

Distributed Development to Enable User Participation

Multilevel design in the HISP network

Ola Hodne Titlestad

Department of Informatics, University of Oslo, Norway
olati@ifi.uio.no

Knut Staring

Department of Informatics, University of Oslo, Norway
knutst@ifi.uio.no

Jørn Braa

Department of Informatics, University of Oslo, Norway
jbraa@ifi.uio.no

Abstract. Through the study of a long term, globally targeted effort to design health information systems in the Global South, we explore challenges to distributed participation within and across countries, and describe efforts at addressing these. Networked action research projects can enable pooling of resources, skills, best practices and tools, and cross-country collaboration does not have to preclude local ownership, as illustrated by the case material in this article. We highlight specifically the need for circulation of people, artefacts, and standards, to both support local practices and foster the capacity of all stakeholders to take active part in the design and implementation of information systems. The deep effects of global technological change call for a multilevel approach bridging local implementations with global research and participatory design efforts and co-evolution of standardised tools.

Key words: Distributed participatory design, boundary spanning, scaffolding, innofusion, Global South

1 Introduction

This article addresses the issue of participatory design (PD) of information systems in resource-constrained settings, with a particular focus on the diverse public health care sector in the Global South. The insights from the PD tradition are widely recognised, and have become an increasingly mainstream part of information system development methodology in rich countries. However, their potential has yet to be fully exploited in regions such as Africa and Asia, where design-reality gaps remain a challenge to computerisation attempts (Avgerou 2002; Heeks 2006). Examining the case of a long term effort to design health information systems across a number of countries, we discuss the scope for PD approaches in networks that are distributed along several dimensions.

In the South, the lack of funds and skilled people underscores the importance of being able to draw on inputs from similar settings, as participation in design is costly. Key challenges thus relate to developing means to leverage achievements and replicating participatory learning processes in new places. Therefore, Distributed Participatory Design (DPD) becomes an important goal – both for empowering developer and user communities in various countries, and for building reusable tools. The spread of the internet is a major factor in enabling PD across space and time (Gumm 2006a). The aim of this article is to explore what DPD can entail in a network of South-South-North collaboration.

There is a strong need for consolidation, integration and collaboration in health information systems globally (AbouZahr and Boerma 2005; Tierney et al. 2008). The HIV/AIDS pandemic and initiatives such as the UN Millennium Development Goals (MDG) have led to prioritisation of the complicated task of improving health services and information flow, and have increased donor support for many health programs (such as TB, malaria, vaccination programs, maternal health care, etc.). Unfortunately, donors often introduce bespoke systems that sidestep established reporting routines. The resulting complexity, coupled with meagre human resources and infrastructure, easily leads to fragmentation and poor data quality. The launch of the Health Metrics Network (HMN) in partnership with the World Health Organisation (WHO) in 2005 was an important step towards promoting standards. A key feature of the HMN Technical Framework is integration of essential data in a shared repository.

Such emphasis on integration and standardisation is quite removed from the original focus of participatory design projects. The PD tradition grew from a number of Scandinavian working life action research projects in the 1970's and early 80's. In particular, union based projects focused on empowerment of workers to:

- Improve their capacity to negotiate the introduction of new technology with management (for example, a project for the Norwegian Iron and Metal workers' union; Nygaard 1977; Sandberg 1979), and
- Foster participatory design of alternative computer applications with the workers' skills and interests in focus (for example, the UTOPIA project, which aimed to create good tools for typographers; Bødker et al 1987).

While the direct outcomes from these projects were relatively insignificant, they did have an effect at the political level, and gave rise to new legislation regulating the introduction of new

technology (Bjerknes and Bratteteig 1995). Even so, the early PD projects had little impact outside singular research sites, and therefore also failed to become sustainable. Summarizing these experiments, Engelstad and Gustavsen (1993) suggested that the action research community needed to move beyond isolated efforts; focus should be shifted “from single organisations and work places [...] to networks” (ibid., p. 219). Adding to this the globalisation afforded since then by the Internet makes it interesting to examine approaches to user involvement in design in distributed networks (Obendorf et al this issue).

The Health Information Systems Programme (HISP) is a multinational research and development network aiming at developing information systems to strengthen local health services in the Global South. Explicitly rooted in the PD tradition (e.g., receiving the Artful Integrator’s Award at the Participatory Design Conference in 2006 for action research and design across country and community contexts), the project has increasingly imbibed lessons from the distributed software development (DSD) literature. HISP was initiated in South Africa in 1994 by researchers from local and Norwegian universities, and health activists from the anti-apartheid struggle. The project was part of the new ANC government’s Reconstruction and Development Program (ANC 1994a), and began as a participatory system design project in three pilot districts, aimed at supporting newly defined health districts (ANC 1994b). The project had two major elements: 1) definition of an integrated essential data set, including data from the full range of health programs and services, to be collected from all health facilities, and 2) development of a software application called the District Health Information Software (DHIS) to manage the data being collected and integrate with other data sources such as population census. In the period 1999 - 2001, use of the DHIS application gradually spread, leading it to become an official national system in South Africa (Braa and Hedberg 2002).

After the turn of the millennium, HISP activities spread to other countries in Africa and Asia. Today, HISP comprises a global network of universities, health authorities and NGOs from around 15 countries in the Global South. Through persistent efforts, the network involved in design and development of the software also slowly expanded from a two-man team in South Africa to a global team distributed across 7-8 countries in Africa, Asia, and Scandinavia. Distribution and coordination of activities within HISP have been enabled by an international PhD program with more than 25 students from Africa and Asia and the establishment of Masters programs in health informatics in several countries (Tanzania, Mozambique, Malawi, Ethiopia, Sri Lanka) funded by the Norwegian government, where students conduct research within the network and circulate between countries, enabling the sharing of experiences and best practices.

Organisation of the article: The next section reviews the literature on participatory design in distributed contexts and related concepts, followed by a section outlining the methodology. Section four details the increasing distribution of design and development activities in the HISP project. This is followed by a discussion where we draw on the case study to describe a global approach to participation across boundaries. The last section sums up our contributions.

2 Literature and background

This section discusses the concept of participatory design in relation to recent trends in the direction of widely dispersed software development and deployment of standard information systems in diverse settings. To analyze the role of participatory design in a globalized world, we introduce concepts around learning and innovation across boundaries with the help of standards.

2.1 DPD, DSD and standards

There has been a marked growth in standards based, *generic* information systems, designed for use throughout whole industry sectors, such as health care, or even across sectors, such as ERP systems (Pollock et al. 2007). Thus, there is a need for approaches to distributed systems development that both ensure coherence and can support diverse local settings, striking a balance between global standards and local needs (Rolland and Monteiro 2002). To be able to develop global software systems, modularisation and generification strategies become paramount, and this is reflected in both organisational arrangements and software architecture (Staring and Titlestad 2008).

The diversity of distributed use settings should be distinguished analytically from the distribution of software developers. The latter is explored in the literature on distributed software development, which discusses challenges in terms of coordination and sharing of knowledge beyond those faced by collocated teams (Lings et al. 2006). This is particularly relevant when developers are spread across cultures and time zones, as is common in offshore outsourcing (Sahay et al. 2003). Offshoring arrangements are usually seen mostly as cost-saving measures, but as pointed out by the Open Innovation literature (Chesbrough 2003; West and Gallagher 2006), the preponderance of ingenuity in the world is distributed outside of any particular organisation, and one should seek to harness it. This would seem to be all the more the case when potential users are also widely dispersed.

