared to bug fixes, since large implementations have strength in numbers.

5. DISCUSSION

OpenMRS had a software development process that was used for nearly 6 years before switching to OpenScrum. While it is a continuously improving methodology, some core concepts have evolved in the last 1.5yrs. The paper refers to this tailored Scrum as OpenScrum. Pentaho, the open-source Business analytics suite has attempted a methodology by the same name, but it is not in practice. OpenScrum in essence is ideologically the same basis as the original idea behind Scrum, as coming from Rugby (scrummage). A quick restart of the game (sprint) happens after an infraction. During this restart, a front-row of highly skilled forwards, pushes with itself a team with a common goal. The OpenMRS team is led by such high-skilled core developers and push together with the community developers on a common set of activities during a sprint. Below are some of the differences between Scrum and OpenScrum as observed in practice within the OpenMRS community.

Feature	General Scrum	OpenScrum
Sprint Planning	A product backlog is created and sprint targets are created	Announcement made to the community. Community shows interest and participates in deciding the features that need to be built and completed. Design is discussed and available for sprinting
Backlog	A product backlog which is all the wish-list and stories from the stakeholders	A sprint backlog takes focus over a general wishlist. This is a list of available tasks that can be done together by the team. Publicly available for prioritization by community

		members and contains design documents for coding during the sprint.
Scrum Master	A person responsible for tracking and co-ordinating activities. Helps team to avoid distractions of changing requirements	Scrum Master needs to actively engage community members to participate. Get their views into design calls. Monitor progress of sprints and organize standup meetings. Bring community inputs into a sprint, while not considering them as distractions, but improve deliverables based on these inputs.
Product Owner	A person responsible for defining a Product Backlog and confirming after a sprint that product is ready for delivery.	Much more loose and difficult to define. The community plays the role of the Product Owner by voting on issues and prioritizing tasks.
Information Radiator	A large display board that is shared between the participants in a sprint	Similar to Scrum, but more important as it gives live updates to incoming community members to understand what tasks can be taken up by them, when joining between sprints.
Sprint Retrospective	A meeting during which features developed during sprints is demoed and lessons learnt are shared	A presentation made to the community through videos, community calls and mailing lists. Crediting participants for their activities and sharing burndown charts and timeline of changes to the information radiators
Sprints	A time-bound 4-weeks effort to create an update to a product	Smaller duration with group of developers with different backgrounds working together. Individual developers might participate in “spikes”, which are experimental, but related to ongoing sprints. Developers share experiences in the same meetings as the full sprint team.

In open-source projects, the main objective of using agile methods might not be agility. When developers coming from disparate backgrounds and interest work together, sharing knowledge becomes quintessential. In OpenMRS, certain modules were developed by individual developers and other developers did not know the inner-workings of the module. Many production environments of OpenMRS implementations used these modules and hence it was logical to make these modules bundled with the core OpenMRS distribution. Since only one developer was actively working on the module, it meant that if this developer moved on to do other things or leave the community, there would be no one to maintain the module. Even if a new developer started to maintain the module, it will take a lot of time to learn about the module after the original developer is gone. Such low bus-factor can only be increased if the community actively spends time and resources to understand how the module works, while the original developer is still with the project. Such processes of shared learning, when creating new modules results in better code review and more people are watching the code that is being written. This should generally improve the code quality, but this research has not looked into

aspects of code quality, other than just reported bugs against releases. More substantial research of the code base needs to be done to understand quality improvements due to OpenScrum. The other goal of using OpenScrum is that best practices which are learnt during a sprint have been actively created by engaging a number of developers. These shared best practices automatically get transformed into organizational practice. This means less ambiguity in the community in reaching consensus about best practices that should be followed. While leanness is a good-to-have outcome from agile processes, it should be fairly obvious that factors such as economic gains or maneuverability are less significant for communities that are working without direct economic bindings. Open-source software development has generally adopted ways of working that are simple for anyone to contribute and leave. This simplicity might be somewhat lost when a community works on building consensus and working together in sprints. Yet, as we have seen, specifically for domain-heavy projects, being able to retain individuals with knowledge is of utmost importance.

6. CONCLUSION AND FUTURE RESEARCH

We conclude by putting forth OpenScrum, a tweaked agile methodology that has empirical basis by which it has helped improve bus-factor in the OpenMRS community. This can be useful to a number of open-source projects that would like to retain developer knowledge and focus on knowledge sharing. To answer the specific research questions – agile methods like OpenScrum have improved community participation and developers know much more about each other's code-base. This in turn answers the second research question of sustainability. More developers continue to know and contribute to more modules. The contributions have widened and bus-factor for a number of modules have increased. We also conclude that Agility might not be an appropriate measure for open-source projects. Instead increasing bus-factor through knowledge sharing, increasing community participation and increasing communication are more important measures in open-source projects.

It will be interesting to use OpenScrum in domain-light (non-vertical sector) open-source projects and see the effect of adopting this methodology in those communities. It will also be useful to view code quality improvements by the use of OpenScrum in open-source projects. OpenScrum is also suited for completely community-oriented projects and might be problematic to use for projects that work in less open fashion or projects that have dual licensing workflows, where organizations need to differentiate proprietary dev team and open-source dev team.

REFERENCES

- [1] Ågerfalk PJ, Fitzgerald B, and Slaughter S. Flexible and distributed information systems development: State of the art and research challenges. *Information systems research* 20.3 (2009): 317-328. DOI: 10.1287/isre.1090.0244
- [2] Jalali S, Wohlin C. Global software engineering and agile practices: a systematic review. *Journal of Software: Evolution and Process*. 2012 Oct 1;24(6):643–59. DOI: 10.1002/smr.561
- [3] Warsta J, Abrahamsson P. Is open source software development essentially an agile method. *Proceedings of the 3rd Workshop on Open Source Software Engineering*. 2003.
- [4] Koch S. Agile principles and open source software development: A theoretical and empirical discussion. *Extreme Programming and Agile Processes in Software Engineering*. Springer Berlin Heidelberg, 2004. 85-93. DOI: 10.1007/978-3-540-24853-8_10

- [5] Beck K, Beedle M, Bennekum AV, Cockburn A, Cunningham W, Fowler M, Grenning M, et al. *Manifesto for agile software development*. (2001).
- [6] Highsmith J, Cockburn A. Agile software development: The business of innovation. *Computer* 34.9 (2001): 120-127. DOI: 10.1109/2.947100
- [7] Williams L, Cockburn A. Agile software development: it's about feedback and change, *IEEE Computer* 36 (6) (2003) 39–43. DOI: 10.1109/MC.2003.1204373
- [8] Boehm B. "Get ready for agile methods, with care." *Computer* 35.1 (2002): 64-69. DOI: 10.1109/2.976920
- [9] Nerur S, Mahapatra R, Mangalaraj G. Challenges of migrating to agile methodologies. *Communications of the ACM* 48.5 (2005): 72-78. DOI: 10.1145/1060710.1060712
- [10] Nawrocki J, Wojciechowski A. Experimental evaluation of pair programming. *European Software Control and Metrics (Escom)* (2001): 99-101.
- [11] Cao L, Mohan K, Xu P, Ramesh B. A framework for adapting agile development methodologies. *European Journal of Information Systems* 18, no. 4 (2009): 332-343. DOI:10.1057/ejis.2009.26
- [12] Mangalaraj G, Mahapatra R, and Nerur S. Acceptance of software process innovations - the case of extreme programming. *European Journal of Information Systems* 18.4 (2009): 344-354. DOI:10.1057/ejis.2009.23
- [13] Moe NB, Dingsøyr T, Dybå T. A teamwork model for understanding an agile team: A case study of a Scrum project. *Information and Software Technology* 52.5 (2010): 480-491. DOI: 10.1016/j.infsof.2009.11.004
- [14] Cockburn A. *Agile software development: the cooperative game*. Addison-Wesley Professional, 2006.
- [15] Lakhani K, Wolf R. Why Hackers Do What They Do: Understanding Motivation and Effort in Free/Open Source Software Projects (September 2003). MIT Sloan Working Paper No. 4425-03. DOI: 10.2139/ssrn.443040
- [16] Fitzgerald B. The transformation of open source software. *MIS Quarterly*. 2006;587–98.
- [17] Crowston K, Wei K, Howison J, Wiggins A. Free/Libre Open-source Software Development: What We Know and What We Do Not Know. *ACM Comput Surv.* 2008 Mar;44(2):7:1–7:35. DOI: 10.1145/2089125.2089127
- [18] Krishnamurthy S, Ou S, Tripathi AK. Acceptance of monetary rewards in open source software development. *Research Policy*. 2014 May;43(4):632–44. DOI: 10.1016/j.respol.2013.10.007
- [19] Mockus A, Fielding RT, Herbsleb JD. Two case studies of open source software development: Apache and Mozilla. *ACM Transactions on Software Engineering and Methodology (TOSEM)* 11.3 (2002): 309-346. DOI: 10.1145/567793.567795
- [20] Scacchi W. Free/open source software development. *Proceedings of the the 6th joint meeting of the European software engineering conference and the ACM SIGSOFT symposium on The foundations of software engineering*. ACM, 2007. DOI: 10.1145/1295014.1295019

- [21] Stephany F, Mens T, Gîrba T. Maispion: A tool for analysing and visualising open source software developer communities. *Proceedings of the International Workshop on Smalltalk Technologies*. ACM, 2009. DOI: 10.1145/1735935.173594
- [22] Conboy K. Agility from first principles: Reconstructing the concept of agility in information systems development. *Information Systems Research* 20.3 (2009): 329-354. DOI: 10.1287/isre.1090.0236
- [23] Jones C. *Software assessments, benchmarks, and best practices*. Addison-Wesley Longman Publishing Co., Inc., 2000.
- [24] Endres A, Rombach D. *Empirical Software and Systems Engineering: Observations, Laws, and Theories*. Addison-Wesley, 2003.
- [25] Mockus A. Succession: Measuring transfer of code and developer productivity. *Software Engineering, 2009. ICSE 2009. IEEE 31st International Conference on*. IEEE, 2009. DOI: 10.1109/ICSE.2009.5070509
- [26] Singh PV. The small-world effect: The influence of macro-level properties of developer collaboration networks on open-source project success. *ACM Transactions on Software Engineering and Methodology (TOSEM)* 20.2 (2010): 6. DOI: 10.1145/1824760.1824763
- [27] Holmström H, Fitzgerald B, Ågerfalk P, Conchúir E. Agile practices reduce distance in global software development. *Information Systems Management* 23, no. 3 (2006): 7-18. DOI: 10.1201/1078.10580530/46108.23.3.20060601/93703.2
- [28] Paasivaara M, Lassenius C. Could global software development benefit from agile methods?. *Global Software Engineering, 2006. ICGSE'06. International Conference on*. IEEE, 2006. DOI: 10.1109/ICGSE.2006.261222
- [29] Herbsleb JD. Global software engineering: The future of socio-technical coordination. *2007 Future of Software Engineering*. IEEE Computer Society, 2007. DOI: 10.1109/FOSE.2007.11
- [30] Lee G, DeLone G, Espinosa JA. Ambidextrous coping strategies in globally distributed software development projects. *Communications of the ACM* 49.10 (2006): 35-40. DOI: 10.1145/1164394.1164417
- [31] Fitzgerald B, Hartnett G, Conboy K. Customising agile methods to software practices at Intel Shannon. *European Journal of Information Systems* 15.2 (2006): 200-213. DOI: 10.1057/palgrave.ejis.3000605
- [32] Russo B, Scotto M, Sillitti A, Succi G. *Agile technologies in open source development*. Information Science Reference-Imprint of: IGI Publishing, 2009.
- [33] Marchesi M, Mannaro K, Uras S, Locci M. Distributed Scrum in research project management. In *Agile Processes in Software Engineering and Extreme Programming*, pp. 240-244. Springer Berlin Heidelberg, 2007. DOI: 10.1007/978-3-540-73101-6_45

- [34] Takeuchi H, Nonaka I. The new new product development game. *Harvard business review* 64.1 (1986): 137-146.
- [35] Nonaka I, Takeuchi H. The wise leader. *Harvard Business Review* 89.5 (2011): 58-67.
- [36] Yin RK. *Case study research: Design and methods*. Vol. 5. SAGE Publications, Incorporated, 2008.
- [37] Eisenhardt KM. Building theories from case study research. *Academy of management review* (1989): 532-550. DOI: 10.2307/258557
- [38] Kaplan B, Duchon D. Combining qualitative and quantitative methods in information systems research: a case study. *MIS Quarterly* (1988): 571-586. DOI: 10.2307/249133
- [39] Creswell JW, Clark VP. *Designing and conducting mixed methods research*. Thousand Oaks, CA: Sage Publications, 2007.
- [40] King N, Cassell C, Symon G. Using templates in the thematic analysis of texts. *Essential guide to qualitative methods in organizational research*. 2004;256–70.
- [41] Klein K., Myers M. A set of principles for conducting and evaluating interpretive field studies in information systems. *MIS Quarterly* (1999): 67-93. DOI: 10.2307/249410
- [42] Galliers RD. In search of a paradigm for information systems research. *Research methods in information systems* (1985): 281-297.
- [43] Yin RK. Research design issues in using the case study method to study management information systems. *The information systems research challenge: Qualitative research methods* 1 (1989): 1-6.
- [44] Cavaye AL. Case study research: a multi-faceted research approach for IS. *Information systems journal* 6.3 (1996): 227-242. DOI: 10.1111/j.1365-2575.1996.tb00015.x
- [45] Kruchten P. Contextualizing agile software development. *Journal of Software: Evolution and Process*. 2013 Apr 1;25(4):351–61. DOI: 10.1002/smr.572
- [46] Walsham G. Interpretive case studies in IS research: nature and method. *European Journal of information systems* 4.2 (1995): 74-81. DOI: 10.1057/ejis.1995.9
- [47] Seebregts CJ, Mamlin BW, Biondich PG, Fraser H, Wolfe BA, Jazayeri D, Allen C, et al. The OpenMRS implementers network. *International journal of medical informatics* 78, no. 11 (2009): 711-720. DOI: 10.1016/j.ijmedinf.2008.09.005
- [48] Schwaber, Ken, and Mike Beedle. *Agile software development with Scrum*. Vol. 1. Upper Saddle River: Prentice Hall, 2002.
- [49] Schwaber, Ken. *Agile project management with Scrum*. Microsoft Press, 2004.
- [50] McConnell, Steve. *Software Estimation: Demystifying the Black Art: Demystifying the Black Art*. Microsoft press, 2009.

P4: Towards a contextual insecurity framework: How contextual development leads to security problems in information systems

Saptarshi Purkayastha¹

¹ Department of Computer & Information Science, Norwegian University of Science & Technology (NTNU), Trondheim, Norway

Abstract. Most research in our field of Information Security highlights the need to consider application security during functional requirements. Yet, we see numerous examples where security of information systems is an afterthought. The paper suggests that the process of functional requirements gathering helps in inscribing “contextual insecurity” within an application. Through the case study of an open-source health management information system, which has large scale, country-wide implementations in developing countries, the paper suggests that if contexts of use are inherently insecure in nature, these insecurities become part of an application's development and use. The paper presents the process of security certification for this large-scale system, by the government's IT ministry and highlights how most improvements done to pass the certification process do not actually make it in the production environment

Keywords: contextual insecurity, health information systems, OWASP, security testing, web security, InfoSec

1. Introduction

When Castells (1996) [1] first described the networked society as a society where the key social structures and activities are organized around electronically processed information networks, he might not have considered the consequences of an organization like Wikileaks and its Hacktivism [2] would breach the circle of trust of the networks and have access to information that could change social structures forever. When Benkler (2006) [3] examines that technology enables collaboration and enables wealth to be created in these networks, he probably underemphasizes the need for closed networks and how society itself is open in one aspect, but closed through many enemies of openness [4]. Thus highlighting that, as information society progresses, we need to research in much more detail our understanding of the human values of open/closed information access and then use these values to define what secure or insecure practice in a network society is.

We see from the field of Information Security (InfoSec) research that much focus of current research is towards developing algorithms and techniques to lock information [5]. We also see that most research in InfoSec highlights that security in the design of Information Systems (IS) is generally an afterthought [6][7]. As more and more information systems are being deployed on networks, all research in InfoSec points to the need of centralizing the role of InfoSec in the design and development of IS artifacts [8].

On the other hand in IS research, the focus of systems development is geared towards matching design to the contextual use of the system. Understanding the current processes of an organization and fitting the technology artifact to the organization's current process has been prescribed time and again [9][10]. In this paper, we argue that this process of developing systems to fit the current business process, results in inscribing insecurities that may be inherent in the existing social processes. The paper suggests that if contexts of use are inherently insecure in nature, these insecurities become part of an application's development and use. Designers of the technology artifacts need to take special care to realize that network society might have different rules of information access compared to existing non-networked society from which the process is being taken. Through the example of a large-scale, country wide

implementation of a health information system and its consequent security certification by the ministry of IT, we look at how best-practice security recommendations are lost in implementation. The paper anonymizes the system which is implemented (further referred as *HMIS system*) and the country of implementation (further referred as *AFIN*) to prevent any misuse of information provided here. The paper also tries to minimize direct references in literature that point to this *HMIS system* and *AFIN* country to avoid identification. Like every security disclosure, the author wants to mention that the information provided herein is only for research purposes and any misuse of the information cannot be attributed to the author directly or indirectly.

The next section (Section 2) of the paper introduces the Health Management Information System (HMIS), its conceptualization, use and the context in which it was designed. Section 3 of the paper describes the process of security certification of the *HMIS system* and findings of the security certification process. In Section 4 of the paper there is discussion and analysis of the findings. Using the analysis, the paper co-relates the context of design to “contextual insecurity”. We discuss how most improvements from the security certification are lost in implementation. The concluding remarks are made in Section 5 and provide directions for future research.

2. The HMIS – Context, Design and Use

Health Information is generally considered to be private information in most parts of the world [11] [12]. This includes patient records, health provider records, health institution records etc. Different Access Control Lists (ACL) are required to be maintained depending on the granularity of the data and read/write permissions, but nevertheless these need to be provided and considered well in advance in the design of HMIS systems.

The *HMIS system* presented in the paper is primarily an aggregate health data repository for public health analysis and reporting of data. The basic unit of data is called data element which is generally a numeric representation of total number of individuals who have a disease or have been treated or reported. There are also logistic and financial data elements which represent stocks of drugs or any other health resource. These data elements then contribute to generation of indicators that report statistics of say, disease prevalence in a given area or helps in management of resources available in the area. Although aggregate-level data is the general use of the system, over the past couple of years, a patient records module has been incorporated into the system and the individual records from the patient records module are now used to generate aggregate data values or indicators for the system. Such use of the system as an electronic patient record is restricted to limited area of implementation, but has been widely advertised as the solution to improving data quality.

2.1 Research Methodology

The *HMIS system* has been developed through Scandinavian action research tradition in IS development, such as user participation, evolutionary approaches and prototyping [13] [14] [15]. The system is part of a global network of action researchers and aims to generate knowledge by taking part in the full cycle of design, development, implementation, use and analysis. The above mentioned steps are done together with all the involved parties before the interventions are adjusted accordingly, and the next cycle begins again [16]. The author of the paper has been involved in all the phases of research mentioned above in *AFIN*. The author has participated in action research for the last 3 years through a non-profit organization, which has been involved in the implementation of the *HMIS system* for more than a decade now. The

author has played a central role in the process of security certification of the software, done by the Ministry of IT of *AFIN*. The research has been done within the framework of interpretive research [17]. Data was collected through different channels of communication with developers of the *HMIS system* on one side and the security testing agency on the other side. The author has been involved in customization and training of the application and the interpretation involves documents from implementations, manuals and being part of meetings to customize the system in different states in *AFIN*. The research for this paper has been done by the author over a period of 2 years, where more than a year was spent in customization and development of the system and about 8-months as part of the certification process. Along with the security testing, functional testing and performance testing was also conducted for this *HMIS system*. The author was involved in these testing processes as part of a larger team. Most of the data collected from functional testing and performance testings are not part of this paper, but do provide the author with insight and help in interpretation of the observed phenomenon. A large fee was charged by the Ministry of IT to perform this testing and was an important factor for the system to be implemented on government infrastructure. The author was employed by the non-profit organization during the period of research.

2.2 Context of Design, Development and Implementation

The *HMIS system* has been designed, developed and implemented in developing countries around the world. The action research project that is at the core of this *HMIS system* is involved in research activities in developing countries in Asia and Africa. This system is thus built around the idea of supporting health systems in these developing countries. The system has been in use over 20 different countries around the world, sometimes as pilots or district-sized implementation, but also has been implemented as country-wide health management information system. In *AFIN*, it has been used by a number of states, but is not implemented as the National HMIS. Nevertheless, these state-wide implementations are online systems that can be accessed over the internet and there are separate implementations for each state. There are anywhere between 100-5000 facilities that report data into these systems and nearly thousands of users in each state using this system. Thus, in *AFIN*, the system can be called as a large-scale web application.

