

12-2010

## Technology Capacity Development through OSS Implementation: The Case of Public Higher Education Institutions in Ethiopia

Nigussie Tadesse Mengesha

*University of Oslo*, Nigussim@ifi.uio.no

Follow this and additional works at: <https://digitalcommons.kennesaw.edu/ajis>



Part of the [African Studies Commons](#), [Higher Education Administration Commons](#), [International and Comparative Education Commons](#), and the [Software Engineering Commons](#)

---

### Recommended Citation

Mengesha, N. T. (2010). Technology Capacity Development through OSS Implementation: The Case of Public Higher Education Institutions in Ethiopia. *The African Journal Of Information Systems*, Volume 2, Issue 1, pp.1-24.

This Article is brought to you for free and open access by DigitalCommons@Kennesaw State University. It has been accepted for inclusion in *The African Journal of Information Systems* by an authorized editor of DigitalCommons@Kennesaw State University. For more information, please contact [digitalcommons@kennesaw.edu](mailto:digitalcommons@kennesaw.edu).





# TECHNOLOGICAL CAPACITY DEVELOPMENT THROUGH OSS IMPLEMENTATION: THE CASE OF PUBLIC HIGHER EDUCATION INSTITUTIONS IN ETHIOPIA

Research Paper

Volume 2, Issue 1, December 2010

ISSN 1936-0282

Nigussie Tadesse Mengesha  
Nigussim@ifi.uio.no  
Department of Informatics  
University of Oslo  
P.O Box 1080, Blindern  
NO-0316, Oslo  
Norway

*(Received April 30, 2010; accepted November 20, 2010)*

## Abstract

The Open Source Software (OSS) literature gives little attention to the study and practice of domain specific end-user OSS implementation in general and in the public sector of developing countries (DCs) in particular. This paper, however, investigates the trajectories of two OSS-based information systems (IS) implementation projects in a developing country (DC) context with the aim to uncover the practice-based learning and resource sharing evident among locally situated and globally dispersed developers and users. The result of the interpretative case study research shows that the OSS philosophy and practice of software development, implementation and ownership, facilitates for the emergence of practice-based learning from the sharing of implementation accounts and artifacts without sharing the same context of work. Thus, the paper argues in favor of an implementation approach that focuses on distributed practice-based experience, knowledge and resource sharing, and learning with the mediation of the information infrastructure in order to facilitate and sustain OSS-based IS implementation in DCs. The paper contributes both to the OSS and IS implementation literatures by showing the mechanisms of developing the technological capacity of indigenous groups and using the trans-situated learning model as a means to understanding the learning dynamics in OSS implementation.

## Keywords

Open source software; developing country; capacity development; trans-situated learning

# TECHNOLOGICAL CAPACITY DEVELOPMENT THROUGH OSS IMPLEMENTATION: THE CASE OF PUBLIC HIGHER EDUCATION INSTITUTIONS IN ETHIOPIA

## 1. INTRODUCTION

Information and Communication Technology (ICT) has been deemed by many as a viable means to facilitate socio-economic developments in developing countries (DCs) even if realization has been challenging to date (Câmara & Fonseca, 2007; Heeks, 2002; Heeks & Stanforth, 2007; Sahay & Avgerou, 2002; Walsham, Robey, & Sahay, 2007). The advent of open source software (OSS), however, is believed to open up new opportunities and possibilities to this end (Kogut & Metiu, 2001; Weber, 2003; Weerawarana & Weeratunge, 2004). In principle, adoption of OSS reduces license costs and total cost of ownership (Waring & Maddocks, 2005); promotes indigenous technological development (Weber, 2003; Weerawarana & Weeratunge, 2004); avoids being hostage to proprietary software and vendors (Weber, 2003); guarantees against buried “espionage software” (Weber, 2003); advances knowledge more quickly (Câmara & Fonseca, 2007); promotes adaptation to changing organizational environments (Gallego, Luna, & Bueno, 2008); and helps to set up an information economy (Weerawarana & Weeratunge, 2004). In practice, however, these advantages are not well established (bridges.org, 2005a). The extant OSS literature also focuses mainly on development projects, developers, products, organization and governance structure, and innovation emphasizing back-end applications such as server software and middleware (von Krogh & von Hippel, 2006). The implementation aspect of domain specific end-user OSS such as library and health information systems (ISs), in general, and in DCs, in particular, has not been given much attention except for a few studies (Fitzgerald & Kenny, 2004; Lungo, 2006; Waring & Maddocks, 2005).