1 Have Clear Distribution Rationale	6 Manage Processes
2 Clarify All Understandings	7 Develop a Sense of “Teamness”
3 Leverage Modularity	8 Encourage Temporary Collocation
4 Use Cultural Mediation	9 Encompass Heterogeneity
5 Facilitate Human Communication	10 Develop an Effective Tool Base

Table 1: Ten strategies for DSD, from Lings et al. (2006)

The distribution of developers geographically introduces problems of mutual understanding which are very similar to issues that PD approaches seek to address, and Gumm (2006a) rightly asks whether the concept of DPD embodies a contradiction in terms. Furthermore, whereas DSD has so far been studied mostly in commercial settings, PD is often associated with research or public sector efforts. In our view, this makes it even more pertinent to compare the two bodies of research. For example, it is interesting to note that the strategies suggested for successful DSD (see Table 1) are quite similar to PD methods for bridging the gap between users and

developers, for example items 2, 5, 8 and 9. In other words, DSD research provides methods to alleviate many of the challenges increased distribution poses, which can help make DPD viable. Similarly, Loebbecke and Powel (2009) analyse links to action research and design science.

2.2 PD in the Global South and learning

Over time, the focus of much PD research has shifted in a more pragmatic direction (Greenbaum and Kyng 1991). A key PD principle is to bridge and blur the user-designer distinction from both directions, through mutual learning processes. By this approach, users must be able to engage with the artefact under design, and designers should build a thorough understanding of the life-world(s) of the users, to have a more complete perspective of the system in use (Bjerknes and Bratteteig 1995). Effective methods to achieve this usually rely on prototyping and intensive face-to-face interaction between users and designers, such as close observation of the work practices of users, joint workshops, and scenarios (Kensing 2003). In the Global South, computerised information systems are still few and far between, and potential users in the health care sector have no or extremely limited prior exposure to them. This amounts to a significant threshold hindering participation, and a visual, interactive prototype is essential to overcome such barriers and enable meaningful reciprocal learning between designers and users (Kimaro and Titlestad 2005). Such prototypes become significant boundary objects shared between problem solving contexts (Star 1989). In new product development, boundary objects help establish a “boundary infrastructure” used to manage knowledge across a given boundary (Carlile 2002).

Mutual design processes are therefore likely to initially have characteristics of exploration and struggle to make things work, captured in the related concepts of innofusion (Fleck 1988) and learning by trying (Fleck 1994). Complex artefacts such as organisational technologies are likely to require processes of experimentation in the contexts in which they are implemented. They may need to be taken apart, broken down, adapted and reconfigured, and sometimes the technology may be more or less completely re-invented in its implementation and use. Repeated reconfiguration of components into systems that fit particular contexts thus blurs the line between innovation and diffusion, leading to the notion of innofusion (Fleck 1988).

Similarly, learning by trying refers to “the knowledge created during innofusion and may occasionally amount to significant changes in technological knowledge bases” (Peine 2008, p. 5). Learning occurs “due to an initial misfit between product characteristics and its use environment that leaves room for improvements through the operating experience” (ibid). According to Peine, if the initial pioneering phase of radical change results in a working product, the process becomes one of *learning by using*, with smaller, progressive modifications to the functioning system. Once the product stabilises and there is less scope for further improvement, learning takes on more of a character of on-the-job training, referred to as *learning by doing*. However, in the Global South, with sparse technological infrastructure and knowledge base, this clean sequence becomes blurred, with the categories of learning occurring in parallel: familiarisation with technology and design of an appropriate system of necessity become intertwined and take place simultaneously.

Furthermore, when considering the Scandinavian PD lessons in the context of the Global South, an important difference is that the typical arena for movements for social change in de-

veloping countries is the political ‘grassroots’, rather than the workplace (Braa 1996). Therefore, a community based approach is recommended (Byrne and Sahay 2003). As the user-designer gap in knowledge of and experience with IT solutions is often wider in the Global South than in Scandinavian settings, PD approaches aiming at fostering mutual learning are actually even more appropriate and relevant in such contexts (Braa et al. 1993).

The concept of inno-fusion is particularly apt when we make the transition from bespoke systems for a single site to more distributed information systems. In order to cope with the variety of present and future contexts, Fischer (2008) introduces the concept of *meta-design*, meaning design that aims to empower users to engage actively in continuous, iterative development, rather than being restricted to the use of existing systems. Meta-design strategies such as flexible standards, modularisation and black-boxing can facilitate generification processes (Braa et al. 2007). The meta-design approach strives at creating not only a flexible technical basis for design, but also social infrastructures in which users can participate actively as co-designers to shape and reshape socio-technical systems.

2.3 Boundary spanning and scaffolding

Successful examples of global scale DSD can be found when studying the phenomenon of Free and Open Source Software (FOSS), which is usually characterised by a communitarian, asynchronous and mediated design practice where both users and developers are widely dispersed (Barcellini et al. 2008). FOSS projects typically blur the separation of roles through transparent processes and open communication and intensive use of online media (Fogel 2005; Lanzara and Morner 2005). Indeed, the FOSS approach can be considered as a continuous form of open ended distributed participatory design, where new functionalities can always be proposed by different kinds of participants, regardless of their stake in the project, and users can potentially be involved in all phases of the development process (Barcellini et al 2008; von Hippel 2005). In order to create successful organisational systems, design teams should include both software architects and domain experts. Thus, mutual learning processes are crucial, though difficult to accomplish (Gumm 2006a). A successful FOSS project will consist of people filling different roles (sometimes simultaneously), such as project leader, administrator, developer, and user, with varying levels of engagement. In particular, the FOSS literature stresses the role of active users who participate by assisting newcomers, reporting bugs, or proposing new features (Fogel 2005; Preece et al. 2004).

Prominent ways to cope with the geographical and organisational distribution and diversity of roles in FOSS projects include boundary spanners and scaffolding. Boundary spanners are persons who act as mediators, traversing boundaries between organisations and teams, enhancing informal communication across networks (Sarant 2004; Sonnenwald 1996). “Becoming boundary spanners implies having developed skills and competencies in the different fields that are spanned. Boundary spanners are well aware of all practices and have achieved legitimacy and credibility in the domains they span” (Barcellini et al. 2008:560). The multiple dimensions of distribution in FOSS projects (and in DPD more generally) mean that many boundaries need to be bridged, and this implies a strong need for developers, implementers and coordinators who can span organisational contexts and geographical boundaries. Whereas with infrastructural

software like databases or operating systems, developers are usually also users, the differences in perspective is larger for organisational software, and thus the need for boundary spanners all the greater. The gap can best be bridged by technically conversant people who engage in implementing the system and training end-users, and who are also adept at communicating with the core developer team. More than just super-users, such mediators normally work closely with users to make systems work and therefore understand their problems, but also know at least some of the developers by name and can relate to them (Finck et al 2004).