Before the use of electronic systems and computers, the health system in *AFIN* has primarily made use of paper forms to report data from health facilities. These paper reports are created by community health workers who operate from facilities and provide health services to the community. The community health worker maintains registers of patients and the services provided to them. These registers are classified separately, based on the type of health program or services offered by the health worker. Thus, the patient record is created by the community health worker and is available only at the facility in which the health worker has provided services to a person. This data from the registers is then aggregated by the health worker every month and reported according to a standardized facility form and its data elements. The aggregated reports are taken by the health workers and submitted to the higher level, which then aggregates all the reports received from health workers to create another form. This form is then sent higher and the higher level aggregates and sends it higher. This hierarchical chain ensures that all data from lower levels are seen by the higher levels and the higher levels can allocate resources to the lower levels.

The *HMIS system* has been designed to mimic the organizational hierarchy of health system and data can be entered by the respective facilities/organizational units by logging onto the system, where they are able to see their own unit and all the units below them. There are Access rules that can be created in the system, which allows the administrator of the system to limit the levels/datasets that is available to any user. The *HMIS system* is also designed to allow

flexible methods of aggregation when viewing data at higher levels. There are many useful analysis and reporting tools that can be used the facilities to manage their own data and analyse it for their own activities. In the patient module of the *HMIS system*, the patient record can be opened by any facility from the organizational unit hierarchy. This is because in the context of *AFIN*, migration is a common phenomenon. As an example, it is the social norm in *AFIN* that after a woman gets pregnant; she goes from her in-laws house to her parent's house for delivery of the baby. The previous treatment received by the woman needs to be available at the other facility where the woman has migrated, so that continuity of care can be provided. At the time of writing of this paper, the HMIS system can only deal with migrations that happen in the same state, i.e. if the organization unit to which the patient has migrated is in the same implementation of the system. This means that exchange of records across different installations of the system is not possible and only patient migration in the same installation is possible. This is an important distinction to understand, so that we understand that patient records are technically not exchanged between systems, but only accessed by other users from the same database.

In *AFIN*, not all the health facilities have access to internet and many do not have a computer available. At such places, the *HMIS system* is deployed at a closest facility where computer is available, but due to lack of internet, offline installations of the system are also made. These offline installations export data to the central online system via USB sticks which are imported into the online system from computers located at another village/town where internet access is available. This combination of online and offline systems is a reality in most countries where the system is implemented and is an important characteristic of the context. The offline installation is exactly the same application as the online application. Only that they are not part of the internet and are restricted in use only by the facility in which it has been installed. Thus, there is a hybrid model of deployment in offline and online modes. With respect to the patient module, in cases where there is no internet or computer available, the whole patient record on paper moves to the higher location where computer with internet access is available. This patient record may be entered into the *HMIS system* by the health worker themselves or by data entry operators who are hired at district offices because the health workers generally do not have enough computer skills. This is an important property of the context because it means that the data entry operator acts as a proxy between the health worker and the system.

The software is customized according to the requirements of each implementation. In *AFIN*, this means that every state has different data elements, organization units, reports. There are new features that are requested by these states very often and local developers are involved in customization of the *HMIS system* for the state. These local developments are then analyzed by a global team of software developers and after negotiations are made part of the central application that is available for use in other countries.

3. The Security Certification Process and Findings

The Ministry of IT in *AFIN* is responsible for managing data and server infrastructure for state governments. Thus, for the *HMIS system* to be hosted on government infrastructure it has to pass through a security certification process. The complete web application testing process involves functional testing, performance testing and security testing. This paper focuses on the security certification process, although the 3 were done in parallel by the testing agency and multiple processes were followed in sync between the three testing.

The Open Web Application Security Project (OWASP) methodology [18] was used by the testing agency for testing the *HMIS system*. OWASP is a popular open group of security analysts, who discuss and share the trends on InfoSec. Through the discussions, OWASP releases the top ten security vulnerabilities in web application and these are updated on a regular basis. Based on the top-ten list of vulnerabilities, different ways to test and fix these

issues are mentioned by the project and these have been further referred in the paper as OWASP methodology. As part of the security testing by the government agency, there were 2 attempts allowed to clear the security certification process. After each attempt of testing a report was submitted by the testing agency to the non-profit organization that is implementing the *HMIS system*. The following were the findings of the first attempt:

No	Category	Observation	Action taken
1.	Input Validation	No input validation on certain form fields	1 issue unsatisfactory
2.	Authentication & Session Mgmt.	1. SSL not used 2. Password sent as clear-text 3. No password complexity imposed 4. No password lockout	4 issues unsatisfactory
3.	Access Control	1. Session ID sniffed and XSS attack performed 2. No audit trail for critical functions and failed logins	2 issues unsatisfactory
4.	Error Handling	1. Various error messages	1 issue unsatisfactory
5.	Data Protection	1. Sensitive information can be stored in proxy 2. Clear text passwords are stored 3. TLS not used for data exchange 4. No secure key exchange 5. No secure algorithm 6. No Key length defined 7. No digital certificate	7 issues unsatisfactory
6.	CSRF Attack	1.Delete user is vulnerable including self-deletion 2.Changing password does not require old password	2 issues unsatisfactory

From the table you can see that the finding were not at all favorable after the first testing of the *HMIS system*. The system had been in use for over 4 years throughout the world with these issues, but yet they were listed out by the testing agency for the first time. A total of 17 issues out of 25 top OWASP issues were unsatisfactory. Most of the issues were dealt with by the developer team easily by just installing an SSL certificate and some quick changes to the source-code. Not that these changes were complex, but because security was never considered in the development of the system these had been ignored for over 4 years of deployment.

After the second round of testing a total of 4 issues were found by the security agency and reported the developers of the system. These issues were slightly more complex and meant creation of a “Captcha” service that would be able to prevent brute-force attacks, by showing a human-recognizable image when password were typed incorrectly for certain number of

times. Another issue was that if there are continuous mistakes in the password the user is logged out for a certain amount of time, before they can re-use the system. This was suggested by the testing agency to prevent Denial-of-Service (DoS) attacks. Other 2 issues were related to data security and were regressions (new issues appear when old issues were fixed) of the XSS flaw and required detailed debugging to solve. Nevertheless, after the second attempt, these issues were solved and re-submitted for security testing.

The testing agency performed its third and final test and did not find any remaining security vulnerabilities in the system. Although as a researcher in InfoSec, the author of the paper knows there were other vulnerabilities that were missed out by the testing agency. These are discussed in the Section 4 of this paper.

After the certification was confirmed, the author of the paper announced it to the global developers' team:

"Hurray! We get the certificate once functionality (ed: testing) is completed. They will send a template that will have instructions for (ed: testing agency) and this template will be attached with our certificate. Finally I must say this has been a loooong one but good for legitimizing the HMIS system use in government"

In response, a global core developer responded

"Fantastic news. We may be got off lightly on a few things or maybe you made some more changes I didn't see. Still looks like a good result to me."

I agree 100% about the legitimizing. I think there is a broader implication that FOSS projects can and should be able to (and be seen to) measure up to these testing regimes. So for (ed: government of AFIN) to issue such a certificate to a FOSS project like ours is really a cause for great celebration".

The quotes shown here are important to understand that the global community of software developers realized that such acknowledgement and security improvements in the system were overall a good thing. The point that such certifications help in institutionalization of the system is also understood from these conversations.

4. Discussion & Analysis – Towards contextual insecurity

The case of the *HMIS system* described earlier is probably a common one in the developing country context. The reasons for this have been discussed in the literature because security is an afterthought in system design. Also, in developing countries making the system work and then sustain is often a bigger challenge. Thus, security in these contexts is considered to be useful, but often as a lower priority.

4.1 Lack of use of Digital Certificates and SSL

Digital Certificates enable the user to identify the authenticity of the server as the certificate is given to the server by someone whom the client considers trustworthy. The Digital Certificates must hence be purchased from a source that has a seal of authenticity and users recognize that source. When web browsers see a digital certificate on the server, the communication by default is through a secure channel known as SSL. Although it is common-sense in modern web technologies that any internet-facing web application should be hosted through https (using Secure Socket Layers), the *HMIS system* all around the world is deployed over http. This means that all data sent from the client browsers to the server are sent as plain text and can be intercepted by anyone, possibly modified and such modifications might be impossible to detect. The security certification process mentioned this in 8 unsatisfactory observations, but after the certification was given, there have been numerous implementations of the *HMIS system* and yet they have all been deployed without SSL or Digital Certificate. The reason for this can be

two-fold. One explanation can be simply to avoid complexity/cost of communicating over SSL (including cost of Digital Certificate). The other more complex challenge is distribution of keys to offline installations. Ideally when using digital certificates, one would want to sign the exports coming from an offline system to be identified by the online system when importing. This process can be thought of as if a health worker and accepting officer, both sign the facility form every month and only accept the forms if both have signed and validated the data. In practice, the officer does not look through the register of patient records and hence cannot validate the data. Thus, in practice, they do not sign the forms of data exchange and the same insecurity has been inscribed into the application. It has also been observed by the author that health workers and district health officers will often deny that the data in *HMIS system* is not their data and it has been changed by someone else. If the security principle of non-repudiation had been implemented in the *HMIS system*, then these claims would not have been possible.

4.2 Lack of patient privacy

In the context of *AFIN*, migration of persons receiving treatment from health facilities is a common phenomenon. To ensure continuity of care, the system must be able to search and open patient records from any system. But this also means that health workers from any location will be able to look for any patient's record. This can be thought of as a privacy nightmare and in many cases an illegal access of records. Since in practice, a migrated patient would take their patient card and visit another facility and receive treatment, similar feature has been built into the system. The contextual insecurity in this case is to cater for the process of migration. Instead of thinking of secure ways to manage patient records to deal with migration, giving access to all health workers in a networked society is simply unacceptable. One solution that has been discussed by the author with the *AFIN*'s health agencies is the use of patient consent when exchanging medical records. When another health worker would like to view a patient's health record, the patient can either using biometric ways (e.g. fingerprint) or through a password, give access to their record to the health worker at the migrated location. This solution has been discussed and appreciated, but no implementation of this solution has been done.

4.3 Lack of Encryption

The data in the *HMIS system* are stored without any form of encryption. This means that a system administrator who has access to the database can look at the records of any person and have access to modify the records without leaving any trace in the system. Another issue that was not discovered by the security testing agency, but needs to be highlighted by the author is the lack of encryption in export files. When offline installations export data and carry these export files in USB sticks, they are represented as plain-text xml files. These XML can be changed by anyone and still will be considered to be valid by the online installation when importing. In practice, a health worker writes the patient records in the register and keeps it at the facility. This does not require the health worker to use any codes to represent information and can be written in any way which she can understand and comprehend. Even when carrying the monthly reports to the officer, she carries the aggregate form with herself and not the patient register. If the same principles from non-networked society, such as plain-text data storage and no security in the medium of exchange are put into practice in networked society, we will have contextual insecurities and chances of data theft.

4.4 Deployment problems and user competence

Although “Captcha” and user-lockout were added to the system as part of the security certification, the global team of developers removed that functionality out of the system in the later version. In practice, it has been seen that users can deliberately type incorrect password and be locked-out to avoid working on the system. Other developers have mentioned that captcha and such security features are not practical on offline installations and since we did not want to maintain two versions of the application it has been removed from the system. This is another example of contextual insecurity because as systems that are online, there are bound to be brute-force and DoS attacks on the servers.

Another common contextual insecurity is the use of passwords and creation of simple passwords. Thomson & Von Solms (1998) [19] have shown that humans who are less technically savvy tend to have common passwords across multiple applications and systems. In the *HMIS system* implementation, we see that there are often same passwords that are given to all the organization units and the only factor of difference is the username. If one knows the username of another user, since the passwords are same, the user can login as the other user and modify the system. The majority of the users of the *HMIS system* might be considered technically unsavvy users. Thus the system does not have built-in requirement for complexity of password creation like compulsory use of special characters.

4.5 Contextual Insecurity – Analysis of the concept

From the discussion we see that most issues highlighted by the security certification agency with respect to OWASP are because of the context in which the system has been in use.

Where processes of use have lack of security (insecurity), these are transferred from the context of use into the technology artifact. Inscription is a concept from Action-Network Theory which the paper uses. The term examines the way designers interpret the world around them and inscribe these into the technology artifact, as well as more implicit translations being negotiated in the context of use [20]. Thus, the insecurity within an artifact which is inscribed because of the context of use has been referred to as Contextual Insecurity.

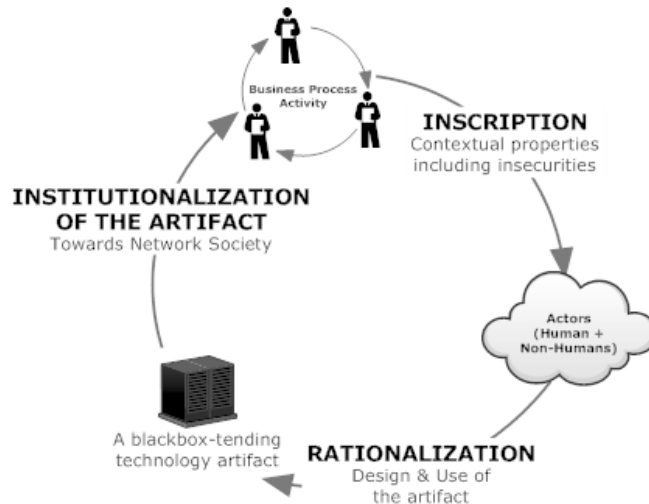


Figure 14: Contextual Insecurity Framework

Contextual insecurities are long drawn processes that have been practiced in networked society. In one of our case of exchange of forms, we see a form of social trust between the health worker and health officer to whom the forms are submitted every month. This process of form exchange in the non-networked society does not call for encryption or secure medium of

exchange. In fact, because the health worker and health officer can see each other physically, they can trust each other's existence and validity of the data becomes an abstract social binding. On the other hand, in a networked society, encryption, secure medium and validating of identity are key pillars on which it stands.

When we look at a process of business activity, we see there are numerous interactions that happen between people, technology or both. These interactions might be part of the non-networked society (where there is no electronic technology mediating) or can be part of the networked society (where exiting technologies are studied). We see that when technology artifacts are designed, many localizations are done by the actors to fit the context. The actors in our case include the government policy, the security testing agency, the OWASP framework and tools, the implementing non-profit organization and the global team of developers. Each of these actors attempts to inscribe the observed business process activity according to pre-existing rules. The government policy tried to inscribe in the HMIS system the requirement of passing security certification. The security testing agency applies the OWASP framework and uses the tools to test the artifact based on some recommended best practices. The implementing non-profit organization and the global team of developers attempt to inscribe the context's use-cases into the system, while fulfilling the needs to the other actors and creating a balance. The interaction between actors to reach a common goal is the called Rationalization where design of the system, modifications to meet security certification is part of this testing process. In other times, the trainings, customizations, implementation and use, helps in creating a technology artifact, which tends to become black-boxed with more and more rationalization. The technology artifact thus always tends towards become black-boxed and as more and more social interactions happen through this artifact, it becomes a common-place in society and then is considered to be part of every business process. Institutionalization of the artifact results in then changing the business process activity.

When looking at our case of the *HMIS system*, we see that along with the functional requirements, even current practices that did not require security considerations in non-networked society became inscribed. But as shown by other actors, some of these inscriptions are in fact insecurities. These contextual insecurities although for the sake of negotiations (getting the certification) are made to go away, since they have been in use and institutionalized in the system, they stay in the system. Thus, even if we observe that there can be problems of data tampering, the factors of inertia are lot stronger and cause people to ignore the insecurities. From our observation, we see that as time progresses these insecurities become accepted as the social norm and become insecurities of society itself. Since these are then part of the process activity itself, we then get a blurred view whether these were ever insecurities in the first place. The making and breaking of what we interpret as contextual insecurity is a continuous one. Regular cycles of inscription, rationalization and institutionalization happen and these continue to change our notions of insecure/secure practices.

5. Conclusion & Future Research

The paper argues that contextual insecurities need to be considered by designers of information systems when looking at functional requirements. The paper shows the need to realize that contexts help inscribe insecurities in the technology artifact and these are generally ignored by popular security testing frameworks. The paper also shows how rules of non-networked society cannot be directly translated into implementation in networked society. New understanding of applying security in the network must be developed and implemented by the information systems.

For good or bad, contextual insecurities are inscribed in an artifact. It is also important that security testing agencies recognize the fact that contextual insecurities are going to be part of any system. Discovering the reasons for these problems by referring to functional requirements and context of deployment should be an important part of popular frameworks such as OWASP. InfoSec has mostly focused on how to harden information in systems, but has rarely looked at the reasons why the security problems arise. Looking at it from the worldview of culture, context and social structures is a good avenue for future research.

References

1. Castells, M. *The rise of the network society*. Wiley-Blackwell (2000).
2. Ludlow, P. WikiLeaks and Hacktivist Culture. *The Nation*. 4, 25–26 (2010).
3. Benkler, Y. *The wealth of networks: How social production transforms markets and freedom*. Yale Univ Pr (2006).
4. Popper, K.R. *The open society and its enemies: Hegel and Marx*. Routledge (2003).
5. Siponen, M.T., Oinas-Kukkonen, H. A review of information security issues and respective research contributions. *ACM SIGMIS Database*. 38, 60–80 (2007).
6. Geer Jr, D., Hoo, K.S., Jaquith, A. Information security: why the future belongs to the quants. *Security & Privacy, IEEE*. 1, 24–32 (2003).
7. Mouratidis, H., Giorgini, P., Manson, G. Integrating security and systems engineering: Towards the modelling of secure information systems. *Advanced Information Systems Engineering*. p. 1031–1031 (2010).
8. Baskerville, R. Information systems security design methods: implications for information systems development. *ACM Computing Surveys (CSUR)*. 25, 375–414 (1993).
9. DeLone, W.H., McLean, E.R. Information systems success: the quest for the dependent variable. *Information systems research*. 3, 60–95 (1992).
10. Heeks, R. Information systems and developing countries: Failure, success, and local improvisations. *The Information Society*. 18, 101–112 (2002).
11. Gostin, L.O. Health information privacy. *Cornell L. Rev.* 80, 451–1756 (1995).
12. Rindfleisch, T.C. Privacy, information technology, and health care. *Communications of the ACM*. 40, 92–100 (1997).
13. Sandberg, A. Socio-technical design, trade union strategies and action research. *Research Methods in Information Systems*, North-Holland, Amsterdam. 79–92 (1985).
14. Bjerknes, G., Ehn, P., Kyng, M., Nygaard, K. *Computers and democracy: A Scandinavian challenge*. Gower Pub Co (1987).
15. Greenbaum, J.M., Kyng, M. *Design at work: Cooperative design of computer systems*. CRC (1991).
16. Susman, G.I., Evered, R.D. An assessment of the scientific merits of action research. *Administrative science quarterly*. 23, 582–603 (1978).
17. Walsham, G. Interpretive case studies in IS research: nature and method. *European Journal of information systems*. 4, 74–81 (1995).
18. Stock, A., Williams, J., Wichers, D. OWASP top 10. OWASP Foundation, July. (2007).
19. Thomson, M.E., Von Solms, R. Information security awareness: educating your users effectively. *Information Management & Computer Security*. 6, 167–173 (1998).
20. Akrich, M. The de-scription of technical objects. *Shaping technology/building society*. 205–224 (1992).

P5: Overview, not overwhelm: Framing Big Data using Organizational Capabilities

Abstract

In contexts where fragmentation of information systems is a problem, data warehouse (DW) has brought disparate sources of information together. While bringing data together from multiple health programs and patient record systems, how does one make sense of huge amounts of integrated information? Recent research and industry uses the term, "Operational BI" for decision making tools used in operational activities. In this paper, we highlight the use of DHIS 2, a large-scale, open-source, Health Management Information System (HMIS) that acts as a DW. Firstly, we present the results of a survey done in 13 countries to assess how Operational BI Tools are used. We then show 3 generations of BI Tools in DHIS 2 that have evolved from action-research done over 18 years in more than 30 countries. Secondly, we develop the Overview-Overwhelm (O-O) analytical framework for large-scale systems that need to work with Big Data. The O-O framework combines lessons from DHIS 2 BI Tools design and implementation survey results.

Keywords: Operational BI, Data warehouse, Big data, Health information systems, Developing countries, DHIS 2; e-Health; Analytics; Action-research

1. Introduction

It is evident from the literature that health information systems (HIS) in developing countries are fragmented (Littlejohns et al., 2003; Mossialos et al., 2005; Monteiro, 2003). This fragmentation causes problems of incomplete and inaccurate information (Chaudhry et al., 2006; Kimaro and Nhamposha, 2005). Yet, this fragmentation of systems is also commonly found in large enterprises, where different departments gather data in separate transactional systems that do not communicate with one another. Although our tools and embedded experiences are from the developing-country contexts, you'll find reminiscences of enterprises across the world and thus, the lessons learnt by us over the last 18 years should be valuable to many practitioners who design or implement Business Intelligence (BI) tools. At the outset, we will like to inform our reader about the practitioner focus of this paper. Thus, we have concise description of our research design and methodology. The paper focuses more on findings and historical evolution of our action-research. We also believe that it is not co-incidence that evolution of our BI tools has gone similar to what Chen et al. (2012) describe as BI&A 1.0, 2.0 and 3.0. Our field-level requirements have directed us in the same direction, where we classify our generation of BI tools in a similar fashion.