The emergence of OSS has changed the philosophy and practice of software production, implementation, and ownership from copy right to copy left, from closing to opening source code and from organization-based to individual-driven production and implementation (Feller, Fitzgerald, Hissam, & Lakhani, 2005; Raymond, 2000; von Hippel & von Krogh, 2003). The production and implementation relies more on loosely coordinated globally distributed developers/users with the mediation of the information infrastructure. These changes have direct ramifications on IS implementation in general, and in DCs in particular, given context-sensitiveness of ISs, complexities of IS implementation and hindrances in DCs (Avgerou, 2008; Heeks, 2002; Sahay & Avgerou, 2002). DCs lack techno-scientific knowhow and the public sector has been dwarfed by brain drain (Avgerou, 2000; Waring & Maddocks, 2005). Arrangements that allow for situated learning such as acquisition of equipment, technical assistance, education and training, and direct foreign investment were unsuccessful to make DCs able to adapt, enhance and maintain ISs (Odedra, 1992). The altered philosophy and practice, however, makes less experienced developers in DCs to be forerunners and in charge of customization and implementation activities with an open participation of developers/users around the globe. These changes can have different ramifications on implementation activities in DCs. This research investigates the micro-processes of an end-user OSS implementation in public higher education sector in Ethiopia in order to uncover the practice-based learning and

resource sharing practices evident among locally situated and globally dispersed developers and users. The research seeks to answer the following questions: How can distributed computing environments be used to facilitate end-user OSS implementation in the public higher education sector of Ethiopia? How can the public higher education sector develop technological capability through OSS implementation? What is the role of information infrastructure in stimulating the learning process?

The research investigates the trajectories of OSS implementation projects in the Addis Ababa University Library (AAUL) and the College of Telecommunications and IT Library (CTITL). The libraries were introduced to OSS in different circumstances, downloaded a similar OSS-based library information system (OLIS) called Koha from the Internet, followed diverse trajectories and implemented Koha with the support of partners. Sections four and five describe the implementation processes of Koha in AAUL and CTITL respectively, following the presentation of the study's research methods documented in section three. The next section (section two) discusses the learning processes that can emerge from the sharing of accounts of related practices without sharing the same context of work and the lens used to understand such learning – the trans-situated learning (TSL) model – the notion adopted in this research. The analysis of the cases in light of the TSL model are presented in section six followed by a discussion in section seven. Finally concluding remarks are presented in section eight.

## **2. OSS, DEVELOPING COUNTRIES, AND TRANS-SITUATED LEARNING**

### **Open Source Software**

OSS has changed the philosophy and practice of software development and ownership and impacted implementation. OSS development and implementation rely mainly upon individuals making choices to volunteer their time and skills to a production activity organized in a bazaar style (Raymond, 2000) dispersed across organization, geography, culture and time (Mockus, Fielding, & Herbsleb, 2005). The license dictates disclosure and dissemination of source code so that anyone with proper programming skills and motivations could use, modify and distribute any OSS written by anyone (Neumann, 2005; von Hippel & von Krogh, 2003). OSS developers/contributors and users coordinate activities, deliver products, and offer field support using the Internet usually without the need for an intermediary or a vendor and face-to-face communication (Lakhani & von Hippel, 2003; Mockus et al., 2005; von Hippel, 2002, 2005). This form of implementation, the community-based OSS implementation model, invites the voluntary participation of anyone interested.

The philosophy and practice of OSS impacted implementation of back-end applications (such as operating systems, server software, etc) and domain specific end-user applications differently. Often IT professionals are the primary users of back-end applications and they are able to understand and change source codes. Development and use of such systems is like “scratching own itch” as stated by Raymond (2000). In the case of front-end applications, however, users may not be computer experts, and hence, unable to configure and use systems by themselves (Ducheneut, 2005). The implementations of such systems require IT experts that mediate users' requirements, systems capabilities and local circumstances. These IT experts might not have domain expertise. Furthermore, such systems embed local idiosyncrasies of the place of

production (Heeks, 2002; Pollock, Williams, & D'Adderio, 2007) demanding for adaptation to use contexts.

## Developing Countries

Some DCs have already set up programs and legislations prioritizing OSS in the public sector and are reaping the fruits while others are following a similar suite (Chan, 2007; Waring & Maddocks, 2005; Weerawarana & Weeratunge, 2004). For example, the governments of China, India, Peru, and Brazil have set up programs and regulations to encourage the growth of OSS through distribution of OSS, education, and government procurement preferences and tax benefits for OSS firms (Chan, 2007; Subramanyam & Xia, 2008). In Africa countries such as Angola, Benin, Djibouti, Kenya, Senegal, South Africa, Tanzania, Uganda and Zambia have formulated OSS specific policies or references to OSS or open standards in publicly available documents (bridges.org, 2005b). Transferring systemic technologies, regardless of license, to DCs has been complex and challenging because of adaptation requirements and the need to transfer operational and technical know-how (Braa, Monteiro, & Reinert, 1995; Nhampossa, 2005; Odedra, 1992). Some authors (Braa et al., 1995; Braa, Monteiro, & Sahay, 2004; Heeks, 2002; Nhampossa, 2005) state that ISs cannot be transferred and be put to use exactly as planned during design, especially from the north to the south and propose the notion of translation instead of transfer. Researches also attest to the use and consequences of ISs to emerge unpredictably from complex interactions between the social and the technical in a specific context in time (Markus & Robey, 1988; Orlikowski, 1992, 1996). The unpredictability and restructuring would be more evident in the context of IS implementation in DCs because of country context differences and the resulting gap between IS designs and realities in implementation contexts (Heeks, 2002). DCs may not have economic resources and indigenous techno-scientific capabilities (Avgerou, 2000; Heeks, 2002) that can facilitate the co-development of the social and the technical that is evident during IS implementation (Monteiro, 2000). Furthermore, DCs have been dwarfed by brain drain and western experts that are new to implementation contexts often fail to deliver expected outputs (Heeks, 2002; Odedra, 1992). And yet, the OSS phenomenon renders the responsibility of customization and implementation to the locals and avoids situated practices.