Supplementing and enabling the activities of boundary spanners are material means of mediation and support for the design process. Seeing parallels between design and learning, we find Orlikowski's (2006) use of the concept of scaffolding useful for understanding how DPD can succeed. A scaffold is not part of the final product, but serves temporary support functions. Scaffolds are flexible and can be set up according to local conditions, and are also portable, i.e. they can quickly be assembled and modified in different places. Thus the metaphor helps us focus on the "temporary and situated engagement of technology in knowledgeable activity. That is, for the duration of a particular human practice, actors draw on various artefacts, spaces, and infrastructures to conduct their activities" (ibid., p. 462), or what Orlikowski terms the 'scaffolding of knowledgeability'. Moreover, scaffolds are generative, in the sense that they serve as a "basis for other (creative) work, facilitating the performance of activities that would have been impractical without material augmentation" (ibid., p. 462).

Thus, the challenges of DPD can be addressed through the combination of a network of collaborators formed around boundary spanners and scaffolding infrastructure such as training material or the typical technical artefacts employed by FOSS communities (see section 3). According to O'Reilly (2003), the truly distinguishing characteristic that FOSS offers is not the availability of the source code, but an overall "architecture of participation", both material and social. The experiences from the case presented below suggest the contours of such an architecture for the Global South.

3 Method

Our empirical investigations are grounded in the Scandinavian action research tradition, which emphasises engagement in the field. A long-term distributed action research effort, the approach followed in HISP builds on the literature on networked action research (Elden and Chisholm 1993; Engelstad and Gustavsen 1993), and has elsewhere been termed *Networks of Action* (Braa et al. 2004a). In essence, it is based on the premise that action should not take place in isolated units, but only as part of a greater community where sharing of experiences, learning processes, and support are facilitated. The Networks of Action approach is specifically designed for the resource limited conditions in the Global South. The 15 years time-span of HISP exceeds traditional 'projects' and is more akin to social movements or programmes (Elden and Chisholm 1993).

All three authors are actively engaged in the Health Information Systems Programme on a daily basis. One of the authors was the originator of the project in the mid-1990's, and the two other authors have been engaged in the action research project for about five years, working on

software development, design and implementation, as well as education activities. Since 2004, this group has lead efforts to distribute participatory processes of development and design of version 2 of the software (DHIS v2). More specifically, we have engaged in the management of developers in the Norwegian node, facilitating collaboration between nodes, guiding PD processes in local implementation projects, and building design capacity in various countries.

The ongoing activities started more than a decade ago. Such a long-term action research project inevitably gives rise to various data sources, both formal and informal, involving multiple types of data and methods of collection. It is therefore difficult to provide extensive quantitative details of the number and timing of interviews and repeat interviews conducted. However, the project has produced vast amounts data from primary and secondary sources. In line with FOSS best practices (Fogel, 2005), extensive use of open communication tools such as mailing lists, issue trackers, and wikis means that much of the informal communication in the project is publicly accessible (von Krogh and Spaeth 2007).

Primary data sources include notes form field observations and discussions with health workers, administrators, technical staff and local developers as we engaged in project implementation. Secondary sources include journal papers, conference presentations, PhD and masters theses (many supervised by us), and various workshops and seminars. Also, since the project received funding from a variety of donors, in particular through the European Commission's 6th Framework Program, extensive project reports have been developed, with detailed description of activities and progress, supplemented by evaluation reports, funding proposals and notes from meetings involving managers and staff. These data sources are listed in Table 2.

<i>Academic</i>	Theses, dissertations, papers, conference presentations
<i>Project related</i>	Funding proposals, progress reports, evaluations, presentations to stakeholders
<i>Communication tools</i>	Mailing lists, wiki, issue tracker, source code commit messages, instant messaging meeting records
<i>Field</i>	Meeting notes, user observations, developer workshops

Table 2: Data source categories

The empirical focus of this paper is on the trajectory of participatory design in the project, from its beginnings following a traditional PD approach in a few districts in South Africa, to an increasingly globally dispersed processes of design and implementation taking place around the network today. Taking Gumm's (2006b) four axes of distribution (spatial, temporal, organisational, and stakeholder related) as an organizing principle, we identified a number of important distributed interactions impacting on the design, both of the software and of the overall process, including implementation. These interactions were then grouped according to similarities in how they were affected by distribution. Surveying these groups, the following main themes emerged:

- Design taking place at multiple levels, and the interaction between local and global design processes

- The organisation and balancing of teams and roles, and their effect on both traditional PD and DPD
- The role of DSD to foster DPD

These grouped themes form the structure of the case description in the next section.

4 Case – distribution in the HISP network

The Health Information Systems Programme (HISP) spans a 15 year period and represents a reservoir of field stories across a wide variety of research topics and country contexts. The project has undergone major transformations from being a small district pilot to becoming a recognised actor on the global arena of health information systems for the South. This case study focuses on the trajectory of PD in HISP, particularly during the shift from a locally focused to a globally distributed project.

We start by presenting a condensed timeline of design activities in HISP (4.1), followed by a description of what exactly is being designed (4.2) in this project. With this background, we discuss how various constellations of teams and roles both strengthen and impede local and global design activities (4.3), and finally describe how the challenges of global software design have been met in and across the various local contexts (4.4) in HISP.

4.1 Timeline of HISP design

The first phase of DHIS development can be characterised as an intensive three year evolutionary PD process which took the system from a district pilot to a country-wide standard for Health Information Systems (HIS) in South Africa (Braa and Hedberg 2002). These efforts were carried out in line with PD practices, and a series of increasingly refined prototypes were tested in close collaboration with users, to enable information for local action. To some extent the prevailing post-apartheid reform goals of decentralisation and local empowerment were “inscribed” into the software. While the iterative design process produced a close fit with the needs for sweeping reform, the system accumulated both rigidities and a messy architecture overall. This proved problematic when it was subsequently introduced in countries such as Mozambique, India, Vietnam, and Cuba after the turn of the millennium. To address this situation, the project embarked on a completely revised and internationalised version of the software in 2004, starting from a full remodelling of the database. The developer team was still confined to Cape Town, and employed the same technology (MS Access), but this time the users were primarily elsewhere (South Africa kept the existing installations, which were stable and well adapted to the local situation). Through extensive travelling of project staff, supplemented by e-mail communication, the new internationalised DHIS version was developed as a participatory process between the developers in Cape Town and implementation teams in Botswana and Zanzibar (Sheikh and Titlestad 2008).

As the new, internationalised v1 was introduced to new countries, the two-person development team became a bottleneck to supporting the expanding network of users with specific re-

quirements. While the technology enabled rapid prototyping, the v1 architecture was not suited for distributed development, and because of the small and co-located team, no source code sharing tools had been employed. Another shortcoming of the existing architecture (called v1) was the dependence on the Microsoft platform, which meant that even though DHIS v1 itself was open source, it required the full MS Windows and MS Office stack to run. These factors triggered yet another revamp of the software. Development of v2 began in 2004 under leadership of the University of Oslo, but aimed at distributing development activities to a number of the countries in the network, in order to bring software development closer to the contexts of use.

A stack of “bleeding edge” Java-based technologies was selected for v2, and in parallel a distributed development platform similar to those employed by many FOSS projects was set up. However, re-implementing DHIS as a modular web application proved quite difficult. Also, the radical break in technologies as well as an over-emphasis on the new online communication platform presented a formidable obstacle to the involvement of existing technical staff in various sites. The new flexible but complex architecture in effect hindered PD efforts, as it took over a year and a half before v2 could initially be deployed, first in India, and even then much important functionality was lacking. The system improved significantly through early use in India and Vietnam and later also in Sierra Leone, as well as through the involvement of new software developers recruited locally. While engaging with the global source code, their main task was to support local implementations, in the process bridging the divide between users and developers.