Recent research points to the use of BI for managing the daily business operations (White, 2005). This has been referred to as Operational BI as it is used to manage and optimize operational activities. Although most literature characterizes Operational BI as real-time and low-latency availability of data, there is also acknowledgment that "*Operational BI puts reporting and analytics application into the hands of users who can leverage information for their own operational activities*" (Keny and Chemburkar, 2006). Others have referred to this as "real-time", but we prefer to use the term "right-time" and highlight operational availability to be the main difference, when referring to Operational BI in comparison to Real-time BI. Similarly, "use of information for local action" has been advocated by numerous researchers working with health information systems (Orenstein and Bernier, 1990). Through this paper's discussion section, we will see that it is not a co-incidence that public health researchers

and operational science researchers have both pointed to the same need, but rarely shared the knowledge between the two domains. We look at a popular and arguably the largest (Webster, 2011) open-source Health Management Information System called DHIS 2 (its full-form: “District Health Information Software version 2” suggesting a health-only domain is irrelevant and is more of a backronym these days) and how its Operational BI tools are used in developing countries to manage country-wide or state-wide health systems.

The paper is organized as follows. In the following section we present the current conceptualization of Big Data and Operational BI. Here we also add the literature on Organizational Capabilities that is central to our survey of health data managers and implementers of DHIS 2. We briefly mention our research approach in Section 3. The findings of the survey are presented and discussed in Section 4 of the paper. In Section 5, we highlight the Operational BI Tools in DHIS2 and how these tools have evolved over time. Here we present a specific experience of integrating mobile data capture and the challenge posed by visibility of work due to data. In Section 6, we highlight the lessons learnt in the design of BI and Analytics (BI&A) tools for Big Data. In Section 7, we present the Overview-Overwhelm (O-O) analytical framework for understanding and managing Big Data and articulating the “bigness” of data through organizational capability perspective. We conclude the paper and highlight future work in the last section.

2. Conceptual Framing

2.1. Conceptualization of Big Data

Big Data is a term widely used in popular media. Words like “Petabyte Age”, “Industrial revolution of data” are common, yet haven’t we had huge datasets of petabytes running on super computers for bioinformatics, space research or other high-performance computing domains for at least the last 20 years? What makes the current times unique is that never before have the “masses” been involved in data creation exercise at this scale, nor has so much general computing power become available. Academic definition for the term is particularly hard to find, but industry reports have defined Big Data with wordings such as:

“Big Data is data that exceeds processing capacity of conventional database systems” (O’Reilly Media)

“Any amount of data that’s too big to be handled by one computer” (Amazon)

“Big Data is data with attributes of high volume, high velocity, high variety and low veracity” (IBM)

“Big Data refers to datasets whose size is beyond the ability of typical database software tools to capture, store, manage and analyze” (McKinsey)

With the integration of multiple sources of data into a DW, we face the challenge of increased Volumes, increased Velocity and increased Variety of data. This has been commonly referred to as the “Three V” of Big Data (Laney, 2001). Other researchers have also added the challenge of Veracity of data as the fourth Big Data challenge (Zikapoulos, 2013). Weiss and Indurkha (1997) in computer science and Diebold (2003) in econometrics were one of the first to suggest practical principles of data mining for Big Data. Still, it is difficult to apply these principles in the context of low-resource settings, where the constraints in themselves add to the “bigness” of incoming data. The nature of data in health systems is obvious to the “Four Vs”, because with population increase, the medical observations, health indicators and health facilities are always on the increase. When any current incoming data point has to be analyzed with the vast snapshot of existing historical data, the exponential increase in complexity poses problems of processing, storage and analytics.

From these reports and related literature, we see that Big Data is not only large volumes of data, but also contains complex interconnections (Snijders et al., 2012). So the term in itself is somewhat poorly descriptive (Manovich, 2011). Should it be called large, quickly changing, complex datasets for people who dislike or find incorrect use of the term “*Big Data*”? We do not attempt to demystify Big Data here. We simply claim that national health DW contains data of similar attributes that needs to become meaningful information, through analytics so that it can improve health services and save lives.

2.2. *Conceptualization of Operational BI*

Over the last 50 years, BI has been expected to change the way in which information is used by organizations to make business decisions. These have been over the years re-invented as Decision Support Systems (DSS), Expert Systems and Executive Information Systems (EIS) (O’Brien, 1991). Much effort has been tied to automation or support of human decision making. Still, systems that provide process-based information for decision making have only become top priority for Chief Information Officers (CIOs) in the last 10 years (Watson and Wixom, 2007). Even the creation of job profile for CIO started in the early 90s with the implementation of ERP systems and ensuring the potentials of these systems are met (Earl, 1996). This long hiatus can be attributed to the complexity of business processes and lack of availability of data from all parts of the organization that can enable making a holistic decision. BI provides insights to managers and information officers to make more informed decisions (Lawton, 2006).

White (2005) classifies BI into 3 main parts, namely strategic, tactical and operational. These are mainly classified based on Business Focus, Primary users, Time-frame and metrics of data. The basic premise in classifying the different forms of BI comes from the level at which information is used and when the information is used. In the health sector, information needs to be made available to people at all levels. This information is required for operational activities such as what diseases are more prevalent, where additional drugs and workforce is required, how to make them available given the available resources. As mentioned earlier, this has been referred to as “use of health information for local action” (Stoops et al., 2003), where local practitioners can make use of information and adjust their work practices. With this conceptualization we see that Operational BI is used for the day-to-day activities of the users, who are information generators as well as information consumers at the same time.

Keny and Chemburkar (2006) provide a slightly different conceptualization of Operational BI. They present the idea of granularity of information as the characteristic separating Operational BI from traditional BI. They suggest that while traditional BI relies on Key Performance Indicators (KPIs) to derive a holistic perspective on corporate performance, operational BI provides much more granularity to address the needs of operational functions. This characteristic of Operational BI is similar to the concept of “hierarchy of standards” (Braa et al., 2007), where it is advocated that each level of the health system should be able to manage their own set of indicators, with increasing granularity as we go to the lower levels. The higher levels only need the aggregate view of indicators from the lower levels. Traditional BI has been complex to implement because it tries to capture all processes and business complexities. Thus, traditional BI implementations involve tricky data modelling and require highly experienced BI developers and data modellers. On the other hand, Kobiels (2009) suggests that Operational BI tools can be used as “Do-It-Yourself” business intelligence, where users themselves can configure analytical instruments like indicators, metadata, data sources and create “mashups” (a view of data from different sources) using graphical tools. In IS research, it has been identified that mashups will become the basis for Web 3.0, where user-driven programming of the web (Wong and Hong, 2007) happens, as users themselves connect the different pages of information on the internet and create new

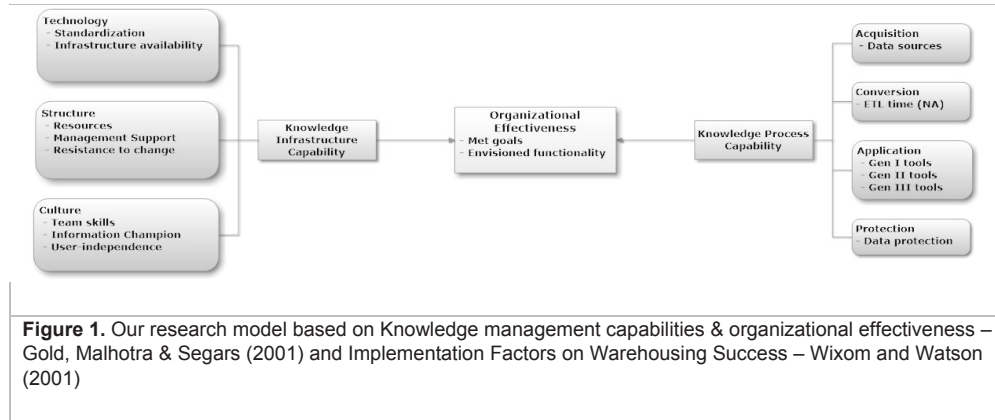
information. This common conceptualization of Operational BI and Web 3.0 through similar approaches is interesting for future research and direction of Operational BI.

2.3. *Organizational Capabilities*

Most industry definition for Big Data deals with only technical artefacts such as database systems, but does not highlight the role of organizational capabilities in determining “bigness” of data. IS Researchers have discussed about Organizational Capabilities (Gold et al., 2001) perspective in Knowledge Management that we have found useful to understand how an organization looks at Big Data. This perspective is useful to define Big Data for an organization and manage Big Data to the best of available capabilities and how these capabilities can be put to best use.

A purely technical definition of Big Data talks about single computer or standard database systems that cannot meet computational demands of analytics. We argue that this term has to be more than technical alone and should encompass organization capabilities. Volume, Velocity, Variety and Veracity of data are relative terms. Just as the superlative network speeds of the 1990s are unacceptable today, we cannot measure computational capabilities available today by the standards of the fastest supercomputer in the world. Not everywhere is such computational power accessible. Hence, the term Big Data should be contextual and be defined by the socio-technical capabilities or Organizational Capabilities.

Our research model stems from the seminal study by Wixom and Watson (2001) that studied implementation success of DW. Instead of implementation success, we postulate a model of comparing implementation factors (further classified as organizational capabilities) and the use of Operational BI tools – both of which help model the organizational effectiveness in terms of delivery of planned vision and effective use of data at different levels in the organization.



In Figure 1, our research model is shown with Knowledge Infrastructure Capabilities being the formative factors on the left and Knowledge Process Capabilities being the reflexive factors on the right. Although, we classify these as factors at the start of the implementation, we realize that during the implementation process, these formative capabilities continue to evolve due to the reflexive factors. Other researchers have described this transition has been highlighted as Architectural Maturity (Chung et al., 2005) and we’ve seen similar changes during the adoption of DHIS2 systems, where technology, structure and culture evolve during the different stages of implementation. Under Technology capabilities, we list Standardization (metadata, user access, vocabulary) and Infrastructure availability. By infrastructure, we mean hardware (servers, networks, power supply) and software (OS, database systems) infrastructure that can allow installation, running and maintenance of the DW. Under Structure

capabilities, we list Resources (amount of finance available, no. of people available), Management Support (memos, work orders) and Resistance to Change. Under Culture Capabilities, we list Team skills (competence), Information Champion (role) and User-independence (self-learning, training, decision making powers). These are also formative factors in defining Big Data for the organization. In Knowledge Process capabilities, we list Data sources (types, velocity), Application (BI Tools used – we describe this in detail in Section 5) and Protection (security, data integrity).

3. Research Approach

Our research has been conducted as part of a long term engagement in the field, doing actions and understanding the effects of our action. This empirical investigation can be broadly placed in Scandinavian Action Research tradition (Engelstad and Gustavsen, 1993), which elsewhere has also been described as *Networks of Action* (Braa et al., 2004). The *Networks of Action* approach is specifically designed for the resource limited conditions in the Global South. Our network, called HISP comprises of researchers, developers, implementers, representatives from ministry of health, all of whom share learnings between the nodes of the network. The 18 years time-span of the research project exceeds traditional ‘projects’ and is more akin to social movements (Elden and Chrisholm, 1993). One of the authors is the originator of the project in the mid-1990’s and has participated in the design, development and implementation of HIS in many countries in Africa and Asia. The other author has been part of this network for the last 4 years, primarily engaged in Asia and is part of the core developer team for the DHIS2 software.

The survey respondents were implementers, who we define as “developers/consultants/customizers or local administrators; those who are involved in setting up the system”. This includes international HIS researchers, public health consultants, in-country health bureaucrats. For the survey we defined Users as “end-users, people who enter data, look at reports or use reporting tools and perform day-to-day operations; those who are using the system”. This survey did not include Users as respondents. We use PLS method to derive the associations between the different reflexive and formative organizational capabilities. The survey was modelled on (Wixom and Watson, 2001), with a 5-point likert scale and similar technique of analysis.

4. Survey results and discussion

From our survey findings, we can see that some organizational capabilities have more direct co-relation with others, while others that we thought also had co-relation are not supported by the findings. We see that although most of the organizational capabilities are co-related to the use of Operational BI Tools, we see some factors not co-related, while others are more significant. We see from our study that structural and cultural organizational capabilities are more relevant in use of Operational BI tools than Technology factors.

TABLE 1.

Technology factors - Even though more infrastructural technology might be available, it does not necessarily mean that users would make use of Operational BI tools. This is an important learning because we often understand that since we have the technology and available data; it will not necessarily mean that these will be useful in meeting the organizational implementation success or envisioned functionality of the data warehouse. This seems to be another observation in the “Build It and They will Come” debate around Information Systems (Markus & Keil, 1994; Hatakka, 2009; Silvestre, 2009). When there has been higher standardization effort, i.e. harmonization of datasets, formats we see that

there is more likelihood that users will make use of the data. This was our hypothesis and is supported by the survey results.

Structural factors - We see resources within the organization as part of the structure. When more resources, better management support and lower resistance to change within the organization, we see better use of Operational BI tools and in turn better implementation and use of data.

Cultural factors - Team skills include implementer's inter-personal skills as well as the user skills in using the tools. Our hypothesis was that with higher skills there would be higher use of Operational BI tools. We also believed that with more Information Champions at all the levels of the organization, there would be better use of BI. Information Champions are people who are proactive in collection, reporting and use of data. Our results that support the view that more Information Champions mean better use of tools, is in some sense contrary to the findings from Wixom & Watson (2001). We find that this might be because there are other cultural and structural factors, similar to Beath (1991) that are in support in the organizations that have been part of our survey.

Data Acquisition - To our surprise, data when coming from more disparate sources does not necessarily mean that users would want to co-relate them and view them. Thus, although the nature of Big Data might be co-relation between the data points, users will not want to make use of the co-relations just by default. We also thought that there may be a reverse co-relation (lower disparate data sources, meant use of BI tools) instead, but even that is not supported by our analysis. So, it may be concluded that even if data came from one department or from many departments (or health programs in our case), use of data is not affected by it.

Data Protection - We also found that data protection has no direct impact at least from our survey respondents on the use of BI tools or the organizational effectiveness in dealing with data. This may be so, because in the developing country context, there is little or no a law or structures around electronic data protection. Some countries in Africa and Asia are moving towards such legislations, but these are far from implementation.

From the survey of the countries, we see that Organizational Capabilities are vital in understanding use of Operational BI and in turn manage Big Data. Along with this understanding, in the next section we describe the evolution of the Operational BI tools in DHIS 2, that are inscribed (Hanseth and Monteiro, 1997) with the lessons that we have learnt in managing data for health systems.

5. Operational BI Tools in DHIS 2

The DHIS 2 is a tool for collection, validation, analysis, and presentation of aggregate statistical data, tailored (but not limited) to integrated health information management activities. It is a generic tool rather than a pre-configured database application, with an open meta-data model and a flexible user interface that allows the user to design the contents of a specific information system without the need for programming.

DHIS 2 has been implemented in more than 30 countries in Africa, Asia, Latin America and the South Pacific, and countries that have adopted DHIS 2 as their nation-wide HIS software include Kenya, Tanzania, Uganda, Rwanda, Ghana, Liberia, and Bangladesh. The software is developed through open-source collaboration and has been iteratively developed through application in "real world" through bottom-up, participatory software development. The software has its roots in Scandinavian Action Research tradition in IS development where user participation, evolutionary approaches, and

prototyping are emphasized (Greenbaum and Kyng, 1991). Its development started with the reforms in health sector in post-apartheid South Africa and has evolved into a large-scale web system that is now used for country-wide management of health information systems. Over this period, lot of BI tools have become part of the application, but its core agenda has been the use of information for local action through the use of flexible standards (Braa et al., 2007).

5.1. *The first generation of Operational BI Tools*

From Offline Pivot tables to Downloadable Pivots to Stuck with Big Data

The first generation of Operational BI tools in DHIS 2 included tools which allowed users to modify data attributes and observe their effects on indicators. These tools allowed data managers to introspect on the effect of changes to certain data points with respect to a set of existing longitudinal data points. In Microsoft Excel™ spreadsheet, such a tool is Pivot tables which allow creating unweighted cross tabulations. Pivot tables are commonly used for row-column interpolation and interchanging rows with columns and vice versa. Pivot tables have been used widely in understanding and analyzing information (Jelen and Alexander, 2005). These have been important tools for integrated decision support systems based on data warehouses (March and Hevnr, 2007). From early 1996 to early 2006, these tools have been used widely by users of DHIS v1. The pivot tables perfectly supported operational activities of health staff at facilities. As other researchers have reported, some effort was required to train users to work with pivot tables, but later it became one of the most used features in DHIS (Williamson et al., 2001; Lungo, 2008; Sheikh and Bakar, 2012). New incoming data points are fairly easy to add to existing pivot tables, but only till such data is limited in size. When we moved to DHIS 2, we decided to completely rewrite it as a web application. In such networked data collection and web application world, when we first tried to implement pivot tables, it proved to be extremely slow. Web browsers were not able to support pivoting even large, static dataset. High velocity data was impossible to accommodate through pivot tables. In 2011, Google provide a plugin for web-based pivot table interface through their Google Docs™ service. But the plugin has since been deprecated because large datasets are impossible to work with. To deal with the issue of large datasets, we decided to partition the data mart based on the facilities, user data and geographical locations. We also allowed users to select indicator dimensions, period, and location, so that pivoted outputs are manageable in size. In addition, we allowed users to connect to a web service and download data mart locally using a Java desktop *mydatamart* application. This desktop-based, downloadable pivot solution was better than the browser-based pivoting solution, but still could not handle 100s of Megabytes, nor high-velocity data as the offline data mart would become stale in a matter of minutes. So, although *mydatamart* is a recipe for flexible Operational BI and used in some implementations of DHIS2, yet it cannot deal with Big Data.

From Data mart Service to Nightly data marts to Caching proxies

The Data mart service is Extract-Transform-Load (ETL) process in DHIS 2. This backend process provides information for reports based on aggregations and disaggregations of transactional data. The reports are generated from the data mart, a common practice in DW tools. In the early days of moving to a web application, DHIS 2 would run the data mart service as soon as new reports were requested. This caused the same information to appear in the reports, even if data values had changed due to high velocity, while the latest data mart generation was not completed. This is a common problem with high velocity data. Over time, there were a few optimizations like having a separate reporting server, algorithm optimizations etc. Yet, we observed that data mart creation became more and more time consuming as data became high-velocity and high-variety. To be able to deal with this challenge of growing computational capacity, we decided to build a feature by which the data mart service would only be executed every night at a scheduled time. This meant that users would be able to store data, but

the analytics could be run only on the previous day's data. This nightly data mart reduces the load on the infrastructure and makes the reports accessible to users, but reduces the operational nature of information. For e.g. in Ghana where 10 yr legacy data was imported, there were close to 33 million data points that needed rebuilding by the data mart service every night. This process is extremely technology intensive and even though using cloud hosted servers have alleviated some computational challenges, it still takes about 6 hours every night for completion. As more and more data is collected over the next few years, we are expecting even with “elastic computing”, which provides on-demand computational resources, we will need more time (more than a night) to complete the data mart. To deal with this problem of correlating large amounts of data and still make information operationally available, one suggested technique which we have started to apply is using caching proxies. These are servers which will partition the requested data and remember it for a user/group of users. So users within a district or county will always request data from their proxies and these proxies will only deal with information that is generated, stored and processed for their geographies. These proxies refresh themselves based on their own caching rules that are written on an organizational unit's operational needs. This partitioning helps reduce the load and time for the data mart process. This feature is experimental and we are evolving new paradigms for partitioning based on operational needs.

5.2. The second generation of Operational BI Tools

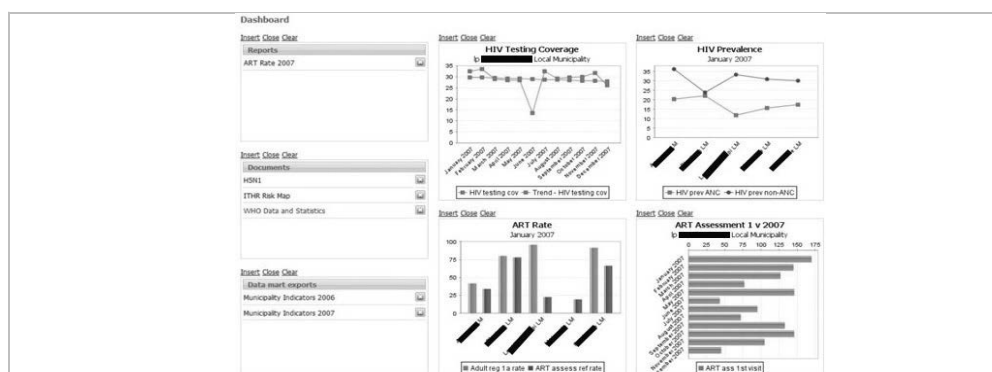


Figure 2. Dashboards with Live Views

From cross-tabulations to Dashboards

From the early days of DHIS 2, graphs and charts have been useful visualization tools. The challenge with multi-dimensional data that has n-number of co-relations is finding a useful visualization for it. Health data is considered in essence to be longitudinal in nature and hence plotting values over time has been the standard approach to visualize it. Still, when co-relating data points, tables have often played a more significant role in such analysis. As the DHIS 2 community of users moved away from pivoting, it was realized that visualizing data through charts and placing them next to each other made it possible to do multi-dimensional visualizing of data. Thus, the DHIS 2 dashboards were born, where co-related data points would be placed next to each other, even overlapping if one wishes to and view data points longitudinally. This allowed users to create their own operational “mashups” and view co-related multi-dimensional data beside one-another. This has been extremely powerful for analytics even in low-level facilities, as it provided easy overview of status of operations. Each user has their own dashboard according to their operational needs. The dashboard is live view of maps, charts and tables. It uses data from the previously mentioned data mart service.