Vendors provide technical and operational supports and trainings to clients when they are involved in implementation activities. Their services also continue for a certain period of time depending on service level agreements. Such situated arrangements and service level agreements do not exist in a community-based OSS implementation practice. Hence, the potential advantages of OSS to DCs depend on the development or the existence (or not) of local capacity that could configure, customize, and implement OSS. For example, Odedra (1992) indicates how ITs donated by the west to DCs failed to be operational because of lack of local skilled personnel, problems with western consultants and ineffective local capacity development strategies. The fate of OSS in DCs could be the same as donated technologies unless the locals are able to assimilate it. In this respect, unlike in the case of other arrangements, the community-based implementation itself, opens up new opportunities for local developers to develop the technological capability and improve IS implementation through a distributed practice-based voluntary participation of experts around the world. This trans-situated form of learning/capacity development is the concern of this research, and the next section explains its essence.

## Trans-Situated Learning (TSL)

OSS facilitates learning and software development and coordination of activities in a loosely coordinated distributed environment (Hossain & Zhu, 2009; von Krogh, 2002). Studies indicate that networking among actors of aligned interests assists learning and IS implementation (Braa et al., 2004; Heeks & Stanforth, 2007). For example, by using local/global network model, Heeks and Stanforth (2007) explain that in e-Government implementation projects in DCs there are a 'global' set of resource providers and a 'local' set of implementers, and the degree and balance of participation in this network determines success. In order for local intervention to be robust and sustainable, skill is required to be transferred from where success is achieved to new sites, which could be possible through networking and formation of alliances. Braa, Monteiro, and Sahay explain that "[e]stablishing networks creates opportunities for sharing experience, knowledge, technology, and value between the various nodes of the experience" (2004, p. 341). The networking is not about growing the size to reach the level of critical mass but to facilitate the necessary learning process, which they say, is pre-requisite to sustainability.

Such practice-based learning and sharing is possible among people that share physical context or are separated by multiple boundaries such as functional, geographical, or organizational. The situated learning theory and the associated communities of practice (CoP) analytical device capture the learning dynamics among people engaged in similar practices and have frequent occasions to interact with each other often face to face (Brown & Duguid, 2001; Vaast & Walsham, 2009). However, it fails to capture the practices of people who share occupational interests but who are not working together and who cannot interact face to face (Brown & Duguid, 2001; Vaast & Walsham, 2009). The emergence of IT, particularly the Internet, facilitates "the acquisition and generation of new knowledge and competences based on similar practices and experiences, and beyond the confines of shared locations or of institutionalized educational settings" (Vaast and Walsham 2009: p. 547). Brown and Duguid (2001) introduced the concept of a "network of practice" (NoP) to describe people that are not necessarily collocated and may never have met face to face but engaged in similar practices. Considering this notion, Vaast and Walsham (2009) propose a trans-situated model of learning (depicted in Figure 1) that makes use of an information infrastructure and supports networks of practices. Trans-situated learning refers to

[P]ractice-based learning dynamics that are not limited by the bounded context of CoPs, [...] Such learning processes may emerge throughout NoPs, i.e., among people who share a certain degree of similarity in their practices but who lack the ability to interact directly with each other on a regular basis. (Vaast and Walsham 2009: p. 547)

The trans-situated learning model builds upon the notions of CoPs, NoPs and an information infrastructure. Newcomers to a practice learn from old timers through an apprenticeship like processes of legitimate peripheral participation (LPP) and by increasing access to other CoPs' practices. Resources move among CoPs, and members engage with peers based on shared practices transcending geographical, organizational and functional boundaries being conditioned and supported by an information infrastructure.

Likewise, the development and implementation of OSS has been realized through the practices of individuals and communities imbedded in local contexts, and dispersed across boundaries

through experience and resource sharing. Sharing beliefs, values, communications, artifacts and tools among OSS developers/users enables not only cooperation, but also provides a basis for shared experience, camaraderie, and learning (Scacchi, 2005). This study analyses the trajectories of two OSS implementation projects in light of the trans-situated learning (TSL) model to find out the processes of indigenous technological capacity development in a resource constrained setting.