After the first pilot in Kerala in the beginning of 2006, use of v2 spread greatly in India, establishing the credibility of some of the HISP team members to participate in a national HIS reform process. As a consequence, v2 is planned for installation at the sub-district level all over the country. The maturation of v2 made it a candidate for inclusion in the Health Metrics Network (HMN) initiative to implement an integrated HIS in Sierra Leone and also meant that it became part of a WHO recommended stack of FOSS tools for public health.

Table 3 summarises different stages of distribution of developers and user settings.

<i>Timeline</i>	<i>Stage</i>	<i>Use and Development</i>
<i>1994-1999</i>	Pilot and national system	Users and collocated software developers, all in South Africa, network of users
<i>2000-2004</i>	Expansion	Multiple countries, core development isolated from local modifications
<i>2004-2007</i>	Technological transition	Two branches (v1 & v2), infrastructure for sharing, but fragmented processes, isolated modifications
<i>2008-></i>	Integration	Multiple local teams, travelling, local developments contributing to global software

Table 3: Different stages of distribution of developers and user settings.

Sub-sections 4.3 and 4.4 describe in more detail how the organisation of both people and technologies was decisive in the design of DHIS during the stages depicted above.

4.2 Global tools and in-country system design

Design of in-country systems – example South Africa

As the existing South African HIS was extremely fragmented, with poor data quality and lacking standards, the first objective was to develop shared essential definitions of data to be collected and key indicators to be calculated across the country. The initial development of a minimum data set in Western Cape Province and software to handle it (DHIS) took place in close collaboration with end users and health professionals. Through a gradual process, several features were added which to some extent inscribed reform aims and best practices into the software; 1) the new health policies with a focus on decentralisation and local management and 2) information for local action approach to public health. The breakthrough came in 1997/98 when first one province, then a second, implemented new minimum data sets using the DHIS.

The experiences from South Africa have been instrumental for HISP in the development of participatory approaches to identify and to support information needs within national health systems. In each country, the more concrete design outputs needed for a well-functioning system include standardised and clearly defined data sets and indicators, good data flow routines, data validation, and various standard formats (tables, charts, maps etc.) for data analysis, presentation and feedback for each organisational level. While the standardisation of health information has made progress at the global level, there are still pervasive contextual differences between and within countries, making in-country design processes essential to the implementation of locally relevant information systems.

A shared toolbox

We use the term global toolbox for systems design to refer to what is being distributed and applied in the various in-country design processes in the global HISP network. The toolbox contains software (the DHIS), best practices and recommended strategies for overall development of in-country systems, as well as training material and other documentation.

DHIS is a generic tool rather than a pre-configured database application, with an open metadata model and a flexible user interface that allow the user to design the contents of a specific information system without the need for programming. This also allows for rapid prototyping: for example, during a short initial visit in Sierra Leone, DHIS v2 was quickly customised with the local organisational hierarchy, and data was imported from various fragmented systems. The implementation team could then present the rapidly assembled system to major stakeholders in the project, visualizing their own data in indicators and charts. By displaying concretely what the final integrated system could provide, both technically and in terms of information, the DHIS served as a powerful interactive tool supporting early discussions.

In a typical implementation sequence, the initial system designed at the national level undergoes a validation process at selected pilot sites, where local managers and health workers play a central role in modifying and further adapting the system to their needs. Being able to communicate through a visual and easily editable tool facilitates local PD processes. For example,

district managers in Sierra Leone became actively engaged in designing additional calculated indicators through the user interface and could quickly see the results in tables and charts, using their recently collected data. In Cuba, the sub-district statisticians played a key role in localizing the organisational hierarchy by replacing “dummy” names such as “Family Doctor Office 1, 2, 3” with meaningful facility names not known at the national level (Braa et al. 2004b). Such design processes were often carried out as part of on-the-job training. In this way, an important requirement in cooperative prototyping was achieved; to bring the current skills of the users in contact with new technological possibilities (Bødker and Grønbaek 1991).

4.3 Teams and roles in HIS design

A number of actors and stakeholders are part of the HISP network; health workers, administrators, public health experts, graduate students, project leaders, and technical staff and trainers in the various sites where the system is introduced. To strengthen the many local implementation teams, both from the technical and medical information domains, the University of Oslo (UiO) introduced graduate programs in health information systems and research projects related to HISP activities. A number of key project members, with either medical or technical backgrounds from Ethiopia, India, Malawi, Mozambique, Tanzania, and Vietnam have so far completed Masters or PhD degrees.

A key strategy in HISP has been to ensure circulation of people across countries, and PhD students, researchers and consultants from many different countries have through their travelling established bilateral links and learning between countries and also helped to both design and disseminate the global toolbox. E.g. in preparation for an upcoming implementation of v2 in Tanzania, a team attended a workshop in India to learn about relevant experiences. Other examples include Ethiopian Masters and PhD students travelling to Tajikistan and India to support v2 implementation, and a Vietnamese Masters student who spent time in Ethiopia to strengthen the local development team there.

The influence of the network on local implementation

When v1 was revamped with global use in mind, the South African development team depended on feedback from Botswana and Zanzibar. Interaction with users in neighbouring Botswana was mainly facilitated by developer field visits, but with the more distant Zanzibar the approach was to communicate through experienced global HISP staff. The process of customising DHIS in Zanzibar revealed a lot of bugs and instability in the new version, causing considerable frustration in the local team, who were simultaneously learning to use the system. From 2005 to 2007, there was an almost permanent presence of HISP members (PhD students and consultants) from Norway, South Africa, and India, both from technical and public health domains. Furthermore, Masters and PhD students from Tanzania mainland (a three hour boat-ride away) who enrolled in Oslo programs did extensive field work in Zanzibar, and participated in the local implementation team. Experienced DHIS implementers from Norway and India had in-depth knowledge of the DHIS, and frequently reported back to the developers in South Africa

(whom they knew well) on problems. The following e-mail exchange illustrates the importance of having technically skilled people in the field who are able to provide constructive feedback to the programming team:

Indian PhD student working in Zanzibar:

Hi, we have some problem in calculated Data element. I tried both in DHIS 1.4.0.37 and 40 but both didnt work. I tried to add some calculated data element and then tried to export to data mart but the field show blank. And also there is no DataelementCalculatedfield table in datamart file. Acutally, i was going through the code but all the steps for export to data mart are about aggregation and indicators and others but not the anything about calculated data element. I dont know whether i am doing any mistake in setting up the dataelement.

South African developer:

There is no procedure to auto-calculate these values in the export to datamart process. It is a process that is currently outstanding but it will be included in build 42.

The Zanzibari implementers were recruited from local technical universities and had no prior experience with either the DHIS or the health information field. They were immersed in an intensive learning by trying process, working under the guidance of the visiting implementers on designing data sets and indicators and customising the slowly maturing software. The local team gradually increased their overall command of the domain and the maintenance of the system, and gained the capacity to conduct extensive training workshops and facilitate on-the-job learning by doing in district offices. Fluent in the local language and culture, they could more easily understand the feedback from local users and engage in often heated design discussions with key stakeholders. Such intensive engagement with users often led to queries or suggested improvements which the implementers then raised with the international team, and subsequently discussed on the global mailing list or directly with the core developers, in effect forming a multi-level chain of mediation. However, as they became more familiar with the core team, the local implementers also communicated directly via email or instant messaging.