Data mashups on Maps

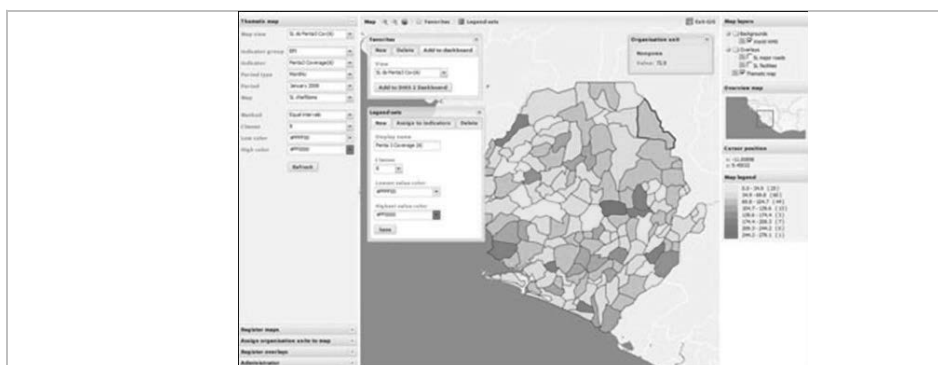


Figure 3. Data mashups on Maps

While the Dashboard provides useful overview, often the contextualization of permanent or semi-permanent data like population estimates, staffing, equipment, survey, geographical topography etc. can better be visualized on a map. DHIS 2 learnt that as users wanted to co-relate live incoming data with permanent or semi-permanent data, geographical boundaries of facilities, districts, provinces etc. provided interesting ways in which users can visualize. Users create indicators from data points and use the values from different regions to plot legends onto a map. These are especially useful to compare geographically close locations for similar kinds of information. These maps allow contextualizing the incoming data with existing data and allow users to create their own “mashups” of data. Although this has been referred to a Geographic Information System (GIS) by other researchers (Braa et al., 2007), it is more along the lines of mash-ups ideology, where maps are used as a surface to represent data extracted from the system. The tool does not allow complex layering, manipulating geo-spatial data or CAD/CAM features that are expected of GIS systems. Instead the tool in DHIS2 allows thematic mapping, while letting the user define custom legend sets and save favourite map views. The tool also allows users to drill-down to the internal organization boundaries and look at the data represented by them. Source of data (organization unit) is automatically matched with the labels on the map/shape files. Users can also manually link data to geographic co-ordinates or polygons. Users are given the flexibility to represent any information from datasets onto any map. The indicators are then seen as different colour representations based on the range of data values.

5.3. *The third generation of Operational BI Tools*

Validation Rules

Validation rules are expressions to evaluate incoming data values with existing data values. These expressions are mathematical models that can be created by users based on existing data points. Validation rules are used to match data elements and find those that are different from the normal values. These include min/max ranges as well as comparison with the values of other data elements. Failing validation rules from a business logic perspective are outliers, but from our experience in the use of mathematical models, these outliers should not be considered as mistakes in the data, but instead should mean we need to question the reasons for the outlier. New insights can often be drawn from these outliers and hence DHIS 2 allows users to add unstructured comments that can be mined for further understanding the reasons for such outliers. The process of executing validation rules cannot be thought of as a simple data cleaning (in DW often part of ETL process) because it involves analysis of what data exists in the system and what is not being entered into the system. An important part of data that violates the validation rules is to look at the comments on such instances. DHIS 2 allows for adding comments, when data is beyond the min/max values or has problems with validation. For e.g. Validation

rules are often created to verify outbreak alerts. Here, standard deviation is used to verify if the number of patients diagnosed with a disease is within the previous min-max range. Similarly, stock outs or buffer levels for inventory is managed through validation rules. So, we see that validation rules help in the local action because operational activities are better managed through the use of such rules. We also have a better understanding of data through these validation rules and comments entered on data that is failing validation rules.

Social network of Interpretations

While we have evolved through the different generations of Operational BI tools, we realize the strength of data analytics lies in the hands of users of these tools, rather than in the tools by themselves. Hence we have recently allowed users to share their interpretation of data, using all of the above mentioned tools. Users can annotate charts, graphs, tables and maps while sharing it with a group of users in DHIS 2. The sharing of these interpretations creates a sort of social network of health system analysts, where users comment and annotate each other's interpretations of data. We find that in many places newer insights have been created by leveraging the interpretations of different skilled analysts in this process. The interpretations are particularly useful when health staff make operational decisions based on interpretations of health staff located in another facility, which would otherwise be disconnected from each other's activities. These social networks allow making sense of the Big Data, but in fact also create more data for each other. Yet, this much more nuanced and extracted information is of greater value to the users, who feel excited and empowered to share their interpretations with peers and superiors. There are structural, cultural challenges that also arise in such social networks, but interpretation tools have indeed provided better extraction of information from large amounts of data. The interpretation tools in DHIS 2 are fairly new and only few countries are widely using it at the moment.

5.4. *Case of Integrated mobile phone based data capture*

HISP India is involved in the implementation of a large-scale mobile-based health information system in a northern state in India (Braa and Sahay, 2012). As part of the implementation, more than 5000 health workers have been given mobile phones that are used for reporting aggregate of health activities performed by field-level health workers. The health workers use a mobile application that captures data and sends this information as an SMS to the state health office. At about the same timeframe, clinicians working in outpatient ward of government health facilities were asked to report their activities on a daily basis through the use of similar SMS-based services. The clinicians started sending daily updates to the state office and a lot of data was being captured. On the other hand, since health workers were being given mobile phones, the state added 10 new data points on which health workers were also asked to report daily to the state. The primary aim for doing this from the state's perspective was to try and strengthen control over the health worker's activities; to know what the health worker was doing on a daily basis. Since, clinicians were already doing this, the state health department officials assumed that health workers would have no problems doing the same. However, as the implementation started, it was observed that health workers were resistant to daily reporting and voiced protests as part of the health worker union. On exploring the reasons for the resistance, it was found that the health workers only performed some of the duties on a daily basis on which they were asked to report. On the other hand, other activities that they performed on a daily basis were not being captured. Beyond that, they had fears that if they reported 'zeros', authorities would assume that they have not worked for the day, whereas in fact they provided other services. In protest to the daily reporting on new datasets, the health workers completely stopped using the mobile-based reporting system. Since the effectiveness of the full project was getting jeopardized, the state agreed that the health workers would rather report weekly on the same activities and the clinicians would continue to report their daily activities. This seemed to

go well with all the parties involved and helped collect ‘adequate’ data about the workings of the health system.

6. Lessons learnt - How to design BI Tools for Big Data

Our experience is that organizational capabilities are most important when designing and implementing BI tools. Using Operational BI tools simplifies management of large datasets and users grow into data use with such tools. Since, our lessons come from experience in participating in a long drawn action-research project involving many different health systems/organizations, the evolution of the tools embed in them the lessons that we have learnt. We summarize some of the wisdom that has been embedded in the tools. This can be used by designers or implementers of other BI tools.

Lessons #1 – Organizational capabilities are difficult to change, evolve processes of data acquisition and analytics

Organizational capabilities have inertia that makes it hard to change. Since people have become accustomed to certain tools or have been trained to use a certain type of BI&A tool, it is extremely hard to get them to change. The Pivot tables was known to be a dead-end for large datasets, yet for users to be able to continue with their existing practices, we had to develop a transitional tool that allowed similar analysis. Similarly, we also see that certain users or organizational departments have capabilities that would allow them to generate data, while other users are not likely to be able to generate the same amounts of data. Organizational culture as seen from the case in India (see Section 5.4) plays a major role in response to acquisition of Big Data, not just processing or analyzing data. Particular care needs to be taken in cases which use ubiquitous technologies like mobile phones that technologically suited to generate Big Data

Lesson #2 – Data partitioning based on operational needs is the key to scalability

Partitioning data has been a common practice for managing data. But when it comes to Big Data, it becomes an essential task. Partitioning can be based on one or combination of the following constructs:

- Time - as we did using nightly access to data
- Space - as we did on geographical limiting
- Activities - as we did using the hierarchy of standards notion

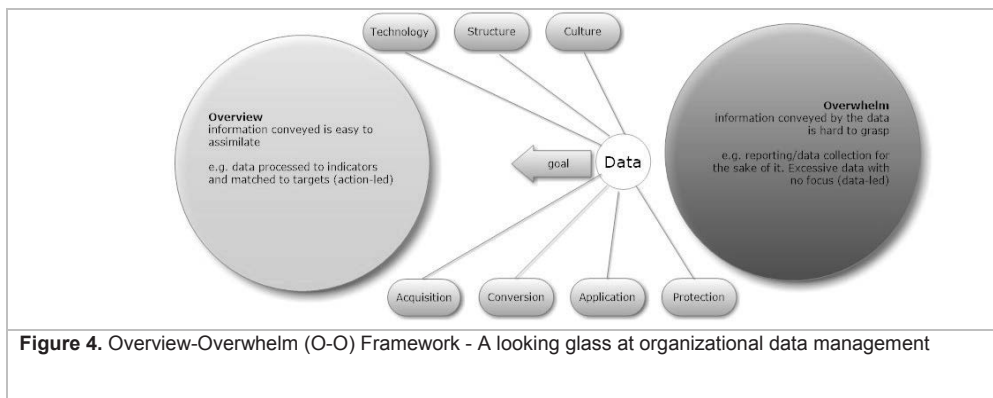
Lesson #3 – Outliers are not unclean data, but points of innovation

Outliers that have been created by a model are often considered to be unclean data or wrong data. In Big Data world, annotating such data and trying to find co-relations is very useful practice. It evolves the use of data, but also provides new insights into organizational practices. As we saw from sharing of interpretations and discussing finding in an enterprise social network creates more useful analysis within the organization about these outliers or different interpretations of data. Being able to extract further information from these discussions is expected to provide better insights into improving practices.

7. Towards Overview-Overwhelm Framework - Discussion

The conceptual view of transforming Big Data to meaningful and actionable information can be understood through the lens of the Overview-Overwhelm (O-O) analytical framework. Our starting point is the postulation that Organizational capabilities are constraining and enabling factors towards use of Big Data. We define the Overview space as optimal understanding of data, such that

organizations can easily assimilate the information that is conveyed by such data. On the opposite end, we define the Overwhelm space, where there is wasteful use of resources to make sense of data or the information conveyed by the data is hard to grasp. In this regard, we refer to the work of Sandiford et. al. (1992), which distinguishes approaches of IS in health systems, where one uses either an “action-led” approach or a “data-led” approach. When we bring Big Data into the picture, we see data-led approach resulting in Overwhelm space because often in countries, health reporting is done for the sake of reporting (Kanjor, 2011). This creates large amounts of data that is rarely used (AbouZahr and Boerma, 2005). Here we see upcoming technologies of mobile phone, ubiquitous sensors and wireless technologies contribute to generating Big Data. On the other hand, the Overview space can be created and maintained within health systems through an action-led approach. An action-led approach to Big Data is to process raw data into indicators; but go beyond creating indicators and further matching them with targets. Millennium Development Goals are such targets that countries implementing DHIS2 want to achieve. The Operational BI Tools in DHIS2 helps achieve this transition of data-led to action-led efforts or from Overwhelm to Overview information space. Organizations have large spaces in terms of information view in between the areas of overview and overwhelm. This space is often filled with data and no information or there is no scalable way in which data can be moved from Overwhelm areas to Overview areas. CIOs are individuals who need to take initiative to allow users to move data across this space. CIOs will find it hard to make this move for the whole organization. But with an army of users with operational needs for information, data can be moved as we have shown from our cases, from overwhelm to overview space. By the very nature of health data acquisition, we see that data has the inertia to move towards the Overwhelm space. We claim that Technology, Structure and Culture are formative factors towards changing this inertia and move information into the Overview space.



The O-O framework can be understood as a looking glass to view Organizational capabilities. O-O helps define what Big Data is and how it can be managed. Each of the formative factors of Technology, Structure and Culture of the organization determine whether the acquired data is Big Data or not. Depending whether the data can be provided as an overview or whether it overwhelms the organization determines if it is Big enough or Not for the organization. Starting with the DHIS 2 first generation of BI tools, they moved from an Overview boundary to Overwhelm boundary and efforts were made to reverse that direction. But, the data became overwhelming for the technology, organizational structure and culture as time went on. That strand of BI tools did not adequately move back to Overview and hence is not a suitable toolset to manage Big Data. The second generation of BI tools through the use of visualizations provides overview of data. Still there are limitations to visualization and it does not completely cover the sharing of gained knowledge through the organization. The use of social networks through interpretation of data is much wider overview of the data.

8. Conclusion

We conclude the paper by realizing that Big Data should not be defined only through technical factors. It should also be described by organizational factors such as Structure and Culture. We highlight some of the lessons that have been learnt in design, development and implementation of 3 generations of Operational BI Tools in DHIS2. We specifically show that instead of Analytics in the broad sense, Operational BI tools will help transform organizations to be able to work with Big Data. In healthcare, our experience has been to move from data-led to action-led approach using Operational BI Tools. The O-O framework can help act as a looking glass through which we can define and manage Big Data in healthcare

Our further research points to how cloud computing can aid in sharing computing resources for processing Big Data. We have been working with countries to use IaaS, PaaS, and SaaS models for their data warehouses. It will be interesting to see what are the opportunities and challenges for countries to use Cloud computing for health data analytics.

References

- AbouZahr, C. and Boerma, T. (2005), "Health information systems: the foundations of public health.", *Bulletin of the World Health Organization*, Vol. 83 No. 8, pp. 578–583.
- Beath, C.M. (1991), "Supporting the Information Technology Champion", *MIS Quarterly*, Vol. 15 No. 3, pp. 355–372.
- Braa, J., Hanseth, O., Heywood, A., Mohammed, W. and Shaw, V. (2007), "Developing health information systems in developing countries: the flexible standards strategy", *MIS Quarterly*, Vol. 31 No. 2, pp. 381–402.
- Braa, J., Monteiro, E. and Sahay, S. (2004), "Networks of Action: Sustainable Health Information Systems across Developing Countries", *MIS Quarterly*, Vol. 28 No. 3, pp. 337–362.
- Braa, J. and Sahay, S. (2012), *Integrated Health Information Architecture: Power to the Users : Design, Development, and Use*, Matrix Publishers.
- Chaudhry, B., Wang, J., Wu, S., Maglione, M., Mojica, W., Roth, E., Morton, S.C., et al. (2006), "Systematic Review: Impact of Health Information Technology on Quality, Efficiency, and Costs of Medical Care", *Annals of Internal Medicine*, Vol. 144 No. 10, pp. 742–752.
- Chen, H., Chiang, R.H.L. and Storey, V.C. (2012), "Business intelligence and analytics: from big data to big impact", *MIS Quarterly*, Vol. 36 No. 4, pp. 1165–1188.
- Chung, R., Marchand, D. and Kettinger, W. (2005), "The CEMEX Way: The Right Balance Between Local Business Flexibility and Global Standardization", *Case Study No.*, Vol. 1501, pp. 3–1341.
- Diebold, F.X. (2003), "'Big Data' Dynamic factor models for macroeconomic measurement and forecasting", *Advances in Economics and Econometrics: Theory and Applications, Eighth World Congress of the Econometric Society*, (edited by M. Dewatripont, LP Hansen and S. Turnovsky), pp. 115–122.
- Earl, M.J. (1996), "The chief information officer: Past, present, and future", *Information Management: The Organizational Dimension*, pp. 456–484.
- Elden, M. and Chisholm, R.F. (1993), "Emerging Varieties of Action Research: Introduction to the Special Issue", *Human Relations*, Vol. 46 No. 2, pp. 121–142.
- Engelstad, P.H. and Gustavsen, B. (1993), "Swedish Network Development for Implementing National Work Reform Strategy", *Human Relations*, Vol. 46 No. 2, pp. 219–248.
- Gold, A.H., Malhotra, A. and Segars, A.H. (2001), "Knowledge management: An organizational capabilities perspective", *Journal of management information systems*, Vol. 18 No. 1, pp. 185–214.

- Greenbaum, J. and Kyng, M. (Eds.). (1991), *Design at Work: Cooperative Design of Computer Systems*, CRC Press, 1sted.
- Hanseth, O. and Monteiro, E. (1997), "Inscribing behaviour in information infrastructure standards", *Accounting Management and Information Technologies*, Vol. 7, pp. 183–212.
- Hatakka, M. (2009), "Build it and They Will Come? – Inhibiting Factors for Reuse of Open Content in Developing Countries", *The Electronic Journal of Information Systems in Developing Countries*, Vol. 37 No. 5.
- Jelen, B. and Alexander, M. (2005), *Pivot Table Data Crunching*, Que Publishing, 1sted.
- Kanjo, C. (2011), "Pragmatism or Policy: Implications on Health Information Systems Success", *The Electronic Journal of Information Systems in Developing Countries*, Vol. 48 No. 1.
- Keny, P. and Chemburkar, A. (2006), "Trends in Operational BI", *DM REVIEW*, Vol. 16 No. 7, p. 20.
- Kimaro, H.C. and Nhampossa, J.L. (2005), "Analyzing the problem of unsustainable health information systems in less-developed economies: Case studies from Tanzania and Mozambique", *Information Technology for Development*, Vol. 11 No. 3, pp. 273–298.
- Kobielus, J. (2009), "Mighty mashups: do-it-yourself business intelligence for the new economy", *Forrester Research*.
- Laney, D. (2001), "3-D Data Management: Controlling Data Volume", *Velocity and Variety*, META Group Original Research Note.
- Lawton, G. (2006), "Making Business Intelligence More Useful", *Computer*, Vol. 39 No. 9, pp. 14–16.
- Littlejohns, P., Wyatt, J.C. and Garvican, L. (2003), "Evaluating computerised health information systems: hard lessons still to be learnt", *BMJ*, Vol. 326 No. 7394, pp. 860–863.
- Lungo, J.H. (2008), "The reliability and usability of district health information software: case studies from Tanzania", *Tanzania journal of health research*, Vol. 10 No. 1, pp. 39–45.
- Manovich, L. (2011), "Trending: The Promises and the Challenges of Big Social Data.", *Debates in the Digital Humanities*, available at: http://www.manovich.net/DOCS/Manovich_trending_paper.pdf (accessed 14 February 2013).
- March, S.T. and Hevner, A.R. (2007), "Integrated decision support systems: A data warehousing perspective", *Decision Support Systems*, Vol. 43 No. 3, pp. 1031–1043.
- Markus, M.L. and Keil, M. (1994), "If we build it, they will come: Designing information systems that people want to use", *Sloan Management Review*, Vol. 35, pp. 11–11.
- Monteiro, E. (2003), "Integrating health information systems: a critical appraisal", *Methods of information in medicine*, Vol. 42 No. 4, pp. 428–432.
- Mossialos, E., Allin, S. and Davaki, K. (2005), "Analysing the Greek health system: a tale of fragmentation and inertia", *Health Economics*, Vol. 14 No. S1, pp. S151–S168.
- O'Brien, R.C. (1991), "Brief case: EIS and strategic control", *Long Range Planning*, Vol. 24 No. 5, pp. 125–127.
- Orenstein, W. and Bernier, R. (1990), "Surveillance. Information for action.", *Pediatric clinics of North America*, Vol. 37 No. 3, pp. 709–734.
- Sandiford, P., Annett, H. and Cibulskis, R. (1992), "What can information systems do for primary health care? An international perspective", *Social Science & Medicine*, Vol. 34 No. 10, pp. 1077–1087.
- Sheikh, Y.H. and Bakar, A.D. (2012), "Open Source Software Solution for Healthcare: The Case of Health Information System in Zanzibar", *e-Infrastructure and e-Services for Developing Countries*, pp. 146–155.
- Silvestre, A.-L., Sue, V.M. and Allen, J.Y. (2009), "If You Build It, Will They Come? The Kaiser Permanente Model Of Online Health Care", *Health Affairs*, Vol. 28 No. 2, pp. 334–344.

- Snijders, C., Matzat, U. and Reips, U.-D. (2012), “‘Big Data’: Big Gaps of Knowledge in the Field of Internet Science”, *International Journal of Internet Science*, Vol. 7 No. 1, pp. 1–5.
- Stoops, N., Williamson, L., Krishna, S. and Madon, S. (2003), “Using health information for local action: facilitating organisational change in South Africa”, *The digital challenge: information technology in the development context*. Gateshead: Athenaeum Press Ltd.
- Watson, H.J. and Wixom, B.H. (2007), “The current state of business intelligence”, *Computer*, Vol. 40 No. 9, pp. 96–99.
- Webster, P.C. (2011), “The rise of open-source electronic health records”, *The Lancet*, Vol. 377 No. 9778, pp. 1641–1642.
- Weiss, S.M. and Indurkha, N. (1997), *Predictive Data Mining: A Practical Guide*, Morgan Kaufmann, 1sted.
- White, C. (2005), “The Next Generation of Business Intelligence: Operational BI”, *DM Review Magazine*.
- Williamson, L., Stoops, N. and Heywood, A. (2001), “Developing a district health information system in South Africa: a social process or technical solution?”, *Studies in health technology and informatics*, No. 1, pp. 773–777.
- Wixom, B.H. and Watson, H.J. (2001), “An empirical investigation of the factors affecting data warehousing success”, *MIS Q.*, Vol. 25 No. 1, pp. 17–32.
- Wong, J. and Hong, J.I. (2007), “Making mashups with marmite: towards end-user programming for the web”, *Proceedings of the SIGCHI conference on Human factors in computing systems*, pp. 1435–1444.
- Zikopoulos, P.C., deRoos, D., Parasuraman, K., Deutsch, T., Corrigan, D. and Giles, J. (2013), *Harness the power of Big Data: the IBM Big Data platform*, McGraw-Hill, New York; Singapore.