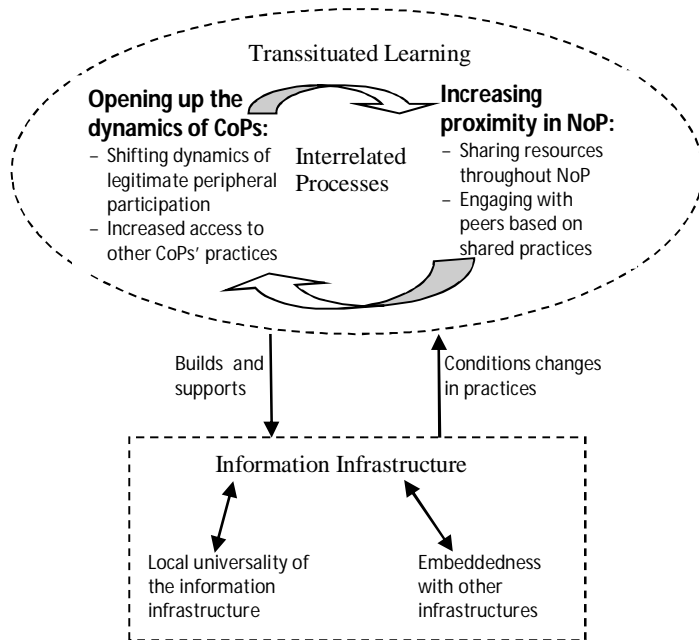


Figure 1: A Model of “Trans-Situated” Learning Supported by an Information Infrastructure (Vaast and Walsham 2009)

## RESEARCH METHODS

The research adopted the qualitative research approach (Silverman, 1998, 2005) with the underlying epistemological and ontological notions of the interpretive philosophy (Klein & Myers, 1999; Myers & Avison, 2002; Orlikowski & Baroudi, 1991; Walsham, 1993, 1995, 2006). The interpretive tradition presumes a social constructionist perspective on reality and the construction of knowledge; it rejects the possibility of an ‘objective’ or ‘factual’ account of events and situations. The interpretative approach has become more interesting and relevant as the focus of IS research shifts from being purely technological to the one that includes behavioral and organizational aspects, consequently when more interest was placed on the interaction between context and innovation (Benbasat, Goldstein, & Mead, 2002; Galliers & Land, 2002). The case study research is particularly well suited to IS research (Lee, 1989; Myers & Avison, 2002) because it facilitates the study of IS in a natural setting and allows the researcher to answer ‘how’ and ‘why’ questions (Benbasat et al., 2002). The research involved two separate contexts leading to a multiple-case study design (Yin, 2003). The case study method suggests data collection through multiple means from various sources including documents, archival records, interviews, direct observation, participant observation, and physical artifacts (Benbasat et al., 2002; Yin, 2003). However, Walsham argues in favor of the interview as an important data source for interpretative case studies (Walsham, 1995, 2006). Thus, as an interpretative case

study, this research employed multiple data collection techniques such as interviews, discussions, observations and analysis of email archives.

The researcher was employee of AAUL from 1996 to the end of 2002, until he joined a teaching department in the same university. During this period, he served the library as chairperson of an automation team (August 2000 – August 2004) and as systems librarian. The automation team was tasked with computerizing services and facilitating IT utilization in the library. The library started OLIS customization (the concern of this study) at the end of 2004, after the researcher left the automation team as a project manager and the library as an employee. However, he was closely following the process partly for a research purpose. The researcher made several field visits and conducted interviews for researches that had different themes during the course of Koha's implementation in AAUL. Fieldwork specifically aimed at this research was conducted from September to December 2008 in both AAUL and CTITL.

In AAUL, two developers were responsible for the technical aspect of Koha, i.e., configuration, customization, enhancement, and maintenance activities. A project manager, who was later appointed to a senior position in the library, was in charge of the overall aspects of the project. Only the cataloging and the OPAC modules of Koha were operational at the time of this research, and it meant that catalogers were the only users of the new system from the staff side. Therefore, developers, the project manager, and catalogers in AAUL were sources of data for this research. The researcher interviewed the two developers, the project manager, and three of the catalogers for about 1:30 to 2:00 hours each. The interviews with the developers and the catalogers were recorded and detailed notes were taken while interviewing the project manager. Furthermore, the researcher observed activities of the staff in the cataloging department while they were performing the usual activities and discussed with them about different aspects of the new system such as, including but not limited to, improvements in work practices, challenges, and overall perceptions.

One person was responsible for customization, implementation and subsequent activities of Koha in CTITL. This person was in charge of the library as well as for implementing and enhancing Koha in CTITL. The researcher interviewed the only developer to understand the implementation and enhancement processes of Koha in CTITL. The interview was recorded and lasted for about 2 hours. In addition, the researcher made on-site observation and informal discussion with another librarian in CTITL to understand his views about the new system especially concerning improvements in work practices, challenges, and overall perceptions.

The developers in both libraries exchanged emails with the Koha community and studied email archives to solve configuration, customization, and enhancement problems. The researcher reviewed the mailing list archives of Koha developers and users; specifically, examined the emails exchanged between local developers in the two libraries and the Koha community between December 2004 and February 2010. December 2004 was the time when AAU started customizing Koha. Table 1 summarizes the data collection methods.