Results of user-developer gaps in HIS implementation

When DHIS v2 was first introduced in the field in 2006, the HISP India team members, who had several years of experience with v1, found it hard to make it useable in their context. At the time, no report generation or graphing tools were in place. Additionally, the initial implementation had poor capacity for processing the large amounts of data collected in India, having been tested mostly with small sample data sets, as illustrated by this e-mail from the local team to the Norwegian developers:

We couldn't export data values even after waiting for one complete night after clicking the button export. Even the calculation of 10 indicators for a year it took more than 7 hours which is not acceptable.

The Indian team also complained that they could not find their data, as the database model was difficult to understand. Although the model was quite similar to v1, the new version incorporated a database abstraction layer, inscribing the use of a Java Application Programming Interface (API) instead of directly accessing the database. However, the Indian team found this hard to understand, and chose to bypass the API in order to obtain quick results and fill in the gaps in the global v2 software, e.g., providing "hard-coded" (but well-performing) reports.

In response to immediate user needs, a Dashboard module was developed by the Indian team in collaboration with health managers, with the purpose of enabling graphical analysis of data and indicators in charts. This quickly became an important instrument for the Indian coordinators when demonstrating the capabilities of the system to local authorities. For quite some time, this module remained outside of the global code repository, and due to many local workarounds and hacks, it was not compatible or able to plug into the globally shared code. As such, this became a constant source of friction between the local team and the core developers, who were asked to help out on problems, yet did not want to deviate from the architectural principles they saw as the foundation of the system. The tension eased gradually as the Norwegian team gained a better understanding of the context and pressures facing the Indian team.

4.4 Expanding the developer network - DSD

Over the expansion stage of the project, local requests for additional functionality in v1 peaked, and increased the pressure on the small development team in South Africa. Where skills were available, local additions were developed in-country, as was the case with an Ethiopian ICD (International Classification of Diseases) module and a Mozambican custom data entry module. However, due to the architecture of DHIS v1 and lack of tools to support a distributed development process, these initiatives resulted in "forks", i.e. local versions incompatible with the centrally maintained source code. The isolation caused by such forks hindered appropriation of global level improvements, and neither did the global toolbox benefit from innovations in the locally forked software.

This situation triggered the development of v2, which emphasised distribution of design in order to better utilise the skills around the network as well as meeting local user requirements. In this way, a DSD process was partly initiated because of the needs to support PD in many locations. However, setting up architecture for distributed development while at the same time replicating all the functionality of v1 proved difficult, as illustrated by the initial failure of v2 to meet user needs in India. The expansion of the developer network meant a wider distribution of skills, mitigated by travel, mentoring and enrolment in graduate education programs, while still contributing to the project, both in the home country and globally.

After a long term trial and error process, during which both the coordinators and the network of developers increased their skills and knowledge of the core architecture and processes, many were able to bring local innovations into the global code base, thus forming a true DPD

network designing the global toolbox. Through close interactions with users and stakeholders in the various settings where the DHIS was deployed, several local teams created add-ons to the core application, often quite outside the purview of most of the key developers. The need for synchronisation with the steadily evolving core code base, combined with the potential use of the functionality in other countries, meant that there was pressure to integrate these add-ons with the global repository and “domesticate” them as generic (i.e. not country specific) modules.

For example, the Dashboard module developed in India quickly became a success there, and caught the interest of the Norwegian coordinators as having functionality with a clear potential for more general use. However, the Dashboard reflected a lack of understanding of the v2 architecture and did not follow the intended separation of concerns between global (core) and local modules. A Norwegian developer visiting India spent a significant length of time pair-programming with its developer and committed it to the repository. Shortly after, the new module was used to great effect at meetings with the World Health Organization in Geneva. Later, the Indian developer stayed in Oslo for a couple of months, working with the team and attending the master level course in v2 technologies.

The Ethiopian team faced an important requirement to collect information with a relatively intricate break down by categories of status, sex, and age, and linked to the International Classification of Diseases (ICD). Some years earlier the Ethiopians had created a fork of v1 to accommodate this functionality. This time, a local developer successfully implemented a module in v2 to support the multidimensional data sets, replacing parts of the core. Although this resolved the urgent need in Ethiopia, the solution was incompatible with central parts of the global code, such as the calculation of indicators, and therefore could not be easily shared by other countries. Shortly afterwards, the Ethiopian developer was recruited into the PhD program in Oslo, where he started discussions with the core developers on how to merge it into the global code base. Despite generating several good ideas, a concrete solution remained elusive. However, when the Ethiopian developer visited Tajikistan a few months later in relation to the customisation of v2, he quickly saw the benefit of his multidimensional model for their detailed report requirements, and under severe time pressure, the module was finalised and large parts of the core system were re-factored. The new module was subsequently utilised to its full potential in Sierra Leone, and underwent several improvements and further stabilisation as a joint effort between Ethiopian and Norwegian developers.

5 Discussion

The stages in the HISP case—Pilot and national system, Expansion, Technological transition, and Integration—illustrate several aspects of the relationship between DSD and DPD. We now proceed to analyze the distribution of design activities through the interplay between local and global levels, and the crucial role of implementers and networks.

5.1 Global design through local use

HISP design can be seen as two distinct yet related processes taking place at different levels: 1) in-country design of health information systems with an emphasis on the system rather than on software and 2) across-country design of a globally distributed toolbox of software and best practices for in-country HIS design. In the development model for DHIS v2 as described in the Integration stage, these processes continuously feed into each other; globally distributed solutions grow out of local designs and use, and the global toolbox is utilised in local design processes (see Figure 2). The multidimensional data module was designed in an evolutionary process through iterations of user interaction and prototyping first in Ethiopia, then in Tajikistan and finally in Sierra Leone, and in order to meet the architectural requirements of the global toolbox it went through an adaptation process in Norway. In this way innovation takes place through local use; through cycles of innofusion where the global and local mutually influence each other.

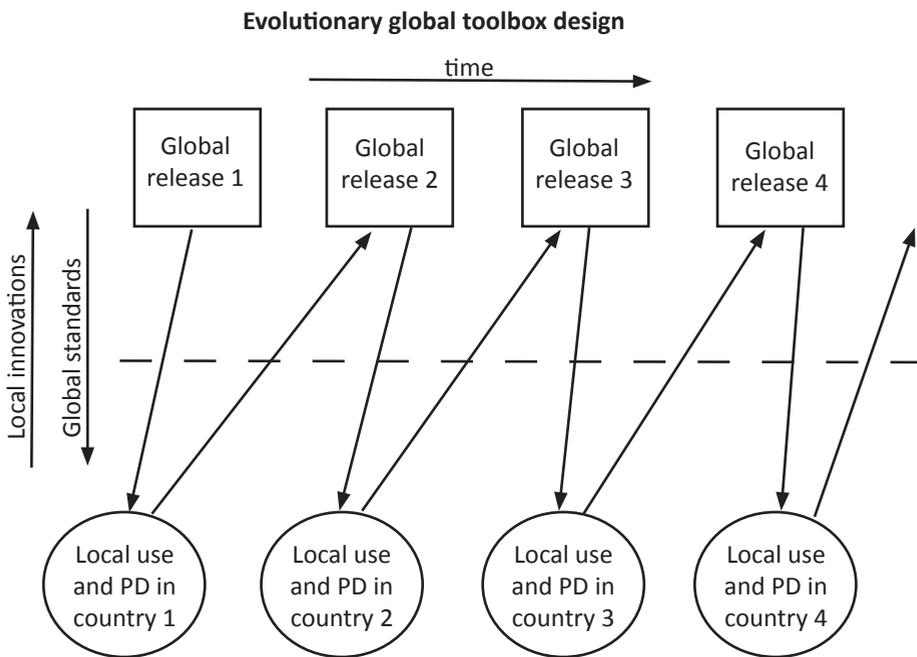


Figure 1: Evolutionary global toolbox design

Maintaining the focus on PD as the network expanded necessitated distributed teams of developers engaging in locally relevant design. In time, and through training and the circulation of people and artefacts, these teams took part in the design of generic modules which had developed from their own or similar contexts, thereby forming a DSD network. While the DSD literature mainly discusses virtual teams and offshore outsourcing for the commercial sector in the US or Europe, the HISP case turns the setting on its head and looks at “outsourcing” of large