P6: Big Data Analytics for developing countries – Using the Cloud for Operational BI in Health

Jørn Braa
Department of Informatics,
University of Oslo,
Oslo, Norway

Saptarshi Purkayastha
Department of Computer and Information Science
Norwegian University of Science and Technology,
Trondheim, Norway

ABSTRACT

The multi-layered view of digital divide suggests there is inequality of access to ICT, inequality of capability to exploit ICT and inequality of outcomes after exploiting ICT. This is evidently clear in the health systems of developing countries. In this paper, we look at cloud computing being able to provide computing as a utility service that might bridge this digital divide for Health Information Systems in developing countries. We highlight the role of Operational Business Intelligence (BI) tools to be able to make better decisions in health service provisioning. Through the case of DHIS2 software and its Analytics-as-a-Service (AaaS) model, we look at how tools can exploit Cloud computing capabilities to perform analytics on Big Data that is resulting from integration of health data from multiple sources. Beyond looking at purely warehousing techniques, we suggest understanding Big Data from Organizational Capabilities and expanding organizational capabilities by offloading computing as a utility to vendors through cloud computing.

KEYWORDS

Cloud Computing; Analytics-as-a-Service; AaaS; Big Data; Digital divide; Organizational capability; Operational BI; Health Information Systems; DHIS 2

INTRODUCTION

Availability of resources for processing data has evolved over the years due to changes in computing technology such as from mainframes to client-server to cloud computing (Rajan & Jairath, 2011). In the past, with every major shift in technology infrastructure, larger datasets are processed in much shorter timeframes. Yet, within organizations in developing countries, observing smooth, linear and longitudinal technology shifts are rare when compared to organizations in developed countries. Many researchers refer to this as digital divide (Cullen, 2001; Hoffman & Novak, 1998; Barzilai-Nahon, 2003). Digital divide in such cases should not be understood from the simplistic notion of limited access to technology, but rather as a multi-layered concept (Wei et al, 2011) - of inequality of access to ICT (first-level), of inequality of

the capability to exploit ICT (second-level) and of inequality of outcomes (e.g., learning and productivity) after exploiting ICT (third-level). Since the spread of internet and easy access to internet-based software services, organizational users in developing countries have started to expect that their work applications will provide the same characteristics, as services offered using cloud computing - like On-demand self-service, multi-device access, resource pooling, rapid elasticity and measured service (Mell & Grance, 2011). In developing country organizational settings, we can see that internet access has becoming widely available. Even though not through high-speed wired connections, but using mobile networks (ITU, 2013). Our research is specifically in the Health Information Systems (HIS) domain, working with the ministries of health in developing countries where health systems are our organizations of study. Here, we ask our first research question – How can cloud computing contribute to bridge the multiple levels of digital divide within health systems?

Buyya et al. (2009) suggested that computing will one day be the 5th utility after water, electricity, gas and telephony due to cloud computing. Some of the currently available service models for providing such computing utility include Software-as-a-Service (SaaS), Platform-as-a-Service (PaaS), Infrastructure-as-a-Service (IaaS) and Analytics-as-a-Service (AaaS). In the future we expect more types of service models being created for cloud computing. Overall, we suggest that such computing utility will be useful to health systems, only if they allow health care providers to make better decisions during their daily health care provisioning. Recent research points to the use of Business Intelligence (BI) for managing the daily business operations (White, 2005). This has been referred to as Operational BI, as it is used to manage and optimize operational activities (Marjanovic, 2007). Although most literature characterizes Operational BI as real-time and low-latency availability of data, there is also acknowledgment that “Operational BI puts reporting and analytics application into the hands of users who can leverage information for their own operational activities” (Keny & Chemburkar, 2006). Others have referred to this as “real-time” (Azvine et al., 2006; Watson et al, 2006), but we prefer to use the term “right-time” and highlight operational availability to be the main difference, when referring to Operational BI in comparison to Real-time BI. Similarly, “use of information for local action” has been advocated by numerous researchers working with health information systems (Kimaro, 2006; Mosse & Byrne, 2005; Stoops et al., 2003). Here we ask our second research question – What service models of cloud computing can provide information for local action, so that better decisions can be made?

Through the examples in this paper, we suggest some strategies to exploit cloud computing such that they may be used to aid decision making in the health system. We look at the popular and arguably the largest

(Webster, 2011) open-source Health Management Information System (HMIS) called DHIS2 and how its Operational BI tools are used in developing countries to manage country-wide or state-wide health systems. In the paper, we highlight some of the new developments in DHIS2 that exploit Cloud Computing utility for Operational BI, moving from traditional warehousing approaches like data marts to more elastic and on-demand real-time analytics tools in DHIS2. In the discussion section, we suggest how these tools help manage Big Data from Organizational Capabilities perspective (Gold et al., 2001). We use the Overview-Overwhelm framework (Purkayastha & Braa, 2013) as a mechanism to understand the organizational view of Big Data and how “bigness” of data needs to be defined in terms of an organization’s capability.

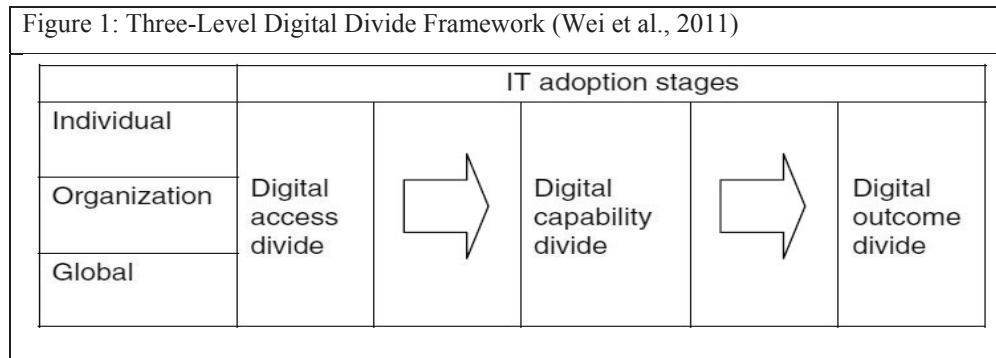
The next section of the paper explains Cloud Computing definitions, directions, concepts and service models. We also look at theoretical concepts of organizational perspective as well as Operational BI that have framed our research design. The research method is explained in the next section, followed by the case of Operational BI Tools in DHIS2. We discuss the synergies of cloud computing for Big Data analytics for health systems, especially from a developing country perspective in the next section. In the last section, we conclude with future directions for Analytics in the cloud.

CONCEPTUAL BASIS

The NIST (2011) defines cloud computing as a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction. This definition of cloud computing is quite useful to understand that the central theme is access to configurable computing resources. Although this definition is probably the most widely accepted, many researchers (Mowbray, 2009; Qian et al., 2009; Weiss, 2007) have highlighted that cloud computing is a vaguely defined, but much described phenomenon. Since the publication of core papers by Google in 2003 to commercialization by Amazon in 2006 and now to learning from the consumer internet and building an “Industrial Internet” of devices, appliances, aircrafts etc. (Bruner, 2013), there has been a steep rise in the number of research articles describing the phenomenon of cloud computing. A search of “Cloud computing” in the Scopus database of published articles in Title, Abstract and Keywords yields 12714 results, all published since 2008!! Yet, in this research spree, little has been said about the role of cloud computing in enhancing organizational capabilities and even less about outcomes resulting from such capabilities.

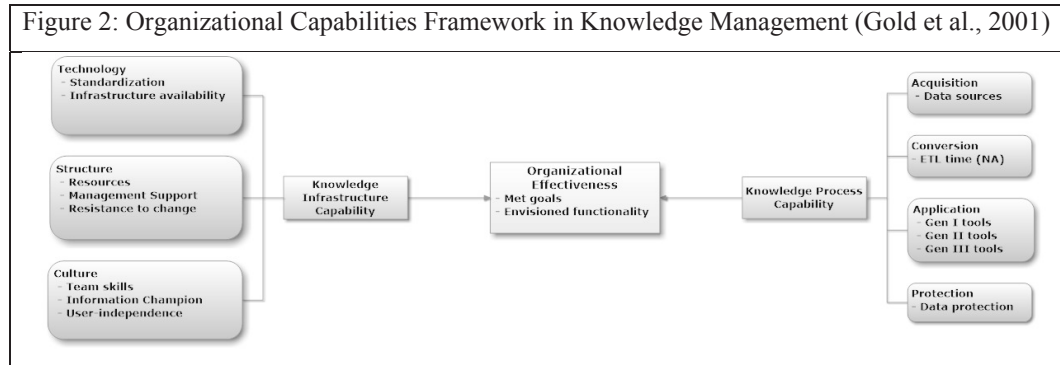
DIGITAL DIVIDE AS A FACTOR OF ORGANIZATIONAL CAPABILITIES

The inequality of capabilities and outcomes is central to the conceptualization of digital divide (Wei et al, 2011) that we believe cloud computing has impacted in the developing countries and in low-resource contexts even in developed countries.



We see that the digital divide ranges from individual to organization to global. We look at this model in our research context through the organization level. The first level of digital divide in our cases refers to the gap between health systems with regards to both their opportunities to access ICT and their use of the Internet for a wide variety of activities. The second level refers to inequality in the ability to use an accessible technology. In the above conceptualization of digital divide, the model relies on concepts from Social Cognitive Theory, particularly self-efficacy – “*the belief in one’s capability to organize and execute the courses of action required to manage prospective situations*” (Bandura, 1997 p.2). While Bandura (1986) articulates cognitive factors in terms of the individual’s capabilities, we draw parallels with Organizational Capabilities (Gold et al., 2001) as the organization’s capability to organize and execute courses of action. In health systems with regards to HIS, such capabilities mean ability to establish and ensure smooth functioning of the large infrastructure of medical services and health information capture and the ability to process this information into organizational knowledge (ibid.). To conceptualize capabilities of organizations for knowledge management, we use the Organizational Capabilities framework in terms of enlisted factors as shown in Fig 2. under Technology, Structure, Culture, Acquisition, Conversion, Application and Protection. With the specifics to DHIS2, we label the Operational BI tools under Application as Gen I, Gen II and Gen III tools (Purkayastha & Braa, 2013). The resultant inequality in Organizational Effectiveness or outcomes can be considered as the third-level

or Digital Outcome Divide. This can be understood as a gap between meeting goals of healthcare and envisioned functionality after the implementation of HIS.

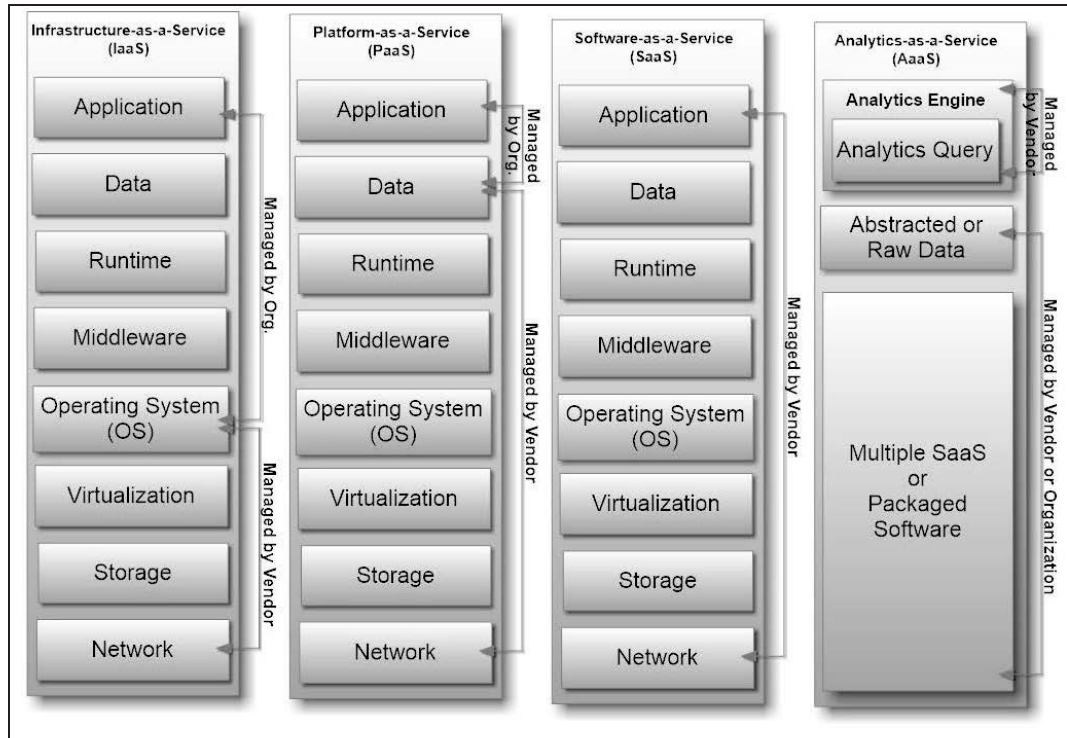


When we talk about access to ICT such as hardware and software within the health context, it is expected to be met through procurement of these in a cost-effective manner. Typically in developing countries, the debate is whether information systems should consume the available limited resources or whether these resources should be put to improving medical services. While this debate is relevant to the context, this paper and our research focus is on the design and use of health information systems. Thus, the access divide, capability divide and outcome divide are understood by us in terms of information processing, instead of health outcomes.

While bringing data together from different health programs and patient record systems, we challenge general techniques for information management. This has been referred to as “Big Data” challenge. With the integration of multiple sources of data into a data warehouse, we face the challenge of increased Volumes, increased Velocity and increased Variety of data. This has been commonly referred to as the “Three V” of Big Data (Laney, 2001). Weiss and Indurkha (1998) in computer science and Diebold (2000) in econometrics were one of the first to suggest practical principles of data mining for Big Data. Still, it is difficult to apply these principles in the context of low-resource settings, where the constraints in themselves add to the “bigness” of incoming data. The nature of data in health systems is also obvious to the “Three V”, because with population increase, the medical observations, health indicators and health facilities are always on the increase. When any current incoming data point has to be analyzed with the vast snapshot of existing historical data, the exponential increase in complexity poses problems of processing, storage and analytics. What if we only compare the incoming data point to a somewhat smaller snapshot of data? A snapshot of data that is relevant to the locale? - Locale being geography and/or health programs and/or facilities and/or health providers etc. The multiplicity of service models for cloud

computing started when consumer internet companies began offering internet-based alternatives to desktop applications, hosted over a number of distributed virtualized servers. So, an organization “*does not buy software license for an application such as enterprise resource planning (ERP), instead a business signs up to use the application hosted by the company that develops and sells the software, giving the buyer more flexibility to switch vendors and perhaps fewer headaches in maintaining the software*” (Dubey & Wagle, 2007 p.1). Such software delivery model from Application Service Providers (ASPs) (Günther et al., 2001) eventually became known as Software-As-A-Service (SaaS). When the software that’s made available as a service is a database system, the model is often labeled as Database-as-a-Service (DaaS). From an historical research perspective, it is important to recognize that early visions of turning software into a service (Turner et al., 2003) suggested use of multiple low-level services and providing a larger service using SOA (Gold et al, 2004; Papazoglou, 2003). But when considered today, SaaS is generally a single ASP product with little integration with other open services (Sun et al., 2007; Wei & Blake, 2010) and sometimes deliberately as walled-gardens to cause lock-ins (Cerbo et al., 2012). Thus, in the SaaS model, as shown in Figure 3, all parts of the service stack such as the hardware, storage, networking, OS, runtime, data and application is managed by an external vendor. On the service stack, when the application is managed internally by the organization and everything else is managed by the external vendor, the service model is commonly known as Platform-as-a-Service (PaaS). In the PaaS model, the vendor provides a platform on which applications are written to access underlying services. Social networking platforms, app hosting platforms and custom-CRM platforms on the cloud are examples of PaaS.

Figure 3: Relevant Cloud Computing Service Models



At the lowest spectrum of the service models, when an external vendor manages the hardware (generally based on virtualization) and the organization internally manages the middleware, runtime, data and application, it is known as Infrastructure-as-a-Service (IaaS). The IaaS model relies on the principle that hardware investments become obsolete and organizations do not have to own such capital costs upfront. Rather, organizations rent server clouds that can host runtime, middleware, data and applications for them and the infrastructure maintenance is left to the vendor. IaaS providers offer customizable OS, Runtime, CPU, Disk etc from the stack and offer on-demand changes to the hardware. This is referred to as “elastic site” (Marshall et al. 2010), so that when the application needs more resources, the underlying infrastructure grows as much as the application needs. Such elastic computing allows running analytic services at reduced cost since larger infrastructure is rented only when analytics need to be performed (Buyya et al, 2010 p.106). At other times, basic computing resources from IaaS provider are used and thus lower cost of renting it. When such analytics is provided as a service by an external vendor, this is referred to as Analytics-as-a-Service (AaaS). So the Analytics Engine consisting of algorithms and query processor is setup by the vendor, while rest of the transactional data and required software, platform and

infrastructure is at the organization. The AaaS vendor generally hosts the Analytics Engine on IaaS and performs the analytics quickly, possibly co-relating with external data in public domain and gives results back to the organization.

RESEARCH METHOD

The research has been conducted as part of an ongoing, long term action-research project. Our empirical investigations can be broadly placed in the Scandinavian Action Research tradition, which focuses on long term engagement in the field, doing actions and understanding the effects of the actions. Our approach is inspired by networked action research (Elden and Chisholm 1993; Engelstad and Gustavsen 1993), and has elsewhere been termed *Networks of Action* (Braa et al. 2004). The Networks of Action approach is specifically designed for the resource limited conditions in the Global South. The 18 years time-span of the research project exceeds traditional ‘projects’ and is more akin to social movements (Elden and Chisholm 1993). This network is called HISP (Health Information Systems Programme) due to the origins in the research project and comprises of researchers, developers, implementers, representatives from ministries who share knowledge and learning between the different nodes of the network. The authors of this paper are part of the central co-ordination node of the network.

Data for this research has been generated through active engagement with a global network of developers, implementers and users, who work in health systems of a number of developing countries. Emails, developer meetings, design discussion notes have been secondary sources of data. Other secondary sources of data include earlier published work by the authors and colleagues as well as consulting assignment for countries where DHIS2 has been widely deployed as the national system for health information management. The analysis of the data has been done with an interpretative perspective. Different cycles of interpretation were shared between the authors and multiple members of the network have participated in discussions about the use of cloud services. One of the authors is the originator of the project in the mid-1990’s and has participated in the design, development and implementation of HIS in many countries in Africa and Asia. The second author has been part of the research project for the last 4 years, primarily engaged in Asia and is part of the core developer team for the DHIS2 software.

OPERATIONAL BI TOOLS IN DHIS2

The DHIS 2 is a tool for collection, validation, analysis, and presentation of aggregate statistical data, tailored (but not limited) to integrated health information management activities. It is a generic tool rather

than a pre-configured database application, with an open meta-data model and a flexible user interface that allows the user to design the contents of a specific information system without the need for programming.

DHIS 2 has been implemented in more than 30 countries in Africa, Asia, Latin America and the South Pacific, and countries that have adopted DHIS 2 as their nation-wide HIS software include Kenya, Tanzania, Uganda, Rwanda, Ghana, Liberia, and Bangladesh. The software is developed through open-source collaboration and has been iteratively developed through an action-research initiative over the last 15 years. The software has its roots in Scandinavian Action Research tradition in IS development where user participation, evolutionary approaches, and prototyping are emphasized (Greenbaum & King, 1991). Its development started with the reforms in health sector in post-apartheid South Africa and has evolved into a large-scale web system that is now used for country-wide management of health information systems. Over this period, lot of BI tools have become part of the application, but its core agenda has been the use of information for local action through the use of flexible standards (Braa et al., 2007).

The Operational BI Tools in DHIS2 have followed 3 large generational changes, similar to what has been identified as BI&A 1.0, 2.0 and 3.0 (Chen et al., 2012). These changes from Pivot tables to Data marts to data mash-ups for interpretations within inter-organizational social networks have been done to meet Big Data needs (Purkayastha & Braa, 2013). These highlight the need for information at every level and information that has been analyzed and understood at each level. In the next sections, we describe how DHIS2 deployments have evolved because the Operational BI tools to manage Big Data require larger computational power.