Site	Data Collection Method
------	------------------------



AAU Library	<ul style="list-style-type: none"> <li>• Field notes from formal and informal discussions</li> <li>• Interview with developers, project manager and catalogers/users</li> <li>• Onsite observation</li> <li>• Developers and users mailing lists archive of the adopted system</li> </ul>
CTIT Library	<ul style="list-style-type: none"> <li>• Interview with developer/project manager</li> <li>• Onsite observation and informal discussion</li> <li>• Developers and users mailing lists archive of the adopted system</li> </ul>

**Table 1: Summary of Data Collection Methods.**

The analysis was centered on the pattern of iteratively reading data, identifying key themes, and then relating them to the conceptual framework. The researcher summarized the field notes soon afterwards the interview and listened to the recorded interviews extracting themes focusing on learning, skill development, improvement in implementation, etc especially focusing on the chronology of events and in light of the trans-situated learning model. The same iterative procedure was followed while analyzing the emails exchanged between the locals and the Koha community. Once the data of each site was analyzed separately, comparison was made between the two sites and subsequently, the findings of the analysis were interpreted in light of the IS literature. As Walsham notes, the researcher's mind supplemented by the minds of others is the best tool for analysis, and this research relied mainly on the minds of the researcher supplemented by colleagues (Walsham, 2006). The results of this research were presented to colleagues at different stages and refined further based on their inputs.

## **LIBRARY SYSTEM IMPLEMENTATION IN AAUL**

Although the history of education in Ethiopia traces its origin back to the fourth century AD, Western education system began taking shape in the second half of the nineteenth century with the arrival of missionaries (Zewde, 2002). The first public school and university were opened in 1908 and 1950, respectively. Formal education expanded ever since while the system and its contents were changing to serve the needs of different regimes and interests. The current education system offers 10 years of general education consisting of 8 years of primary education and 2 years of general secondary education (9-10) with the second cycle of secondary education (11-12) which prepares students for continuing to higher education (FMOE – Ethiopia, 2010). The statistics of the government of Ethiopia indicates growth in major education indicators, including education budget and expenditure relative to the total government expenditure (FMOE – Ethiopia, 2010). The number of higher learning institutions also started to grow rapidly very recently. In 2006/07 alone the government opened 12 new universities bringing the total number of government universities to 23 (FMOE – Ethiopia, 2010). The universities collectively had a total of 9,496 teaching staff and enrolled a total of 263,979 students in both undergraduate and graduate programs in 2008/09. Although the number is increasing, the higher learning institutions have been suffering from lack of trained manpower and educational resources.

AAU is the oldest and the largest higher learning institution in Ethiopia which was founded in 1950. It has contributed a lot towards, among others, the expansion of education, including higher learning institutions in the country. Its library, the AAUL is the first academic library in Ethiopia which was founded in 1961. Although the history of modern libraries in Ethiopia begins

in 1930 with the opening of a public reading room and have grown in number throughout the years, their collections and services were very limited (Gupta, 1995; Pankhurst, 1988). AAUL was the major public information resource center in the country, especially for research and academic purposes. The library was composed of more than 19 branches and serves mainly the AAU community, and extends services to external organization and individual users. The Library was organized in a centralized manner, i.e., technical and managerial activities were performed centrally at the main library. A university librarian was responsible for overall operation of the Library with the help of two assistant librarians each responsible for technical and public services. The technical services include acquisition and cataloging while the public services include circulation, reference, departments and branch libraries.

The library used to face challenges because of shortcomings of a manual library system and had financial limitation to implement a computer-based library system. The manual operations of the libraries had limitations to satisfying the needs of users and staff. For example, it did not allow searching for an item by combining keywords or subject headings. Even if a search is successful, locating the material was difficult, i.e., knowing whether the material is in circulation, on the shelf, out of circulation, etc. The Library staff faced difficulties to know the collection size, circulate books to users, compile reports, etc. By introducing library information system (LIS), the libraries aimed at improving services, alleviating drawbacks of the manual system, facilitating interaction with similar libraries, and enhancing their images. The LIS would automate the processes of acquisition of books and serials, cataloging and circulation functions, and facilitates report generation. Using OPAC (online public access catalog), users could easily interact with the libraries, search collections, and operate some circulation activities, for example, extend loan periods.

AAUL started customizing an OLIS at the end of 2004 after previous attempts to purchase and develop a library system failed to bear fruits. In 2004 while preparation for purchase was going on, collaboration between AAUL, ICTDO (ICT Development Office) of AAU, a University in the USA (a professor and his students) and an NGO (co-established by the professor) had initiated a project to customize an OLIS called OpenBiblio. ICTDO was established to manage a University wide computer network (AAUNet) and carry out ICT related R&D activities. Insufficient IT personnel demanded the Library to form an alliance with ICTDO that had a pool of IT experts which were drawn from teaching departments of AAU (computer science, electrical and computer engineering, and information science). To promote sharing of knowledge and experience, the international arrangement mixed local and global members together in groups and started the project but did not go further the planning stage. Despite its failure to deliver the expected product, the arrangement introduced the local team to the world of OSS.