parts of development of information systems for the social sector in the Global South, explicitly aiming to train developers in many locations. In fact, HISP tries to apply the DSD techniques listed in section 2, in combination with FOSS approaches to dispersed collaboration.

A key design method to spread local innovations across a heterogeneous network is abstraction or generification (Pollock et al. 2007). The struggle to make the Indian Dashboard module part of the global toolbox illustrates that it takes considerable effort to make the products of local design both relevant and known. To achieve this, the software architecture must take the form of flexible meta-design (Fischer 2008) to balance local and global needs (Rolland and Monteiro 2002). Thus, DPD raises two important issues; 1) the results take the form of standards, if they are to be distributed and shared, and 2) the participants customise the results to suite their own needs. The benefit of standards, combined with the democratic principle of users being able to adapt the system according to their needs, balances the global-local tensions. The standards in question therefore have to be flexible (Braa et al. 2007) in order to facilitate innofusion.

As long as the initial technological transition was mainly confined to Oslo, there was little room for sharing of local improvements (which were derided by the Norwegian developers as “hacks”). However, as the requisite infrastructure and skills gradually evolved through learning by trying (Fleck 1994), local designs became available as material in the global toolbox, in a process of generification. Thus, as Figure 1 illustrates, innofusion in a network creates opportunities for learning by trying designs developed elsewhere. In this way, the sharing of local innovations within the network means that there is extensive scaffolding material (Orlikowski 2006) available to the local design of information systems.

5.2 Coping with distribution

The case study demonstrates that the global toolbox does not travel well across geographical boundaries without being linked to the circulation of people who care about it. The implementation team in Zanzibar with experienced foreign HISP consultants and PhD students co-designing with locally recruited implementers combined best practices from the global toolbox with a thorough understanding of the local context and needs. HISP implementers take the role as boundary spanners to balance global standards with local needs and to bridge the technical and health domains (see Figure 2). Implementers acting as mediators (Finck et al 2004) are involved with extensive negotiations in the redesign of existing paper forms, the determination of data to be collected, and the determination of appropriate indicators to be calculated and reported. The HISP project do not only seek to automate existing processes, but act politically to support the reform of work practices in the direction of effective use of information for action.

Through co-design with more experienced HISP implementers from other countries and long term processes of learning by using, local team members gradually become effective boundary spanners and more active participants in the global network. Furthermore, PhD students from other countries learn important lessons from a new context and from interactions with other implementers in the network.

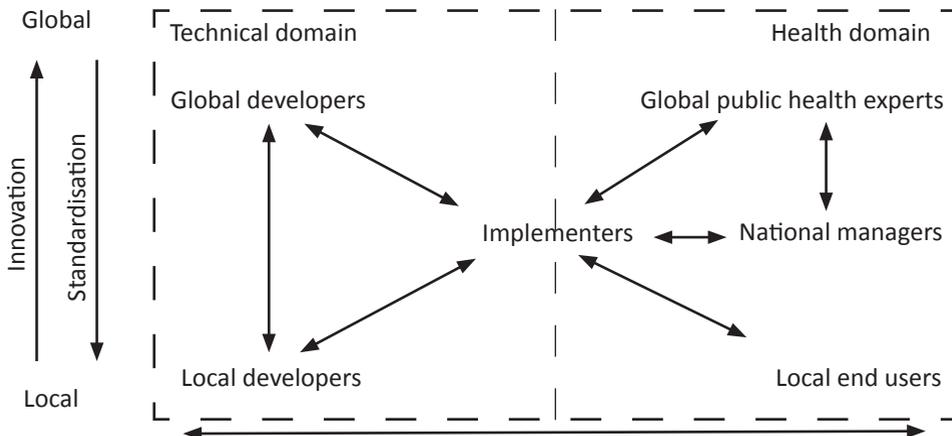


Figure 2: Implementers as mediators

The platform-like character of the software means that it is simultaneously part of an end product which includes reports and local data standards, as well as scaffolding used by boundary spanners in collaboration with local users to make systems function in their work settings. We have seen from the initial phase in Sierra Leone how the flexibility of the platform enabled rapid production of relatively elaborate boundary objects that facilitated communication between users and developers.

The HISP variety of DPD is one where local implementers fill crucial roles as intermediaries with end users. While the project has seen instances of enthusiastic health managers providing excellent feedback to core developers, such direct communication with end users are very much the exception. The South African user mailing list shows some promise in this regard, but strong “champions” and other boundary spanners are needed in each country to enable anything similar to strong FOSS communities (Barcellini et al. 2008). The long term global support to capacity building for local design has been crucial to providing sustainability. In most of the HISP countries implementers were enrolled in graduate programs in health information systems. However, emphasis was placed on maintaining strong links with field activities, and all students were encouraged to link their studies to ongoing design projects, as illustrated i.e. through the long term involvement of Tanzanian graduates in the Zanzibar project.

5.3 The case for networked PD

While acknowledging the very real obstacles to PD introduced by increased distribution, we believe that the HISP case shows that Engelstad and Gustavsen’s (1993) call for networked action research is relevant to DPD, and this emphasis on networking is also very much part of FOSS best practices (Fogel 2005). Thus, one has to find ways of coping with use in a growing number of contexts (Obendorf et al, 2009). The benefits of networking and distribution can be explored along two dimensions; one political and one practical, which share common needs for standardisation.

The political dimension may be seen in the continuation of the Scandinavian projects from the 1970's and 80's. These projects addressed a key political issue at the time: the threat felt by workers from the introduction of new technologies. Similarly, HISP is addressing the "hot" issue regarding equity of health services in the Global South. Also, there was increasing global focus on reliable health data and practical ways to design HIS. Sustainable development of HISP has repeatedly involved bottom-up distribution and networking to gain significant traction. Once a sufficiently broad basic coverage was established, a kind of snowballing effect attracted more top-down interest from political (and administrative) quarters, as illustrated in the three examples below.

First, the momentum created by results (and good timing, as other efforts faltered) enabled gradually widened distribution of use and design in South Africa and a spread of HISP activities from three districts to the provincial level, then to more provinces, and finally to recognition at the ministerial level and approval as a national standard. Second, the expansion in India from a pilot district to state-wide rollouts in several regions of the country established the credibility of HISP India team members to participate in a national reform process. Problems with the existing system were highlighted by visualisation of the data using DHIS v2, which thus became a key reform tool. The revised data sets are currently being rolled out nationally, and v2 has been formally adopted for the process, which is one of the largest HIS implementations in the Global South.