MOVING FROM PACKAGED OFFLINE DEPLOYMENT TO CLOUD

The first versions of DHIS were developed as a Microsoft Access™ application that was meant to be used at each district. Districts would then share indicators and the national level would co-relate the data between districts. This decentralized system was widely acknowledged as an empowering tool for health workers (Williamson et al, 2001). When work on the next major version called DHIS2 started nearly 10yrs after the first release, it was developed as a web application that could be accessed over a browser and could be deployed on a variety of proprietary or open-source OS, Runtime and Databases. Thus, the focus of DHIS2 was for web deployments, but not losing sight that information had to be available for local action such that facilities and districts could analyze and access their own data, as well as probably get comparisons with other health facilities. Since internet access was still limited in countries like Sierra

Leone, there were offline installations of DHIS2 at each district office (Kossi et al., 2013). Thus, the DHIS2 web application was installed at District health offices like packaged software and accessed over a web browser. At nearly the same time, DHIS2 was also being implemented in Kenya, where internet penetration over mobile networks was increasing. In Kenya, the plan was initially to do many standalone implementations in districts, but due to availability of mobile internet through USB dongles, it was decided that the Kenya deployment would be done on an internet-based web server and all facilities across Kenya would use mobile internet to access the DHIS2 system through a computer (ibid.). This has been described as Participatory Design in the Cloud and empowered rural communities to participate in the design of the datasets in the HMIS system in Kenya (ibid.). This was a major change from how DHIS2 had been deployed earlier. Although deployment in Kenya may be described as a cloud deployment, it is really a hosted deployment using IaaS provider like Linode or AWS. These IaaS vendors allowed hosting DHIS2 outside Kenya's health department's servers. This puts the management of infrastructure with an external vendor instead of expecting Kenya's health department to build expertise in ICT management capabilities. Interestingly, since the Kenya deployment, most DHIS2 deployments are now hosted on IaaS externally and in the next sections of the paper we describe how DHIS2 service model has moved to exploit cloud capabilities even more by moving to PaaS, SaaS and eventually AaaS.

DEPLOYING DHIS2 AS A PAAS

DHIS2 has often been described as a platform by many researchers (Manoj et al., 2013; Lowe et al., 2012). DHIS2 in these instances is a platform for integrating health programs and sharing or comparing indicators from different sources of data. As DHIS2 deployments moved to a hosted environment, the ease of access to data over the internet at any-time, gave incentives to health programs to use DHIS2 as their health data repository (Sæbø et al, 2011). DHIS2 deployment in Kenya took more than a year for starting to get data from all facilities, but it continues to evolve by integrating other health programs. Interestingly, cloud-hosted DHIS2 deployments today not only integrate aggregate data from health programs, there are also attempts to integrate patient-level health information coming from multiple Electronic Medical Records (EMR) systems. In these cases DHIS2 deployment from the national ministry acts as a platform provider for different health programs in the country. The DHIS2 platform is available for customization by the different administrations, where they include their own datasets that are representative of their work practices and still are able to co-relate to data elements from other programs. In Malawi, the DHIS2 system has been deployed as a much more complex PaaS model. Lowe et al., (2012) describe the use of Ministry

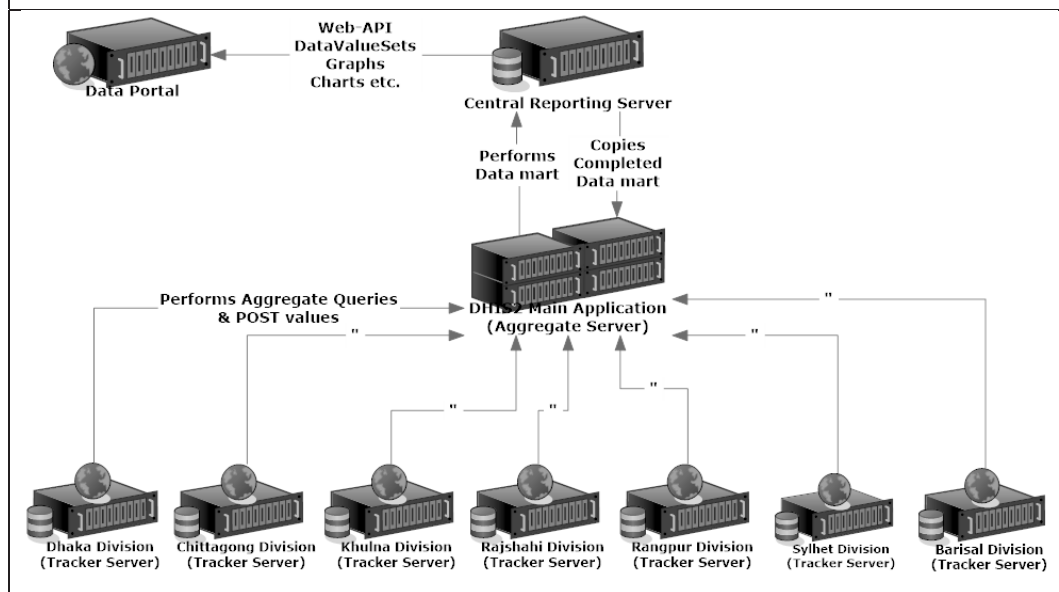
of Health's DHIS2 PaaS to integrate climate information with rural telemedicine. They use the PaaS platform for statistical and dynamical disease prediction models to be rapidly updated with real-time climate and epidemiological information. This permits health authorities to target timely interventions ahead of an imminent increase in malaria incidence (*ibid*). Yet, we have only been able to scratch the surface of the PaaS model in the DHIS2 deployments. With Open Data and further activities of Open Government (Janssen, 2011), it seems that DHIS2 PaaS deployment could scale to much larger levels in the future. These will enable governments to become PaaS providers to citizens, so that better transparency and information is available to citizens to build different applications on top of these. Recent versions of DHIS2 have also included an App framework that allows installation of new user-interfaces and customizations to the DHIS2 REST interfaces. This not only simplifies customizations for the users, but extends DHIS2 platform capabilities to a completely new level.

CONSIDERATIONS FOR CLOUD DEPLOYMENTS

A medium sized country in South-East Asia has deployed DHIS2 as packaged software within the Ministry of Health's datacenter since the year 2011 to manage routine HMIS. Unlike Kenya, they did not want to use IaaS provider mainly because of privacy and data ownership issues. Their DHIS2 deployment is mainly dependent on 2nd generation of Operational BI Tools (Purkayastha & Braa, 2013) like charts, comparative reports between facilities etc. These Operational BI tools require creation of data marts from the ETL process of the DHIS2 data warehouse. At the current stage, the country database for 2 years contains 36 million records until sub-district level consisting of ~9000 organization units. Below this level, there are ~16000 health facilities from which more granular data entry will soon start. The plan is then to capture patient-records from these facilities by converting paper registers from programs into electronic program tracking with DHIS2. Given the current capacity of the server, this scaling to lower level of health facilities and higher granularity of data is extremely computing resource intensive. At present, the data mart service on their DHIS2 deployment on a single, expensive server does not get finished within an expected period of nightly maintenance, so that during the next working day, the resultant analytics can be used. It requires more than 16hrs to complete the process of building the data mart. After the data mart is completed, the resulting analysis is somewhat stale and loses meaning for the facilities that want to track patients on a real-time basis. Thus, to be able to manage such Big Data, the architecture of the HMIS needed to change. Figure 4 shows a federated architecture that allows distributing load and data between multiple servers. This new federated architecture requires buying new servers and setting up of required

infrastructure is expensive. The maintenance effort to manage data across multiple servers is also an overhead in the performance, whenever data needs to be co-related between organizations that are hosted in different servers.

Figure 4. – An expensive federated architecture in a SE-Asia country to manage Big Data



The real use of the federated system is only nightly when the analytics service is running, so that patient records from the last day's activities can be used for services that need to be provided on the next day. The limited use of analytics in their current 16hr long data mart process has resulted in lack of perceived usefulness of system by data managers, health administration and district health officers. Other than being expensive, with Big Data the Total Cost of Ownership (TCO) of the system has considerably increased due to maintenance, setup, bandwidth etc. This is expected to increase exponentially as more patient records are captured and more programs are included in tracking of the patient. The 3Vs of Big Data has now resulted in high-demand for computing, only during a short period of that time, that if rented out through an AaaS provider can decrease the TCO.

EXPANDING DHIS2 TO AAAS SERVICE MODEL

While the DHIS2 PaaS service model has been adopted in a number of countries, providing a platform for health programs, organizations, international donors etc, there have been till date limited number of SaaS providers for DHIS2. But in the last year or so, AaaS service model is starting to gain momentum,

particularly with integration with patient-level data and need for dynamic reporting. When DHIS2 is hosted by a ministry of health, it publishes a list of indicators that allow meaningful use and minimum datasets that are useful for monitoring and evaluation of services provided by health providers and facilities. The transactional data from these services are located in paper registers or EMR System in facilities. A facility can then capture these paper records as transactions in the DHIS2 tracker system or a separate EMR. They can also just calculate aggregates on paper and report on these as aggregate values into DHIS2. The DHIS2 can then generate the required indicators for the facilities.

While the above mentioned workflow is common, there is a growing need for existing EMR systems with patient-level records to be able to perform analytics on data in their systems and compare with public health data that is available, as well as comparative studies between EMR systems for disease surveillance, outbreak alerts etc. Such analytics are computationally intensive tasks as we have seen earlier. Rather than expecting powerful computers available at facilities, the suggested model of AaaS is where an AaaS vendor performs analytics and provides results back to the EMR system, along with the earlier mentioned workflow of reporting to the ministry of health or another monitoring organization. AaaS service provided by DHIS2 relies on the hosted IaaS and covers the computational load on behalf of the EMR systems. This offloading of resources at the central location, allows better use of the underlying IaaS over time. E.g. while the analytics service from Philippines is running, there is still data entry happening in the Kenya system and Kenya does not require the analytics to be run, until the time Philippines begins its data entry. If such data can be well abstracted such that data between the organizations are separated with strict access controls, there are enough cost savings to justify the bandwidth required to send data to an analytics provider. Thus, the offloading computing utility is better shared with actually sharing the patient-records from the EMR systems. The AaaS service provider only needs to cover the analytics algorithm, abstracted data and the time-to-load from the incoming data from the EMR systems.

Since the Analytics engine runs on IaaS service provider, it can expand its required computing resources on-demand, multi-device access, resource pooling, rapid elasticity and measured service. The EMR systems individually rent only a miniscule amount of the AaaS computing resource and hence do the same type (or even advanced) analytics at a much more reduced pricing.

The DHIS2 analytics API can also be combined with the latest technology innovations from HTML5 standard such as web sockets for low-latency IO and quick response, when compared with earlier AJAX or similar web technologies. The DHIS2 analytics API also produces images, pdf, excel sheets etc. based on the parameters sent to the web service. These resources can be cached on the EMR client-side and

reduce the amount of bandwidth required on each web service call. The resources also allow simplified implementation of the client system. The client-side does not have to embed e.g. charting libraries or libraries that convert to different data formats. These data formats can be directly retrieved from the DHIS2 Analytics API over standard HTTP responses.

ANALYSIS & DISCUSSION

To think of our earlier concepts on digital divide, we see each service model as being able to increase organizational capabilities and thus building bridges for digital divide. The DHIS2 deployments at Kenya in comparison to Sierra Leone highlights the significance of deployment over IaaS, as providing access to many different types of users. Shared access to information across different hierarchical boundaries enables improved access to services. This has been made possible by moving for a packaged software model to an offline deployment model to use of a cloud service provider. We can think of this as a bridge to reduce the access-level digital divide. So, while we actually centralized the infrastructure through the cloud deployment in the IaaS model, the access to data actually became much more decentralized. This decentralization helps bridge access-level digital divide as it provides the same services to a larger group of facilities and health providers by allowing them to connect over the internet at any location.

On the other hand, then different organizations, facilities, health programs and users share the data and underlying infrastructure, we see that they can share their interpretations that have been generated from Operational BI tools. This we refer to as Gen III Operational BI tools that are closely linked to building inter-organizational social network over which users share their data interpretations. Deployment over PaaS service model allowed users in Kenya to share their interpretations that are generated from Operational BI Tools. This has helped improved shared understanding of information that is generated in DHIS2. We consider this to be a large leap of capabilities that are now available to organizations due to PaaS deployment. Without this shared access to data and its interpretations, such knowledge capabilities would be very hard to achieve within the organization. While we believe, that such knowledge generation capabilities are as much about access first and then about use, when we see certain users share interpretations, it encourages other users to comment and share their own interpretations of the data. Thus, PaaS helps bridge capability-level digital divide.

As EMR systems are primarily transaction systems that capture all the services provided by facilities or providers to patients, the amount of data generated by singular, disconnected systems become much more meaningful when they can share their data with other EMRs. In terms of continuity of care, it provides

patients with much more flexibility to get services at different locations. Yet, when different EMR systems want to communicate with each other about outcomes from patient transactions, there are challenges of semantic and syntactic interoperability (Sheth, 1999). These challenges can be simplified if outcomes can be understood through a common set of minimum datasets leading to meaningful use of data. Such services are easily applicable through AaaS service models, because they are dependent on providing analysis without all the systems having to talk to each other. The analysis from AaaS allows EMR systems to realize the similarities or dissimilarities in the outcomes of the services provided in different facilities. The limited capabilities of the EMR systems and the organizations that use these systems can be increased in terms of analytic capabilities through the use of AaaS services. Thus, AaaS service model helps in bridging the outcome-level digital divide.

While the move from packaged software deployment model to IaaS deployment model helped bridge the access-level digital divide, there is simply more to it than meets the eye. Countries like Sierra Leone and Ghana during deployment had limited technology capabilities within their organizations. The IaaS deployment model allowed focusing organizational resources towards data management, health management and service provisioning rather than learning new technologies for infrastructure management. On the other hand, use of PaaS allows bridging capability divide between the organizations. The shared indicator repository for example, allows organizations to share best practices in health information management at a country-level or multi-country organization (e.g. WAHO). Over the extended DHIS2 platform, the countries can deploy their own application (e.g. Liberia, Togo, Gambia all deploying applications on the WAHO DHIS2) and these applications are only parts that the country needs to manage. Rest of the service stack is all managed by WAHO. Some indicators and datasets are specific to the country level, like the hierarchy of standards (Braa et al. 2007). This data from the service stack is managed by the countries, while the WAHO-level indicators and datasets including the analytics is managed by WAHO. Thus, WAHO here acts as a vendor for the country managing the PaaS infrastructure and providing services to the member states.

DHIS2 deployments off late have not yet adopted the SaaS service model. Basically this is due to a limitation of a vendor that can provide services to multiple countries. Still, there is emerging possibility that universities in East Africa can provide such SaaS services to countries or organizations in the East Africa region. There is a strong business case for a DHIS2 SaaS provider. Global funding agencies might also play an important role in providing seed funding to a vendor who can provide DHIS2 SaaS services here. Though, the conflicts between the organizations need to be resolved first for such a SaaS model to

take-off. Countries that are in conflict would not want to have their computing resources shared with others on the same SaaS provider. This could become a strong deterrent and it will be up to the vendor to resolve such conflicts on an urgent basis. Yet, we have not studied independent variables of power in our research and is thus out of scope for this paper.

Although this creates a type of dependence on an external vendor, the organization focuses on building its own capabilities of structure and is able to better utilize its inherent, sentient capabilities. While our surveys do not clearly derive a co-relation between data security and organizational effectiveness (Purkayastha & Braa, 2013), this might be a result of the openness to use cloud computing. If the organizations involved in DHIS2 deployments were in countries that had stricter guidelines to data ownership and place where data needs to be hosted, the results would probably be different. But given that we have been able to actively deploy DHIS2 in countries using cloud computing, we are convinced that the co-relation between data security and perceived use of information system is less strong than what Wixom & Watson (2001) suggest based on studies in developed countries. In DHIS2 deployments, our findings highlight that cultural and structural factors play a larger role with a stronger co-relation between these factors and organizational effectiveness of Operational BI Tools (Purkayastha & Braa, 2013)

In the earlier mentioned SE-Asia country, where data ownership issues meant that they had to invest in a lot of infrastructure, has resulted in a higher TCO for the DHIS2 implementation. The overall direction for the use of DHIS2 on cloud computing is thus positive. There are still concerns of data ownership and many countries or ministries in these countries have fear of snooping, changes to data by vendor organizations have been highlighted in our discussions. We also see that some ministries of health would be willing to offload abstracted patient-level information to AaaS. This finding would be interesting to a number of IaaS service providers and would like to move their focus to marketing their solutions to AaaS developers.

BIG DATA ANALYTICS ON NEXT-GEN HEALTHCARE INDUSTRY

While we see that there is a general shift in the HIS industry from a single EMR to Health Information Exchanges (HIE), there is lack of clarity on how this shift will happen in the developing world. There is the technical challenge of moving from patient health records in a single EMR system to shared health record coming from multiple transactional systems such as EMRs, Lab Information Systems, Human Resource Systems, Drug Logistics Systems etc. But beyond this technical challenge of integration, there is also a larger organizational integration challenge that will involve lot of workarounds (Ellingsen et al,

2013). For example in the U.S, HIE implementations have faced persistent challenges over the past 20 years (Vest & Gamm, 2010) and we expect similar challenges, but also additional challenges of resource constraint and policy formulation will be faced by implementations in developing countries.

The main driving force behind implementing an HIE is mainly of public health significance. Shared health records may allow continuum of care and better patient-care based on preserved medical history might be a side-effect, the larger goal of HIEs is to be able to monitor health programs, government spending and detect disease outbreaks. This primary goal of HIEs has been rightly acknowledged by American Reinvestment & Recovery Act (ARRA), 2009 and similar policies around the world. These policy frameworks and laws put “meaningful use” through the sharing of required set of indicators from health-care providers to the forefront and monetize the use of EMRs through these indicators. This suggests that in the future of healthcare for medical providers to get paid, analytics will play a major role in medical practices (Frisse et al, 2011). As shown earlier, it is difficult for small EMR systems to build in such analytics, purely from an economic ROI perspective, and thus AaaS services will have an important role to play in “meaningful use” of medical information. In terms of HIE analytics, the core technical challenge has been Big Data Analytics and being able to handle large scale transactions from non-standard medical terminology in patient records. Both of these problems may be addressed through large-scale computing resources that can be made available through the cloud. For example in recent years, we see about 73% of trading in the US stock markets uses Algorithmic Trading (AT). This has been possible through co-relating data from various sources and allowing algorithms to self-learn and decide on investment strategies (Chaboud et al., 2013). To achieve the same in health care is somewhat optimistic because of the Variety challenge of Big Data, it is not impossible. Given the exponential rise in computing capabilities through the use of virtualization in data centers that provide cloud computing services, it is quite possible that machine learning algorithm for health care can be effectively designed in the near future. With shared medical terminologies between providers, the Variety challenge can be solved and AaaS cloud models can then easily deal with the Volume and Velocity problems. The Veracity challenge of Big Data is a little more complicated in health care than other industries. Especially in developing countries, reliable diagnosis is a challenge and often medical practitioners re-order tests (Alvarez, 2005). Even with advanced electronic system, this challenge of quality and Veracity of data is difficult to solve.

The strategy that we have followed with DHIS2 is that EMR systems should be able to work together through common medical terminologies at the patient-level and exchange metadata with a DHIS2 data warehouse to send public health information. DHIS2 can then act as an analytics engine and provide

feedback about data quality to the EMR systems. Standardizing shared metadata is often the biggest challenge in HIE implementations. Because of the flexibility of defining metadata in DHIS2, different kinds of transactional systems can be integrated through it and made part of an HIE initiative. DHIS2 thus participates in the development of the Open Health Information Exchange (OpenHIE – <http://ohie.org>) initiative that has been started to create open-source standards, metadata exchange and applications that can be brought together to create an HIE implementation in developing countries.

CONCLUSION

We see that cloud computing service models will continue to evolve over time, with more and more combination of the current models as well as new types of computing utilities being provided. These could become widely accepted, not just based on cost-savings to organizations, but will be based on how these models can improve organizational capabilities. How vendors that provide such utility can become assets to organizations to offload tasks that are outside an organization's core competencies. These bring focus to the organization's activities and help consolidate their existing capabilities. In our experiences in the health system, providing health services is the core competency of the organizations and offloading computing-related tasks to external vendor who provide reliable and efficient services will be the direction for the future.

To answer our original research questions, we definitely see that cloud computing helps bridge digital divide. IaaS (and probably SaaS) helps bridge access-level digital divide. PaaS model helps bridge capability digital divide. AaaS model helps improve outcomes and bridge outcome-level digital divide. In terms of characteristics of cloud services that allow information for local action, the on-demand and scalable services are the core to provide better decentralized access to data and analysis. Especially in terms of cost-savings, AaaS that's running on IaaS is a great leap forward. Capabilities to manage Big Data Analytics that were impossible to meet with limited hardware are now possible with the use of AaaS providers.

In terms of future direction of this research, we'd like to study the privacy and security concerns related to AaaS providers in much more details. In the light of recent events related to government snooping of private information, these discussions are relevant for the future of cloud computing. Privacy is highly relevant to the health context and hence will be important for future studies.