The justification behind switching from purchase to OSS adoption was associated to reducing total cost of ownership and developing local expertise that would serve as knowledge hub for libraries in Ethiopia. According to project documents, vendors quoted prices ranging from 150,000 to 450,000 USD for a LIS that could run on 25 workstations each for staff and user. The price excludes adaptation costs that may be incurred to make the system able to accommodate materials written in Amharic script. Therefore, immediately after the discontinuation of OpenBiblio implementation, AAUL and ICTDO initiated a joint OLIS adoption project in October 2004 and organized a project team comprised of library and ICTDO staff, and appointed a project manager from the library. The alliance targeted at exploiting the technical expertise of

the IT personnel both in customization and implementation, and in developing the technical capacity of the library staff. The objective of this project was to explore and identify OLIS and proceed accordingly. To this end, the team shortlisted OpenBiblio and Koha<sup>1</sup>, finally selected the latter for implementation, and planned to make the library's catalog online by October 2005. The project team met once every week for at least two hours to discuss problems, suggest solutions, evaluate progress, and discuss and approve plans.

Once Koha was approved, the technical group downloaded and made version 2.0.1 ready for testing on AAUNet while the nontechnical group analyzed its functions against requirements of AAUL. According to the project manager, the team piloted Koha before embarking on full-fledged implementation and learned about technical problems, scalability, performance and organizational issues such as coordination and management. During the pilot phase ICTDO trained some of the technical staff of the Library on Linux and Perl (the language used to develop Koha) and made them to participate in technical aspects. These technical people did not have library science education but worked in the library for at least two years. They were serving as an interface between requirements of the library and the ICTDO staff. AAUL has dropped its partnership with ICTDO after the pilot phase and full-fledged implementation continued under the Library's ownership. The previous project manager retained his position but required to staff the project from within the library only. He then organized a team composed of deployment, retrospective conversion (recon), and training sub teams to customize and configure Koha, convert card catalog into electronic format, and train staff and end users respectively. The plan was to implement Koha module by module in such an order cataloging, OPAC, circulation, acquisitions, etc.

The training team organized hands-on trainings on computer basics and operations of Koha mainly to the Library's staff. The retrospective conversion team populated Koha with bibliographic data of books and serials that were written in Latin script. The deployment team (hereafter developer) had two members who were trained on Linux and Perl by ICTDO and were anchor to making Koha up and running. The new arrangement demanded reinstallation and configuration of Koha on the library's server that was separately located in the main library. One of the developers describes the situation as follows:

The initial training was essential to acquaint ourselves with Linux and Perl but it was not sufficient to make us able to install, configure and customize Koha. We were far away from that.... we were overflown by tasks related to database management system, operating system and Koha itself.

The developers downloaded Koha version 2.0.1 and other required software from the Internet and installed on Unix Solaris operating system, and posted the first ever message to the Koha mailing list on 6 November 2004. The message briefly introduces the guy, the library and the version of Koha in use and asks "[...] IS THERE A DOCUMENTATION FOR THE KOHA DATABASE, IF SOME ONE HAS ALREADY DEVELOP IT? OR IS THE POSSIBILITY TO GATE ANY DOCUMENT THAT TALK ABOUT KOHA DATABASE?"<sup>2</sup> The next day someone from France posted

---

<sup>1</sup> Koha is an integrated library system meant for computerizing services of a library. It was initially developed in New Zealand and distributed under the open source General Public License (GPL). Koha has modules for circulation, cataloging, acquisitions, serials, reserves, patron management, branch relationships, and has additional basic and advanced features. It was first deployed in 2000. Further information can be found here <http://koha.org/>

<sup>2</sup> The quotations are directly taken from the emails.

“www.koha.org/wiki is the place where you will find all our documentations.” On 8 December 2004, the same developer posted a similar message “[...] Can any body help me on Koha database design? That is, Is there a documentation for KOHA DATABASE DESIGN? How it looks like?” and another respondent posted on the same date “You asked this question yesterday. If you did not like the answer [Mr. A] gave, perhaps you should be more specific.” Then the developer asked how to interface Koha with a cataloging database and someone else posted the experiences of one library in the USA that uses the same cataloging database.

Besides posting practical information in how to solve a given problem, members of the list advised newcomers in various respects. The following posting is an example response to an inquiry posted by a developer from AAUL “[...] are you sure that's all it says? Look above that line in the log and you should see something more informative. Let us know what it says :-).” The developers in AAUL changed the operating system from Solaris to Redhat and one of them posted “[...] Is there a problem of using Koha 2.2.0 on Redhat 9 Professional Edition?” The advice was as follows:

As far as I know Koha should run fine on Red Hat 9.0. However, I would advise against it unless you know how to install security updates aside from the packaging system (support for Red Hat 9 has ceased). Fedora Core 3 is an option, as is Debian Woody or Sarge, or any other supported disto.

The developers later changed the operating system from Redhat to Debian Linux (Sarge) following the advice. Besides posting inquiries, the developers studied archives of the Koha mailing list and other related materials in order to solve problems and advance their understanding of Koha and its technologies. Sometimes an inquiry posted to the list may not get a response but the job could be done, and a job might not be done also despite advices. For example, despite the advice Koha does not interact with the cataloging database.