Third, the national success in South Africa and new initiatives in a number of countries in Africa occurred at the same time that international actors placed national health information systems at the top of the global health agenda, generating interest from a wider audience. Since action was needed in many countries, and few other actors were operating in several countries with the same generic and free software, HMN asked HISP to support Sierra Leone with DHIS v2, and WHO initiated collaboration on software development. As the scale of efforts expanded, increased emphasis was unavoidably put on efficiency and practical considerations, seemingly overshadowing the overt political agenda, i.e., somewhat similar to the trajectory of the PD tradition. Nevertheless, the focus on deprived communities remains the same.

On the practical side, the HISP case also displays clear benefits from distribution and networking. On one hand, a string of local implementations have benefited from the global toolbox of best practices and software, stemming from multiple implementations and cross country collaboration. On the other hand, as illustrated in Figure 1, the material support afforded by the global toolbox has been greatly enhanced through the extensive distribution, and the shared material fit well with Orlikowski's (2006) notion of scaffolding.

However, the HISP case shows that to ensure working information systems in the South, we must look beyond temporary scaffolds. The circulation of people is essential to properly exploit the toolbox. An overall observation from the case is that the most efficient design happens in many different contexts where technical developers, domain experts and stakeholders meet. Day-to-day distribution should therefore, if possible, be interspersed with co-location, in line with the DSD recommendations from Lings et al (2006) listed in section 2.

But even isolated boundary spanning is not sufficient. Thorough local capacity building is crucial, underscoring the importance of engaging students and health staff in learning by trying, using and doing. These efforts must have a long term vision, and we propose the concept of *institutional scaffolding* to emphasise the importance of activities such as the development of training

programs, academic degrees, and various formal standards, as well as the creation of information system cadres and ministerial HIS units. Such efforts must be durable to have real impact, and therefore, this kind of scaffolding must be relatively permanent (analogously with the century perspective on constructing the Sagrada Familia in Barcelona), and involve institutionalisation. As these become fully institutionalised and independent, they can be said to have morphed into Fischer's (2008) technical and social infrastructures for meta-design.

6 Conclusion

The contributions of this paper consist of adding perspectives to the study of distributed participatory design, by 1) extending it to the Global South and 2) emphasizing the role of implementers as boundary spanning mediators, 3) enabling active generification of innovations that arise in local settings and 4) collaboration on the DPD of a common toolbox which in turn 5) affords learning-by-trying through scaffolding.

In the perspective of the PD literature, parallels emerge between the early Scandinavian projects and the current challenges facing information systems in the Global South. The HISP case shares many features with for example the influential UTOPIA project (Bødker et al. 1987), in terms of both practical methods and political aims. The approaches are similar in their focus on empowerment through participatory capacity building and on the generation of useful tools for and by people at the margins in global development, technological and otherwise. However, while the UTOPIA project stranded as commercial technologies improved, the timing is better for HISP. Today, FOSS technologies are at the cutting edge of technological development, and huge corporations are embracing them. For example, IBM is using FOSS for competitive advantage (Eclipse Foundation 2007). These practices are potentially of immense importance in the Global South (Weerawarana and Weeratunga 2004). However, as discussed in this paper, much scaffolding and learning by trying is needed before developing countries will be able to master them "as their own language".

Still, as we have shown, the HISP approach to institutional scaffolding of DPD contains elements of a model that can plausibly be extended beyond its origins in the public health care sector. Moreover, the major contribution of the HISP case lies in the demonstration that the learning engendered by use of the system in the Global South can make considerable contributions to generic software design. In other words, emphasis should be put on facilitating two-way learning and innovation processes. Despite the challenges, the design of global software potentially stands to benefit from wide distribution of use.

7 References

AbouZhar, C., and Boerma, T., "Health information systems: the foundations of public health," *Bulletin of the World Health Organisation*, (83:8), 2005, pp. 578-583.

- ANC “The Reconstruction and Development Programme,” African National Congress, Johannesburg, 1994a.
- ANC “A National Health Plan for South Africa,” African National Congress, Johannesburg, 1994b.
- Avgerou, C. *Information Systems and Global Diversity*, Oxford University Press, 2002.
- Barcellini, F., Détienne, F., Burkhardt, J.M. “Users and developers mediation in an Open Source Software Community: boundary spanning through cross participation in online discussions,” *Int. J. Human-Computer Studies*, (66:7), 2008 pp. 558–570.
- Bjerknes, G. and Bratteteig, T., “User Participation and Democracy: A Discussion of Scandinavian Research on System Development,” *Scandinavian Journal of Information Systems*, (7:1), 1995, pp. 73-98.
- Bødker S. and Grønbaek, K., “Design in Action: From Prototyping by Demonstration to Cooperative Prototyping,” in *Cooperative Design of Computer Systems*, Greenbaum, J. and Kyng, M., editors, Lawrence Erlbaum Associates, Hillsdale, NJ, 1991, pp. 197–218.
- Bødker, S., Ehn, P., Kammersgaard, J., Kyng, M. and Sundblad, Y., “A Utopian Experience – On Design of Powerful Computer-based Tools for Skilled Graphic Workers,” in *Computers and Democracy – A Scandinavian Challenge*, Bjerknes, G. et al., editors, Avebury, 1987, pp. 251-278.
- Braa, J., Monteiro, E., Riiser, V., Vedel, E., Aanonsen, K., “Experiences from European health informatics: Lessons for African systems development?,” *Health Informatics in Africa: HELINA 93*, Mandil, S., Moidu, K., Korpela, M., Byass, P., Forster, D., editors, Elsevier, 1993, pp. 161-165.
- Braa, J., “Community-based participatory design in the Third World,” *PDC '96: Proceedings of the Participatory Design Conference, Computer Professionals for Social responsibility*, Blomberg, J., Kensing, F., and Dykstra-Erickson, E, editors, USA, 1996.
- Braa, J., Hedberg, C., “The struggle for developing District health information systems in South Africa,” *Information Society*, (18:3), 2002.
- Braa, J., Monteiro, E., and Sahay, S., “Networks of action: sustainable health information systems across developing countries,” *MIS Quarterly*, (28:3), 2004a, pp. 337-362.
- Braa, J., Titlestad, O.H., and Sæbø, J., “Participatory health information systems development in Cuba – the challenge of addressing multiple levels in a centralised setting,” *Proceedings of PDC 2004* (1), Toronto, Canada, 2004.
- Braa, J., Hanseth, O., Mohammed, W., Heywood, A., and Shaw, V., “Developing Health Information Systems in Developing Countries: The Flexible Standards Approach,” *MIS Quarterly*, (31:2), 2007, pp. 381-402.
- Byrne, E., and Sahay, S., “Health information systems for primary health care: Thinking about participation,” *Proceedings of the International Federation of Information Processing, IFIP 9.4 and 8.2 Joint Conference on Organisational Information Systems in the Context of Globalisation*, Korpela M., Montealegre R., and Poulymenakou, A., editors, Dordrecht, The Netherlands, Kluwer, 2003, pp. 237–249.
- Carlile, P. R., “A Pragmatic View of Knowledge and Boundaries: Boundary Objects in New Product Development,” *Organisation Science*, (13:4), 2002, pp. 442-455.
- Chesbrough, H., *Open Innovation: The New Imperative for Creating and Profiting from Technology*, Harvard Business School Press, Cambridge, MA, 2003.