REFERENCES:

- Alvarez, R. (2005). Health care has to move into the hi-tech age. *Bulletin of the World Health Organization*, 83(5), 323–323. doi:10.1590/S0042-96862005000500003
- Azvine, B., Cui, Z., Nauck, D. D., & Majeed, B. (2006). Real Time Business Intelligence for the Adaptive Enterprise. In *The 8th IEEE International Conference on and Enterprise Computing, E-Commerce, and E-Services, The 3rd IEEE International Conference on E-Commerce Technology, 2006* (pp. 29–29). doi:10.1109/CEC-EEE.2006.73
- Bandura, A. (1986). Fearful expectations and avoidant actions as coefficients of perceived self-efficacy. *American Psychologist*, 41(12), 1389–1391. doi:10.1037/0003-066X.41.12.1389
- Bandura, A. (1997). *Self-efficacy: The exercise of control*. New York: Freeman.
- Barzilai-Nahon, K. (2006). Gaps and Bits: Conceptualizing Measurements for Digital Divide/s. *The Information Society*, 22(5), 269–278. doi:10.1080/01972240600903953
- Braa, J., Hanseth, O., Heywood, A., Mohammed, W., & Shaw, V. (2007). Developing health information systems in developing countries: the flexible standards strategy. *MIS Quarterly*, 31(2), 381–402.
- Braa, J., Monteiro, E., & Sahay, S. (2004). Networks of Action: Sustainable Health Information Systems across Developing Countries. *MIS Quarterly*, 28(3), 337–362.
- Bruner, J. (2013). *Industrial Internet*. O'Reilly Media, Inc.
- Buyya, R., Broberg, J., & Goscinski, A. M. (2010). *Cloud Computing: Principles and Paradigms*. John Wiley & Sons.
- Buyya, R., Yeo, C. S., Venugopal, S., Broberg, J., & Brandic, I. (2009). Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility. *Future Generation Computer Systems*, 25(6), 599–616. doi:10.1016/j.future.2008.12.001
- Cerbo, F. D., Bezzi, M., Kaluvuri, S. P., Sabetta, A., Trabelsi, S., & Lotz, V. (2012). Towards a Trustworthy Service Marketplace for the Future Internet. *The Future Internet* (pp. 105–116). Springer Berlin Heidelberg.
- Chaboud, A., Chiquoine, B., Hjalmarsson, E., & Vega, C. (2013). *Rise of the Machines: Algorithmic Trading in the Foreign Exchange Market* (SSRN Scholarly Paper No. ID 1501135). Rochester, NY: Social Science Research Network. Retrieved from <http://papers.ssrn.com/abstract=1501135>
- Chen, H., Chiang, R. H. L., & Storey, V. C. (2012). Business intelligence and analytics: from big data to big impact. *MIS Q.*, 36(4), 1165–1188.
- Chisholm, R. F., & Elden, M. (1993). Features of Emerging Action Research. *Human Relations*, 46(2), 275–298. doi:10.1177/001872679304600207
- Cullen, R. (2001). Addressing the digital divide. *Online Information Review*, 25(5), 311–320. doi:10.1108/14684520110410517
- Diebold, F.X. (2000). Big Data Dynamic Factor Models for Macroeconomic Measurement and Forecasting," Discussion Read to the Eighth World Congress of the Econometric Society, Seattle, August 2000. Retrieved from <http://www.ssc.upenn.edu/~fdiebold/papers/paper40/temp-wc.PDF>
- Dubey, A., & Wagle, D. (2007). Delivering software as a service. *The McKinsey Quarterly*, 6, 1–12.
- Elden, M., & Chisholm, R. F. (1993). Emerging Varieties of Action Research: Introduction to the Special Issue. *Human Relations*, 46(2), 121–142. doi:10.1177/001872679304600201
- Ellingsen, G., Monteiro, E., & Røed, K. (2013). Integration as interdependent workaround. *International Journal of Medical Informatics*, 82(5), e161–e169. doi:10.1016/j.ijmedinf.2012.09.004
- Engelstad, P. H., & Gustavsen, B. (1993). Swedish Network Development for Implementing National Work Reform Strategy. *Human Relations*, 46(2), 219–248. doi:10.1177/001872679304600205
- Gold, A. H., Malhotra, A., & Segars, A. H. (2001). Knowledge management: An organizational capabilities perspective. *Journal of management information systems*, 18(1), 185–214.

- Gold, N., Mohan, A., Knight, C., & Munro, M. (2004). Understanding service-oriented software. *IEEE Software*, 21(2), 71–77. doi:10.1109/MS.2004.1270766
- Greenbaum, J., & Kyng, M. (Eds.). (1991). *Design at Work: Cooperative Design of Computer Systems* (1st ed.). CRC Press.
- Günther, O., Tamm, D.-W.-I. G., Hansen, D.-V. L., & Meseg, D.-K. T. (2001). Application Service Providers: Angebot, Nachfrage und langfristige Perspektiven. *Wirtschaftsinformatik*, 43(6), 555–567. doi:10.1007/BF03250818
- Hoffman, D. L., & Novak, T. P. (1998). Bridging the Racial Divide on the Internet. *Science*, 280(5362), 390–391. doi:10.1126/science.280.5362.390
- ITU, The World in 2013: ICT Facts and Figures. (2013). *ITU*. Retrieved September 5, 2013, from <http://www.itu.int/en/ITU-D/Statistics/Pages/facts/default.aspx>
- Janssen, K. (2011). The influence of the PSI directive on open government data: An overview of recent developments. *Government Information Quarterly*, 28(4), 446–456. doi:10.1016/j.giq.2011.01.004
- Keny, P., & Chemburkar, A. (2006). Trends in Operational BI. *DM REVIEW*, 16(7), 20.
- Kimaro, H. C. (2006). Strategies for Developing Human Resource Capacity to Support Sustainability of ICT Based Health Information Systems: A Case Study from Tanzania. *The Electronic Journal of Information Systems in Developing Countries*, 26(0). Retrieved from <https://www.ejisdc.org/ojs2/index.php/ejisdc/article/view/272>
- Kossi, E. K., Sæbø, J. I., Braa, J., Jalloh, M. M., & Many, A. (2012). Developing decentralised health information systems in developing countries –cases from Sierra Leone and Kenya. *The Journal of Community Informatics*, 9(2). Retrieved from <http://ci-journal.net/index.php/ciej/article/view/861>
- Laney, D. (2001). 3-D Data Management: Controlling Data Volume. *Velocity and Variety*", *META Group Original Research Note*.
- Lowe, R., Chadza, T., Chirombo, J., Fonda, C., Muyepa, A., Nkoloma, M., ... Zennaro, M. (2012). A platform to integrate climate information and rural telemedicine in Malawi. In *EGU General Assembly Conference Abstracts* (Vol. 14, p. 9503). Presented at the EGU General Assembly Conference Abstracts. Retrieved from <http://adsabs.harvard.edu/abs/2012EGUGA..14.9503L>
- Manoj, S., Wijekoon, A., Dharmawardhana, M., Wijesuriya, D., Rodrigo, S., Hewapathirana, R., ... Dissanayake, V. H. W. (2013). Implementation of District Health Information Software 2 (DHIS2) in Sri Lanka. *Sri Lanka Journal of Bio-Medical Informatics*, 3(4). doi:10.4038/sljbm.v3i4.5431
- Marjanovic, O. (2007). The Next Stage of Operational Business Intelligence: Creating New Challenges for Business Process Management. In *40th Annual Hawaii International Conference on System Sciences, 2007. HICSS 2007* (p. 215c–215c). doi:10.1109/HICSS.2007.551
- Marshall, P., Keahey, K., & Freeman, T. (2010). Elastic Site: Using Clouds to Elastically Extend Site Resources. In *Proceedings of the 2010 10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing* (pp. 43–52). Washington, DC, USA: IEEE Computer Society. doi:10.1109/CCGRID.2010.80
- Mell, P., & Grance, T. (2011). The NIST Definition of Cloud Computing, Recommendations of the National Institute of Standards and Technology. *National Institute of Standards and Technology*.
- Mosse, L., & Byrne, E. (2005). The role of identity in health information systems development: a case analysis from Mozambique. *Information Technology for Development*, 11(3), 227–243.
- Mowbray, M. (2009). The Fog over the Grimpen Mire: Cloud Computing and the Law. *SCRIPT-ed*, 6(1), 132–146. doi:10.2966/scrip.060109.132
- Papazoglou, M. P. (2003). Service-oriented computing: concepts, characteristics and directions. In *Proceedings of the Fourth International Conference on Web Information Systems Engineering, 2003. WISE 2003* (pp. 3–12). doi:10.1109/WISE.2003.1254461

- Purkayastha, S., & Braa, J. (2013). Overview, not overwhelm – Operational BI tools for Big Data in health information systems. *Operational BI for healthcare in developing countries*. Retrieved from http://folk.ntnu.no/saptarsp/Operational_BI_healthcare_LMIC.pdf
- Qian, L., Luo, Z., Du, Y., & Guo, L. (2009). Cloud Computing: An Overview. In M. G. Jaatun, G. Zhao, & C. Rong (Eds.), *Cloud Computing* (pp. 626–631). Springer Berlin Heidelberg.
- Rajan, S., & Jairath, A. (2011). Cloud Computing: The Fifth Generation of Computing. In *2011 International Conference on Communication Systems and Network Technologies (CSNT)* (pp. 665–667). doi:10.1109/CSNT.2011.143
- Sæbø, J. I., Kossi, E. K., Titlestad, O. H., Tohouri, R. R., & Braa, J. (2011). Comparing strategies to integrate health information systems following a data warehouse approach in four countries. *Information Technology for Development*, 17(1), 42–60.
- Sheth, A. P. (1999). Changing Focus on Interoperability in Information Systems: From System, Syntax, Structure to Semantics. In M. Goodchild, M. Egenhofer, R. Fegeas, & C. Kottman (Eds.), *Interoperating Geographic Information Systems* (pp. 5–29). Springer US. Retrieved from http://link.springer.com/chapter/10.1007/978-1-4615-5189-8_2
- Stoops, N., Williamson, L., Krishna, S., & Madon, S. (2003). Using health information for local action: facilitating organisational change in South Africa. *The digital challenge: information technology in the development context*. Gateshead: Athenaeum Press Ltd.
- Sun, W., Zhang, K., Chen, S.-K., Zhang, X., & Liang, H. (2007). Software as a Service: An Integration Perspective. In B. J. Krämer, K.-J. Lin, & P. Narasimhan (Eds.), *Service-Oriented Computing – ICSOC 2007* (pp. 558–569). Springer Berlin Heidelberg.
- Turner, M., Budgen, D., & Brereton, P. (2003). Turning software into a service. *Computer*, 36(10), 38–44. doi:10.1109/MC.2003.1236470
- Vest, J. R., & Gamm, L. D. (2010). Health information exchange: persistent challenges and new strategies. *Journal of the American Medical Informatics Association*, 17(3), 288–294. doi:10.1136/jamia.2010.003673
- Watson, H. J., Wixom, B. H., Hoffer, J. A., Anderson-Lehman, R., & Reynolds, A. M. (2006). Real-Time Business Intelligence: Best Practices at Continental Airlines. *Information Systems Management*, 23(1), 7–18. doi:10.1201/1078.10580530/45769.23.1.20061201/91768.2
- Webster, P. C. (2011). The rise of open-source electronic health records. *The Lancet*, 377(9778), 1641–1642. doi:10.1016/S0140-6736(11)60659-4
- Wei, K.-K., Teo, H.-H., Chan, H. C., & Tan, B. C. Y. (2011). Conceptualizing and Testing a Social Cognitive Model of the Digital Divide. *Information Systems Research*, 22(1), 170–187. doi:10.1287/isre.1090.0273
- Wei, Y., & Blake, M. B. (2010). Service-Oriented Computing and Cloud Computing: Challenges and Opportunities. *IEEE Internet Computing*, 14(6), 72–75. doi:10.1109/MIC.2010.147
- Weiss, A. (2007). Computing in the clouds. *netWorker*, 11(4), 16–25. doi:10.1145/1327512.1327513
- Weiss, S. M., & Indurkha, N. (1997). *Predictive Data Mining: A Practical Guide* (1st ed.). Morgan Kaufmann.
- White, C. (2005). The Next Generation of Business Intelligence: Operational BI. *DM Review Magazine*.
- Williamson, L., Stoops, N., & Heywood, A. (2001). Developing a district health information system in South Africa: a social process or technical solution? *Studies in health technology and informatics*, (1), 773–777.
- Wixom, B. H., & Watson, H. J. (2001). An empirical investigation of the factors affecting data warehousing success. *MIS Q.*, 25(1), 17–32. doi:10.2307/3250957

Appendix B: Secondary papers

The attached paper provides an avenue to highlight the future work that has been started in leHIs. The paper presents a new idea for health information exchanges that can deal with the problem of semantic interoperability that has been central to technology discussion around integration. It is a different take on integration from the thesis and hence is not part of the collection of paper that is part of the thesis. But it is appended here as related piece of work.

SP1: HIXEn: An integration engine for multi-vocabulary health information using REST & semantic metadata mapping

HIXen: An integration engine for multi-vocabulary health information using REST & semantic metadata mapping

Saptarshi Purkayastha

Department of Computer & Information Science

Norwegian University of Science & Technology

Trondheim, Norway

e-mail: saptarsp@ifi.uio.no

Abstract—Integration of Health Information Systems (HIS) has been a challenge because of the different semantics that are used by health-care practices and different levels of health-care systems need different kinds of information. Looking through the case of two widely used open-source HIS (one patient-level and one aggregate country-level system), the paper analyses multiple approaches for integration. The paper develops a novel integration engine (HIXen) that uses concepts of distributed hypermedia systems from RESTful architecture and Resource Descriptors from semantic web that can allow health information exchanges to flexibly connect different HIS

Keywords—health information systems; integration; semantic web; REST; RDF; DHIS2; OpenMRS

Introduction

Today, Health Information Systems (HIS) range from clinically relevant patient-level data to aggregate national indicators related to quality of care delivery and health program effectiveness. Research[1] has indicated that these information systems, especially in the context of developing countries [2], function in their own respective silos both technically and institutionally, and do not necessarily talk to each other. Health-care poses a special challenge in semantics because of the way medical practitioners convey health information. The systems that manage health information also use different terminologies and vocabularies [3]. Research has shown us that there is great value in health care information exchange and interoperability [4] [5]. More importantly in the case of primary healthcare where there is a shortage of skilled health care providers in developing countries [6]. While this challenge of integration and need has been highlighted by many researchers, it has also been discussed through different perspectives such as semantic [7] [8], political [9], web-services [10].

Other researchers [11] have classified integration solutions into models such as information-oriented integration, process-oriented integration, service-oriented integration, user-oriented integration. Information-oriented integration models convey how information exchange and its use are bound together. This involves exchange of information by understanding the semantics of what is required and bringing together information from various sources together. Process-oriented integration models bring together activities and processes that can be done together within an organization. This type of integration matches similar processes in vertical structures of organizations and integrates available information. Service-oriented integration makes common services available as an integrated service. A common login (SSO) or shared access to documents (DMS) or common printing services (CUPS) within an organization are examples of service-oriented integration. This paper presents a real-world integration engine that implements all of the above mentioned models through the use of RESTful web-services and Resource Description Framework (RDF).

Health Information eXchange Engine (HIXen) is a novel integration engine presented in the paper can be considered to be a reference implementation based on two large-scale and widely deployed open-source

systems. Similar engines can be implemented on top of other systems that are based on a meta-model design, also known as Entity-Attribute-Value (EAV) or open schema. The next section of the paper talks about theoretical basis of using an EAV in HIS and the challenges in integrating systems modeled as EAV. In section 3, data model of the patient-level system (OpenMRS) and the aggregate system (DHIS2) are described, along with high-level introductions of REST and RDF. The paper assumes that the reader is familiar with these technologies or can read detailed descriptions for REST [12] and RDF [13] for reference. Section 4 of the paper describes how medical vocabulary is managed in the two individual systems and some existing examples of attempts at integration. Section 5 presents the detailed working of HIXen and how it is different from ad-hoc methods. This section also describes how HIXen uses hypermedia, metadata mapping, rule interchange format and SPARQL.

Theoretical Basis of EAV in HIS

The EAV model is one of the more commonly used modeling styles in HIS [14]. Since clinical information for each patient has different values than the next patient, the “one-fact-per-column” notion of traditional database design is futile. Instead a method known as EAV or row modeling is used where “fact descriptors” i.e. attributes are treated as data. Thus, addition of new facts does not mean adding new columns to the database. Instead these facts become rows in the metadata table and their values are stored in another relational table. In a patient-level system, the patient or patient visit is the Entity; the type of fact (e.g. blood group) is the Attribute and answer (e.g. B+ve) of the fact is the Value. In aggregate-level system, the facility is the Entity, the aggregated fact (e.g. Number of malaria cases) is the Attribute and number (e.g. search result from patient-level system) is the Value.

This allows the information system to store a number of varied facts in the least number of columns as possible, but the Attribute binds the Entity and Value through a relationship, generally in a relational database system (RDBMS). The relationship can be expressed through a set of tuples in a relationship schema. A tuple t is defined as:

$t: R \rightarrow \bigcup_{i \in R} \text{Dom}(A_i)$ such that for every $A_i \in R$, $t(A_i) \in \text{dom}(A_i)$ and $\text{dom}(A_i)$ is a set of values of A_i . This may be represented in an entity diagram as follows:

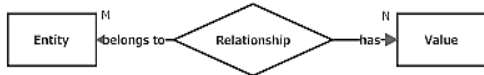


Fig. 15: An attribute specifying a relationship

A pseudo-query in EAV model for attributes (a, b) might be represented as *RETRIEVE values of Entities that have attributes such as a AND b*. When such queries are to be processed at multi-level systems (for integration of patient-level to aggregate-level systems), the problems are compounded. Attributes for aggregate system are basically sub-selects of *RETRIEVE* queries from patient-level systems as shown above and perform with complexity as $O(n^2)$ notation. This increases the complexity with each subsequent level of aggregation.

The EAV discussion is critical to understanding the challenges of integration because of the way metadata in HIS are flexibly created. The result is what we refer to as multi-vocabulary aspect of HIS, where the same software can be customized to hold different metadata. The abovementioned theoretical EAV modeling becomes clearer in the next section where we describe the way OpenMRS and DHIS2 store their data and metadata.

Data Modeling in Patient-level System (OpenMRS) and Aggregate-level System (DHIS2)

Open Medical Records System (OpenMRS)

OpenMRS is a widely used, free and open-source medical records system [15]. People from over 40 countries have implemented it at one level or another, from individual clinics to whole nations [16]. OpenMRS uses a hybrid model [17] where its model to store patient observations (*obs* table) are EAV, but patient demographic information is store in a dense non-EAV model where each attribute like *given_name*, *family_name*, *address*, *gender* are columns in the *person* table. This has been done because basic demographics data is constant. Flexible demographic is also storable through EAV *patient_attributes* table. OpenMRS also makes use of multiple columns in the *obs* table to represent Values of different types. So there is *value_numeric*, *value_coded*, *value_text* as columns in the *obs* table. Attributes for entities are referred to as Concept in OpenMRS and are store in the *concept* table in the database. Concept is the main unit of metadata in OpenMRS and these are medical questions, answers, symptom, diagnosis, procedure, tests etc. i.e. Concepts (also known as vocabularies) represent all kinds of Attributes that an Entity (Patient or an Encounter) might convey. This allows different kinds of facts to be stored as Concepts, but also makes the understanding of these for an external system more complex. Fig. 2 is a simplified ER-diagram to represent the discussed metadata in OpenMRS

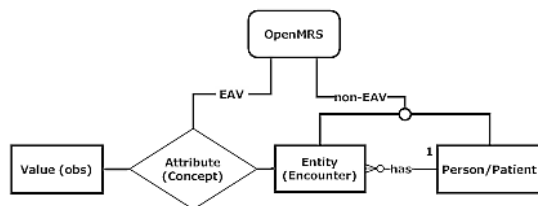


Fig. 16: OpenMRS EAV

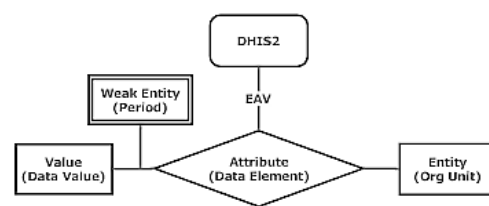


Fig. 17: The DHIS2 EAV

District Health Information Software v2 (DHIS2)

The DHIS 2 is a tool for collection, validation, analysis, and presentation of aggregate statistical data, tailored to integrated health information management activities. It is a generic tool rather than a pre-configured database application, with an open meta-data model and a flexible user interface that allows the user to design the contents of a specific information system without the need for programming [18]. The system is in use in over 20 different countries in Africa and Asia and arguably the largest implementation of any health information system by coverage of the number of health facilities [19]. By using a flexible standards approach [20], DHIS2 can allow configuration of indicators formulas for calculating and analyzing data from health facilities in a district, state or entire nation. DHIS2 uses a fully sparse EAV model and allows complete freedom to configure the data model through the web interface. The main Entity for data collection in DHIS2 is Organization Unit (OU). Each OU is the basic unit of reporting, which in our case is a health facility. Attributes are called as Data Elements (DE) and Values are stored as Data Values (DV). DHIS2 also has grouping of OU called Organization Unit groups and grouping of DE called as Datasets. Each dataset is the reported in practice over a fixed frequency of time-period, but it is the DE which are the attributes through which DV are stored and retrieved. Fig. 3 is a simplified ER-diagram to represent the discussed metadata in DHIS2.