Besides solving own problems the developers in AAUL were posting various solutions both to local and international partners. For example, on August 15, 2006 and July 16, 2008 the developer posted the following solutions to a guy who was implementing Koha in another library in Ethiopia:

It is fine to hear words of success in instakking and configuring Koha. [...] add the following line at the end of the apache2.conf file [...] Now go to the koha-httpd.conf file which is located at /etc directory and the virtual host name as follows [...]

If you are behind a proxy, Z39.50 may not work. I have tried this feature [...] You need to talk to network administrators to release the ports that you are using for Z39.50. [...] Regarding the barcode try this if you didn't do before: attach the koha server with an ordinary printer and reboot the system. And try again. [...]

Since the middle of 2007 the project ceased and dealing with Koha became the responsibilities of the reorganized and re-staffed computer center of the library. AAUL started with Koha 2.2.0 and was using version 2.2.4 in June 2009. The last stable release of Koha in June 2009 was 3.0.1, i.e., Koha was changed 11 times after it was introduced in AAUL. Failure to migrate to new releases impacts support as it concentrates on new releases than older versions. The following postings reveal the level of difficulty AAUL faced in migrating to new releases.

I found the guideline on the koha wiki to upgrade from koha 2.2 to koha 3.0 lacks many details and couldn't helped me. I am looking for some who did this successfully. (February 12, 2009)

Recently, we have tried to upgrade [Koha to] the current stable version [3.0.1] and faced some difficulties. So, we would like to outsource it and seeking for an expert who can provide commercial support for the upgrade and other minor customizations [...] (June 3, 2009)

From the beginning of the project in November 2004 until February 2010 AAUL posted a total of 15 questions and 5 solutions. And until the end of 2009 it was utilizing the cataloging and OPAC modules only. Users were searching for availability of a material using the Koha regardless of location and time. Catalogers have been populating the database of Koha with bibliographic data of books but have lots of complaints. They explain, Koha is not properly configured and customized to facilitate their activities and produce reports. The following two excerpts exemplify their problems.

One of the good side of Koha is that you cannot leave the call number field blank. In the rest of the fields, you can write whatever you want and the system never complains. Retrieving data using specific barcode returns lots of unnecessary hits and no one explains why. [...] Koha is not well configured here in our library and I do not recommend it for use unless these problems are resolved.

I sometimes operate the circulation module, which I am not allowed to do so. Technically, nothing prevents me from operating the circulation module as of now. Lending books to users or changing the status of unreturned books to “returned” is possible. This tells me how loose the configuration is and how far behind our system’s configuration is compared to what it should be.

Due to, partly the above problem, both the old and the new system were running in parallel. As a result of the experience with Koha, one of the developers was actively engaged in the development of an OSS digital library called Greenstone. Various officials from local universities also visited the library and asked for support to implement Koha in their libraries.

## **LIBRARY SYSTEM IMPLEMENTATION IN CTITL**

The college of telecommunications and information technology (CTIT) was established in the first half of the 2000s with the sponsorship of the state owned Ethiopian telecommunication corporation. The CTIT Library (CTITL), which was established together with the college, has two branches – the graduate school and telecommunications training libraries and serves about 1000 users composed of students, faculty and employees. It had a collection of around 100,000 materials composed of books, journals and others. The library used to face similar challenges with that of AAU because of the shortcomings of manual operations and lacked of resources to implement a computer-based library system.

The Head librarian (the developer hereafter), who was Librarian and IT expert by training, was central in implementing Koha in CTITL. He was unaware of OSS altogether until he learned about the introduction of Koha in AAUL. Following a brief conversation with the project people in AAUL, he downloaded Koha version 2.2.0 and redhat Linux, and started installation. But he encountered several problems with the operating system before proceeding further, and as a solution he changed from redhat Linux to Fedora. Furthermore, he expected the operating system to install everything that Koha required but found out that it was not the case; identifying the sources of bugs (operating system, database management system or Koha) was problematic; and where and how to ask questions was also among the problems that he encountered during the project. As a manifestation, on August 10 and 12 2005 he has posted messages to the Koha mailing list without content but the term help appearing in the subject field.

In order to solve customization, installation and configuration problems and advance his understanding, the developer has studied documentations of Koha and its technologies (MySQL, Linux, and Perl), and got support from developers and users. He also mentions that outdated and incomplete documentations of Koha created inconvenience in making use of latest release. However, he explains

Users provide valuable information comparable to formal education. Each question and answer of the forum has been educational not only on Koha but also on operating systems and the whole implementation process. I have been learning a lot about Linux, bug fixes, how to implement fixes, etc besides carrying out customization and implementation.