- Eclipse Foundation, Eclipse Open Innovation Networks, <http://www.eclipse.org/org/foundation/membersminutes/20070920MembersMeeting/07.09.12-Eclipse-Open-Innovation.pdf>, 2007, visited October 2008.
- Elden, M., and Chisholm, R. F., "Emerging varieties of action research: Introduction to the special issue," *Human Relations*, (46:2), 1993, pp. 121-142.
- Engelstad, P. H., and Gustavsen, B., "Swedish network development for implementing national work reform strategy," *Human Relations*, (46:2), 1993, pp. 219-248.
- Finck, M., Gumm, D., Pape, B., "Using Groupware for Mediated Feedback," *Proceedings of PDC 2004*, (2), Toronto, Canada, 2004.
- Fischer G., "Rethinking software design in participation cultures," *Automated Software Engineering*, (15:3), 2008, pp. 365-377.
- Fleck, J., "Innofusion or Diffusation? The Nature of Technological Development in Robotics," Edinburgh University PICT Working Paper (4), 1988.
- Fleck, J., "Learning by Trying: The Implementation of Configurational Technology," *Research Policy*, (23), 1994, pp. 637-652.
- Fogel, K., *Producing open source software*, Sebastopol, CA, O'Reilly, 2005.
- Greenbaum, J., and Kyng, M., (editors), *Design at work: Cooperative design of computer systems*, Lawrence Erlbaum Associates, Publishers, 1991.
- Gumm, D.C., "Distributed Participatory Design: An inherent Paradoxon?," *Proceedings of IRIS29*, Helsingør, Denmark, 2006a.
- Gumm, D. C., "Distribution Dimensions in Software Development Projects: A Taxonomy," *IEEE Software*, (23:5), 2006b.
- Hanseth, O., Monteiro, E., and Hatling, M., "Developing information infrastructure: The tension between standardisation and exibility," *Science, Technology, & Human Values*, (21:4), 1996, pp. 407-426.
- Heeks, R., "Health information systems: failure, success and improvisation," *Int J Med Inform*, (75:2), 2006, pp.125-37.
- von Hippel, E., *Democratizing Innovation*, Cambridge, MIT Press, 2005.
- Imslund, V., and Sahay, S., "Negotiating Knowledge: The Case of a Russian-Norwegian Software Outsourcing Project," *Scandinavian Journal of Information Systems*, (17:1), 2005, pp. 101-130.
- Kensing, F., *Methods and Practices in Participatory Design*, ITU Press, 2003.
- Kimaro, H., and Titlestad, O.H., "Challenges of user participation in the design of a computer based system: The possibility of participatory customisation in low income countries," *Proceedings of IFIP WG. 9.4*, Abuja, Nigeria, 2005.
- von Krogh, G., and Spaeth, S., "The open source software phenomenon: Characteristics that promote research," *Journal of Strategic Information Systems*, (16:3), 2007, pp. 236-53.
- Lanzara G. F., Morner M., "Artifacts rule! How organizing happens in open-source software projects," in *Actor-Network Theory and Organizing*, Czarniawska B., and Hernes T., (editors), Copenhagen Business School Press, Copenhagen, 2005, pp. 67-90.
- Lings, B., Lundell, B., Ågerfalk, P. J., and Fitzgerald, B., "Ten Strategies for Successful Distributed Development," in *The Transfer and Diffusion of IT for Organisational Resilience*, Donnellan, B., Larsen, T. J., and Levine, L., (editors), Springer Science & Business Media, 2006.

- Loebbecke, and Powell, "Furthering Distributed Participatory Design: Unlocking the walled gardens," *Scandinavian Journal of Information Systems*, (21:1), 2009, pp. 79-108.
- Nygaard, K., "The Iron and Metal Project. Trade Union Participation," *Proceedings of the CREST Conference on Management Information Systems*, London, Cambridge University Press, 1977.
- Obendorf, Janneck, and Finck, "Intercontextual Distributed Participatory Design: Communicating Design Philosophy and Enriching User Experience," *Scandinavian Journal of Information Systems*, (21:1), 2009, pp. 53-78.
- O'Reilly, T., *The Architecture of Participation*, Available from <http://www.oreillynet.com/pub/wlg/3017?wlg=yes>, 2003, visited October 2008.
- Orlikowski, W., "Material knowing: the scaffolding of human knowledgeability," *European Journal of Information Systems*, (15:5), 2006, pp. 460-466.
- Peine, A., "The sources of knowledge: Towards a framework about use, consumption and industrial dynamics," *25th Celebration Conference on Entrepreneurship and innovation: organisations, institutions, systems and regions*, Copenhagen, CBS, Denmark, 2008.
- Pollock, N., Williams R., and D'Adderio, L., "Global software & its provenance: generification work in the production of organisational software packages," *Social Studies of Science*, (37:2), 2007, pp. 254-280.
- Preece, J., Nonnecke, B., and Andrews, D., "The top five reasons for lurking: improving community experience for everyone," *Computer in Human Behavior*, (20:2), 2004, pp. 201-223.
- Rolland, K. H., and Monteiro, E., "Balancing the local and the global in infrastructural information systems," *The Information Society*, (18:2), 2002, pp. 87-100.
- Sahay, S., Nicholson, B., and Krishna S., *Global IT Outsourcing*, Cambridge University Press, 2003.
- Sandberg Å., (editor), *Computers dividing man and work*, Swedish Center for working life, Demos project report no 13, Utbildningsproduktion, Malmö, 1979.
- Sarant, S. A., *The role of organizational boundary spanners in industry/university collaborative relationship*, Doctor of Philosophy in Psychology Dissertation Thesis, North Carolina State University, USA, 2004.
- Sonnenwald, D.H., "Communication role that support collaboration during the design process," *Design Studies*, (17:3), 1996, pp. 277-301.
- Sheikh, Y. H., and Titlestad, O. H., "Implementing Health Information System in Zanzibar: Using Internet for communication, information sharing and learning," *IST Africa conference*, Windhoek, Namibia, 2008.
- Star, S. L., "The structure of ill-structured solutions: Boundary objects and heterogeneous distributed problem solving," *Morgan Kaufmann Series In Research Notes In Artificial Intelligence*, (2), Huhns M., and Gasser, L., (editors), Morgan Kaufman, Menlo Park, CA, 1989, pp. 37-54.
- Staring, K., and Titlestad, O. H., "Development As Free Software: Extending Commons Based Peer Production To The South," *ICIS conference*, Paris, 2008.
- Tierney, W. M., Kanter, A. S., Fraser, H. S. F, and Bailey, C., "Establishing Partnerships to Promote eHealth in Developing Countries: Lessons from Africa," available from http://ehealth-connection.org/files/conf-materials/Establishing%20Partnerships%20to%20Promote%20eHealth_0.pdf, 2008, visited January 2009.

Weerawarana, S., and Weeratunga, J., *Open Source in Developing Countries*, Swedish International Development Cooperation Agency, Stockholm, Sweden, 2004.

West, J., and Gallagher, S., "Patterns of Open Innovation in Open Source Software," in *Open Innovation: Researching a New Paradigm*, Chesbrough, H., Vanhaverbeke, W., and West, J., (editors), Oxford University Press, 2006, pp. 82-106.