Representational State Transfer (REST)

REST is an architectural style for distributed hypermedia systems. REST is a hybrid style derived from several of the network-based architectural styles and combined with additional constraints that define a uniform connector interface [12, pp.76]. REST is defined by four interface constraints: identification of resources; manipulation of resources through representations; self-descriptive messages; and, hypermedia as the engine of application state (ibid.).

The four constraints play an extremely important in the design and behavior of HIXEn. These constraints are applied on the components, connectors and data of the HIS that need to be integrated.

1. **Data Element** is way in which the architecture understands data. Resource, resource identifiers, representation, representation metadata, resource metadata and control data are data elements in REST. **Resource** is a unit of information abstraction in REST. A resource can be a document, image, temporal service, a collection of resources etc. Resources are identified by resource identifiers that are defined by a naming authority. The selection of identifiers in a distributed system is political as well as technical challenge that needs to be addressed by implementers of an integration engine. **Representation** is combination of - information of the state of the resource, its metadata and occasionally metadata about the metadata (for verifying integrity). The data format of the representation is called media-type.
2. **Connector** is used to encapsulate the behavior of a resource as well as transferring representation. REST prescribes connector types such as client, server, cache, resolver, tunnel etc. REST connectors are stateless, meaning they don't remember any previous interactions. There are many important side effects of statelessness that are beyond the scope of this paper.
3. **Components** are different actors in the REST-styled network classified based on their role.

An unassuming and critical constraint in REST is that all interactions between components are driven by hypermedia. Hypermedia means there are links that create branches of choices that can be followed by a reader. Hypertext is a subset of hypermedia in that there is simultaneous presentation of information in text and controls. Hypermedia includes temporal anchors inside of a media stream so that a reader of the media stream can use these anchors as controls.

Resource Description, Interchange Format & SPARQL

Resource Description Framework (RDF) is a data model for describing information about a resource. It can be an XML-based or shorthand called Notation 3. The HIXEn implementation as a choice uses XML-based RDF. In the REST paradigm, RDF includes its own resource identifier as the subject and resource representation as its object and predicates are identifiers of the objects. Thus, each object from the resource can be identified uniquely using the predicates inside the RDF.

Resource Interchange Format (RIF) is a format for writing rules about how to parse the objects based on the predicates in the RDF document. The rules allow the processing engine to be a reader and follow branches in the hypermedia. There are multiple dialects for RIF as described by the W3C standard. HIXEn uses the logic rules to translate data and to create branches when parsing RDF of the incoming requests and data.

“SPARQL Protocol and RDF Query Language” is a recursive acronym for the query language standard to search RDF documents. SPARQL allows distributed searching between nodes of a distributed system containing RDFs. SPARQL searches return objects from within RDFs. There is also a dialect of SPARQL

to make Update statements, but HIXEn does not update RDF documents directly. RDFs are presented by the connecting applications and only queried by HIXEn to match and compare RDFs.

Medical Vocabulary Management in OpenMRS and DHIS2

Both OpenMRS and DHIS2 are modular, extensible systems that add functionality through modules. These modules add new database tables and change the schema. This paper refers to the core of the two systems only. OpenMRS in its core constitutes of a Concept Dictionary for management of medical vocabulary. A Concept has name, description, synonyms (with other concepts), class (type of vocabulary being described as mention in sec.III.A earlier), data type and mappings with vocabulary from other sources. Concepts locally use ID, but these are useless externally since these are sequentially generated for the system and are not unique across different implementations of OpenMRS. The concepts do have uuids that allow them to be uniquely identified and can be used as machine-capable resource identifiers. These when merged with implementation id (a name given to the installation) is used by HIXEn as the resource identifiers for concept sharing. The concept mappings also play an important role in describing the metadata and linking with other systems. Although this has to be done manually when defining your own concept, HIXEn processing adds a layer over this and simplifies manual mapping with other vocabulary sources. OpenMRS in its core uses HL7 to share records of individual patients with other patient-level system, but not metadata.

Table 6: OpenMRS vocabulary metadata for HIXEn

Metadata name	Metadata description
uuid	The globally unique identifier for a concept, but not identifiable without implementation id
name	A common name for the concept
description	Description about the concept
synonyms	Related to another concept name in the system
class	Classification in medical terms of the concept
data type	The data type of the value that the concept describes
mappings	These are links to map a concept with another medical terminology from an external source

DHIS2 uses the DE management and Data dictionary management to manage the vocabulary in DHIS2. Each DE has name, alternative name, code, description, domain type, value type, aggregation operator, URL, combination of categories etc. These listed are the important parts that are to be used by external systems for integration. Each DE can be part of a DE group that can be used for analysis or reporting purposes. DHIS2 is used as a data warehouse and allows creation of data marts that store data in a more organized manner. This manner of organization is flexibly left up to the user depending on the kind of analysis that one wants to perform on the data. HIXEn uses raw data at the moment, but more intelligent decision support systems could create data marts based on query requirements. Similar to DE, DHIS2 also uses a calculated DE called Indicator. An indicator is created by applying a formula with numerator, denominator which is data values of data elements. So an Indicator can be defined from data elements and the result can be used in analysis tools inside DHIS2. DHIS2 also has a built in import/export format by which data as well as metadata can be exported. This is called the DHIS eXchange Format (DXF) and is used to exchange data and metadata between DHIS2 instances, but also other systems can send aggregate data using DXF.

Table 7: DHIS2 vocabulary management for HIXen

Metadata name	Metadata description
uid	The globally unique identifier for a concept, but not identifiable without implementation id
name	A unique name required to create a DE
shortname	A unique name (short length) for a DE
code	An optional attribute to give more specific meaning to a DE
description	Description about the DE
value type	The data type of the value (number, text, yes-no, yes-only, date). Number and text have additional properties for number type (decimal, positive, negative etc.) and text type (normal or longtext) respectively.
URL	A URL to uniquely describe the DE
Category Combination	A set of possible fixed values for a DE

While there have been numerous attempts to integrate these two systems, none of them have been flexible enough to be used in different situations. There are different reasons why these have been less flexible or specific to an integration exercise, these are outside the purview of this paper. Some of the integration attempts are described below

Generating compatible datasets on one-side

Since two systems want to communicate with each other, the systems could translate their data into formats that are understood by the other system and sent for importing. This is probably the simplest form in which data can be exchanged between systems. Thus, if OpenMRS wants to send data to a monthly dataset in DHIS2, it will generate a monthly report about the patients for the required period using its reporting module, convert this into DXF and DHIS2 would import DXF into its systems. Some mobile solutions have tried to import data into DHIS2 in similar fashion [21]. For such integration to take place, the patient-level system needs to understand each of the DE inside DHIS2 and be able to generate the value for the DE based on its monthly patient's records. Thus, if there are some formulas to generate the value for a DE, those have to be created by the user manually and mapped to the exact element ID of the DHIS2 system. This kind of integration is simple at first, but very specific to the instances of the two systems. Also if there are changes to the DEs or to the datasets or to the concepts inside OpenMRS, the user again has to make those changes and perform the mapping. Thus, such kind of integration is time-consuming, inflexible and prone to errors.

Handshaking before data exchange

In this type of integration, the two systems first communicate before sending to verifying if they have the same metadata as has been mapped in the previous occasion. Although this ensures that any changes to metadata from either system are checked, manual mapping and understanding of the metadata on both the sides need to be done the first time. Thus, the formulas for aggregation and timing for aggregation needs to be known by the systems beforehand. In an HTTP-style communication, this would mean that any system that wants to send data to the other system will do a GET on the metadata, verify that it has the same copy of the metadata (or meaning of the metadata) and after it has matched, send data to the other system. Each system does this matching independently and whoever wants to send data ensures that the other system is able to understand the data. Although this type of integration solves the problem of metadata mis-match, it does not solve the problem of semantic integration, where the user has to manually ensure that the meaning of the metadata, DE and Concept in our case has not changed. The user has to

manually intervene each time such a vocabulary change has happened and needs to update the other system, when a change in one system happens.

Shared metadata repository and standard data format

One of the latest methods for integration has been to start with a common, shared metadata repository to which integrating systems sign up, map their own vocabulary and share information with other systems. There has been a global focus from healthcare agencies that are pushing forward creation of such metadata registries/repositories [22]. The WHO Indicator Metadata Registry (IMR) is an example of such global efforts to create a shared repository. Applications that want to exchange information match their metadata and make a common understanding of the vocabulary. Although a common vocabulary is shared, there are still problems in sending data in a format that is understandable by the other systems. Here the standardization of data exchange formats plays a key role. There have been some recent efforts in this matter. The Statistical Data and Metadata Exchange – Health Domain (SDMX-HD) standard is useful to send data in an XML format that is self-descriptive and shares data and metadata with other systems. Systems that want to share data can refer to IMR and understand SDMX-HD format. Although this sounds easy, the challenges to maintain a common repository of medical terminologies to cover all kinds of information is a momentous task. The argument towards doing this momentous task is that the upfront capital cost of this kind of work is large, but the result will be a network effect, where smaller systems that want to integrate with other systems do not have to deal with the problem of vocabulary mismatch. Another challenge for software systems that want to communicate using a shared repository is that they have to be abreast with the changes in the IMR and map their custom vocabularies to the IMR. This is somewhat of a small, yet continuous challenge for small systems and systems where resources are scarce.

```
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:dc="http://hixen.org/dc/elements/1.1/"
  <rdf:Description rdf:about="http://myimpldomain.org/dataElement/sfgware4344">
    <dc:name>No. of ANC First Visit</dc:name>
    <dc:publisher>MyImplementation</dc:publisher>
    <dc:dataMembers>
      <dc:concept name="ANC">http://myimpldomain.org/concept/fsdfs32dfs</dc:concept>
      <dc:concept name="First Visit">http://myimpldomain.org/concept/gfs435234ve</dc:concept>
      <dc:concept name="Period">http://myimpldomain.org/concept/gfs435234ve</dc:concept>
    </dc:dataMembers>
  </rdf:Description>
</rdf:RDF>
```

Fig. 18: DHIS2 transactional RDF

HIXen Conceptual Design & Architecture

HIXen is an integration engine that sits in the architecture between any applications that want to integrate. The engine can be embedded in a data warehouse application or it can be part of the patient-level system. Although Health Information Exchanges have been created [23] and also share health information at a large-scale [24], these have been focused towards a single-level information exchange. i.e. patient-level systems sending data to other patient-level system [25] or aggregate systems sending data to aggregate systems or data warehouses. HIXen is an engine that can integrate data between patient-level systems as well as between aggregate-level systems. This presented design is between a patient-level and an aggregate-level system.

Fig. 21: OpenMRS transactional RDF

```
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:dc="http://hixen.org/dc/elements/1.1/"
  <rdf:Description rdf:about="http://myimpldomain.org/concept/fsdfs32dfs">
    <dc:name>ANC</dc:name>
    <dc:publisher>MyImplementation</dc:publisher>
    <dc:dataMembers>
      <dc:concept name="Pregnancy">http://myimpldomain.org/concept/fsdfs32dfs</dc:concept>
      <dc:concept name="Visit">http://myimpldomain.org/concept/kfdnv34</dc:concept>
      <dc:concept name="States">http://myimpldomain.org/concept/zhs346hdfh</dc:concept>
      <dc:concept name="LMP">http://myimpldomain.org/concept/frq851546sq</dc:concept>
    </dc:dataMembers>
  </rdf:Description>
</rdf:RDF>
```

An RDF generator

The first component of HIXEn is an RDF generator. HIXEn looks metadata of systems and analyzes them into Subject→Predicate→Object interpretation. This is matched through the EAV-model by direct database access or API-level access to the supported systems. There are a few ways to generate RDF from the metadata available in these systems. It would be best if the systems themselves could have generated the RDF and the integration engine can then use SPARQL to search for the appropriate metadata, extract the data for the given period, using this metadata information and then send data to the other systems. Currently, none of the popular health information systems generate RDF for the metadata natively. The FOSS systems DHIS2 and OpenMRS also do not generate RDF as well. This paper provides examples of generated RDF for DHIS2 metadata in fig. 4 and generated RDF for OpenMRS metadata in fig. 5. The

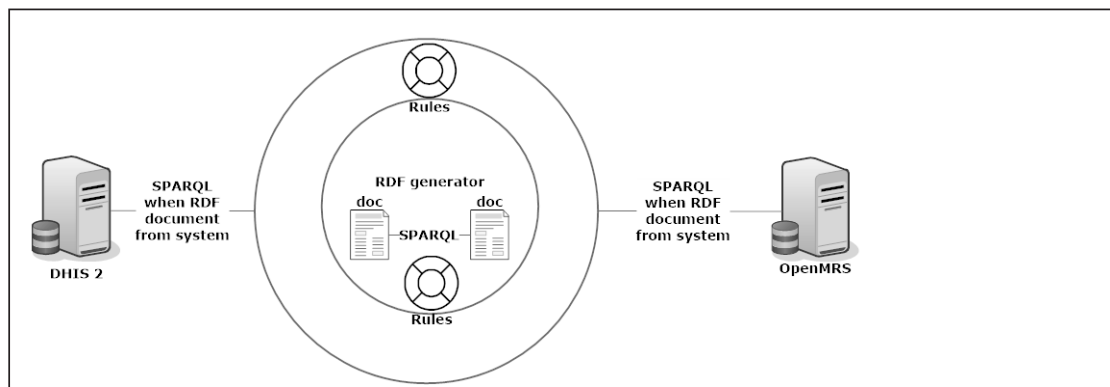


Fig. 24: The HIXEn architecture - Working internals

DHIS2 RDF shown here is an example of how it could be organized for the representation of a dataset and the values of the DEs for that reporting period. There could be similar for the indicators, reports etc. but that much detail is out of the scope of this paper.

The OpenMRS RDF is representation of an Obs and how these are assembled as part of an Encounter. The underlying Concept is a resource that is uniquely represented through its RDF and describes it independently.

Designing rules for RDF Interchange (RIF) and SPARQL

The Rules for mapping between interchanged RDF is done through the Basic Logic Dialect. This makes use of the different profiles of Rule Language such as Document, Base, Prefix, while the Conditions are written using Formula, Atom, Equal and other operators. This combination is mapped between how indicators/data values from DHIS2 and Concepts from OpenMRS. When queries are made by HIXEn, it uses SPARQL and the Rules specified to link the different Resource documents. While it might be obvious once the terminologies are mapped, this might not always be the case and HIXEn might have to parse through multiple layers to rules to reach the actual calculated values to be exchanged between the Resource documents.

Hypermedia used to represent state of the Resources

Every resource has to represent its current state completely through links, so that the RDF can be parsed by HIXEn search when searching for values of the resources. Every link points to a Calculation, Aggregation, Singular Resource or Collection of Resources that can be further traversed. Since every value can be reached using links and the value in the links might change on

every instance, it gives a snapshot at the given point in time. This state is the value of the DE or Concept, depending on which RDF is being traversed by HIXen's SPARQL query.

Conclusion and Further Considerations

Although this provides a flexible approach to integration, every step needs to be implemented in much more detail than described here. Each RDF document itself might vary from system to system and so that the parser in HIXen can understand many documents is a huge challenge that can be met only after more reference systems are available for testing. Performance of the search is another challenge, given that there might be numerous levels for parsing that needs to be done to reach a single Data Value. There are many such Data Values in a single Dataset and this slows down the whole searching through SPARQL. A cache that remembers the last searches and only refreshes when a change even occurs might mitigate the problem, but this needs to be explored further.

References

- [1] R. M. Coffey, J. A. Buck, C. A. Kassed, J. Dilonardo, C. Forhan, W. D. Marder, and R. Vandivort-Warren, "Transforming Mental Health and Substance Abuse Data Systems in the United States," *Psychiatric Services*, vol. 59, no. 11, p. 1257, 2008.
- [2] E. Nyella, "Challenges in Health Information Systems Integration: Zanzibar Experience," *Journal of Health Informatics in Developing Countries*, vol. 5, no. 1, 2011.
- [3] J. J. Cimino, "Vocabulary and health care information technology: state of the art," *Journal of the American Society for Information Science*, vol. 46, no. 10, pp. 777–782, 1995.
- [4] J. Walker, E. Pan, D. Johnston, J. Adler-Milstein, D. W. Bates, and B. Middleton, "The value of health care information exchange and interoperability," *HEALTH AFFAIRS-MILLWOOD VA THEN BETHESDA MA-*, vol. 24, p. 5, 2005.
- [5] D. J. Brailer, "Interoperability: the key to the future health care system," *HEALTH AFFAIRS-MILLWOOD VA THEN BETHESDA MA-*, vol. 24, p. 5, 2005.
- [6] J. Frenk, "Reinventing primary health care: the need for systems integration," *The Lancet*, vol. 374, no. 9684, pp. 170–173, 2009.
- [7] R. Lenz, M. Beyer, and K. A. Kuhn, "Semantic integration in healthcare networks," *Studies in health technology and informatics*, vol. 116, p. 385, 2005.
- [8] S. Garde, P. Knaup, E. J. S. Hovenga, and S. Heard, "Towards Semantic Interoperability for Electronic Health Records–Domain Knowledge Governance for open EHR Archetypes," *Methods of information in medicine*, vol. 46, no. 3, pp. 332–343, 2007.
- [9] S. Sahay, E. Monteiro, and M. Aanestad, "Towards a Political Perspective of Integration in IS Research: the case of Health Information Systems in India," in *9th International Conference on Social Implementation of of Computers in Developing Countries. Sao Paulo, Brazil*, 2007.
- [10] J. Mykkänen, A. Riekkinen, M. Sormunen, H. Karhunen, and P. Laitinen, "Designing web services in health information systems: from process to application level," *Int J Med Inform*, vol. 76, no. 2–3, pp. 89–95, Mar. 2007.
- [11] J. Mykkänen, J. Porrasmaa, M. Korpela, H. Häkkinen, M. Toivanen, M. Tuomainen, K. Häyrynen, and J. Rannanheimo, "Integration models in health information systems: experiences from the PlugIT project," *Stud Health Technol Inform*, vol. 107, no. Pt 2, pp. 1219–1222, 2004.
- [12] R. T. Fielding, "Architectural styles and the design of network-based software architectures," University of California, 2000.
- [13] O. Lassila, R. R. Swick, and others, "Resource description framework (RDF) model and syntax specification," 1998.
- [14] P. M. Nadkarni and C. Brandt, "Data Extraction and Ad Hoc Query of an Entity—Attribute—Value Database," *Journal of the American Medical Informatics Association : JAMIA*, vol. 5, no. 6, p. 511, Dec. 1998.
- [15] B. W. Mamlin, P. G. Biondich, B. A. Wolfe, H. Fraser, D. Jazayeri, C. Allen, J. Miranda, and W. M. Tierney, "Cooking Up An Open Source EMR For Developing Countries: OpenMRS – A Recipe For Successful Collaboration," *AMIA Annu Symp Proc*, vol. 2006, pp. 529–533, 2006.
- [16] N. Boyce, "The Lancet Technology: August, 2011," *The Lancet*, vol. 378, no. 9790, p. 475, Aug. 2011.
- [17] OpenMRS, "Data Model (OpenMRS Implementer Meeting 2010)" [Online] Available: [https://wiki.openmrs.org/display/RES/Data+Model+\(OpenMRS+Implementer+Meeting+2010\)](https://wiki.openmrs.org/display/RES/Data+Model+(OpenMRS+Implementer+Meeting+2010))
- [18] DHIS, "District Health Information System v2" [Online] Available: <http://www.dhis2.org>
- [19] P. C. Webster, "The rise of open-source electronic health records," *The Lancet*, vol. 377, no. 9778, pp. 1641–1642, May 2011.
- [20] J. Braa, O. Hanseth, A. Heywood, W. Mohammed, and V. Shaw, "Developing health information systems in developing countries: the flexible standards strategy," *MIS Quarterly*, vol. 31, no. 2, pp. 381–402, 2007.

- [21] K. Braa and S. Purkayastha, "Sustainable mobile information infrastructures in low resource settings.," *Studies in health technology and informatics*, vol. 157, p. 127, 2010.
- [22] M. Chan, M. Kazatchkine, J. Lob-Levyt, T. Obaid, J. Schweizer, M. Sidibe, A. Veneman, and T. Yamada, "Meeting the demand for results and accountability: a call for action on health data from eight global health agencies," *PLoS Medicine*, vol. 7, no. 1, p. e1000223, 2010.
- [23] J. Halamka, M. Aranow, C. Ascenzo, D. Bates, G. Debor, J. Glaser, A. Goroll, J. Stowe, M. Tripathi, and G. Vineyard, "Health Care IT Collaboration in Massachusetts: The Experience of Creating Regional Connectivity," *Journal of the American Medical Informatics Association*, vol. 12, no. 6, pp. 596–601, Nov. 2005.
- [24] C. J. McDonald, J. M. Overhage, M. Barnes, G. Schadow, L. Blevins, P. R. Dexter, and B. Mamlin, "The Indiana Network For Patient Care: A Working Local Health Information Infrastructure," *Health Aff*, vol. 24, no. 5, pp. 1214–1220, Sep. 2005.
- [25] D. Brailer, N. Augustinos, L. Evans, and S. Karp, "Moving toward electronic health information exchange: interim report on the Santa Barbara County Data Exchange," *California HealthCare Foundation*, 2003

Appendix C: PLS output for Paper 5