Finally, after intensive work on installation and configuration, and with the support of the community, Koha went live in CTITL in February 2005. Once it was operational in full, succeeding activities concentrated on keeping up-to-date with new releases, dealing with problems that arose as a result and enhancing features. For example, on October 11, 2005 the developer posted an inquiry which says “I update my Koha 2.2.3 to 2.2.4. I think it looks good.” mentions the problem he encountered and inquires for a solution. Later, on May 10, 2007 he posted “[...] Koha z39.50 was working until I upgraded to koha 2.2.9. [...] the error log looks like the following...” and got the following response “This one is not related to dates. It's related to mySQL that is off. I think you can't do anything with Koha atm ;-).” The following inquiry and response on March 29 2007, for example, deals with adding features and localizing Koha further.

I want to translate the template to Amharic (Ethiopian official language). Do you know whom I should contact if it is acceptable? I want to contribute. I want also to link my bibliographic records to full-text pdf file. [...]

First off, welcome to the Koha community. Glad you're interested in contributing. I've set up an Amharic OPAC translation instance at <http://translate.koha.org>. This is a Beta translation site, so if you have any difficulty or have suggestions for how to improve it let me know. (Response)

Since then Koha has been in translation to the Amharic language with the support of other developers from Ethiopia. Koha was fully operational on the college's Intranet since February 2005 with bibliographic details of books, journals and monographs, and also hosts digital books. The developer was continuously upgrading Koha whenever there were new releases, but migration from 2.2.9 to 3.0.1 required major shifts than previous migrations. He further explains that in the past he used to run a command to effect changes but now more shifts that need detailed study are required. In general, the developer was happy with OSS and Koha.

Mostly we talk about cost implication of OSS but it has additional advantages beyond freely acquiring the software. It is flexible for customization and the magnitude of people that take part in such projects is huge which facilitates enhancement and learning. For example, Koha has been changed six times within a short period of time; had it been proprietary it would have not been changed several times within such a short period.

Because of the implementation success of Koha in CTITL, presidents and librarians of various local universities had visited the Library and asked the developer for consultation and support. He advised local librarians on best practices, technologies, learning processes, and networking, etc. CTITL started with version 2.2.3 and was using version 2.2.9 in June 2009 while the latest stable version at the time was 3.0.2. This means that Koha has been changed 8 times after it was introduced in CTITL. From the beginning of the project in January 2005 until February 2010, the

developer has posted 9 messages to the Koha community mailing list and received 6 solutions. The following table summarizes some of the core points in the implementation trajectories of the two organizations.

<b>Feature</b>	<b>AAU Library</b>	<b>CTIT Library</b>
<b>Idea and software acquisition</b>	<ul style="list-style-type: none"> <li>- Unsuccessful previous OSS adoption initiative introduced the locals to OSS.</li> <li>- The Internet was source of the software.</li> </ul>	<ul style="list-style-type: none"> <li>- AAUL was the source for both the idea of OSS and the specific library software.</li> <li>- The Internet was source of the software.</li> </ul>
<b>Implementation Environment</b>	<ul style="list-style-type: none"> <li>- Initiated the project with the support of the University. The project manager was member of the Library administration. There was no external financial support.</li> </ul>	<ul style="list-style-type: none"> <li>- The head librarian had initiated and carried out the implementation with the support of the College. There was no external financial support.</li> </ul>
<b>Implementation Related</b>	<ul style="list-style-type: none"> <li>- A pool of IT experts involved in initial installation and configuration, and training technical staff of the Library.</li> <li>- Focused on mailing list archives for past information, posted inquiries, utilized responses, and studied documentations of Koha and its technologies including from other sources.</li> </ul>	<ul style="list-style-type: none"> <li>- The Librarian had prior knowledge of some of the technologies of Koha and was anchor to the implementation.</li> <li>- Focused on mailing list archive for past information, posted questions, utilized responses, and studied documentations of Koha and its technologies including from other sources.</li> </ul>
<b>Future aspects</b>	<ul style="list-style-type: none"> <li>- Re-established the computer center &amp; hired staff making them responsible for Koha.</li> </ul>	<ul style="list-style-type: none"> <li>- No specific plan was adopted. The same guy was responsible for all activities.</li> </ul>

**Table 2: Summary of OSS Adoption Trajectories in AAUL and CTITL.**

Although AAUL and CTITL have implemented Koha, its implementation level in the two libraries was different. CTITL implemented in full and was utilizing all of the functions of Koha except the acquisitions module because of the non existence of online transaction in Ethiopia. Moreover CTITL was able to include full-text materials besides bibliographic data and customized Koha as its own web site. AAUL however was utilizing only the cataloging and OPAC modules. Even the cataloging module was not properly configured and customized to facilitate cataloging and report productions. Koha was changed 11 and 8 times after it was introduced in AAUL and CTITL respectively. These frequent releases demanded the libraries to frequently change Koha to keep abreast of developments and improvements, and benefit from the active support of Koha community. Upgrading, however, demanded capacity. AAUL was using the older version than CTITL which might mark for the existence of capacity problem in keeping up-to-date with developments and fixes. This problem finally led AAUL to announce for a payment-based support from the Koha community. The developers in both libraries have extended their support to other local libraries although it was in a limited scale and several local university officials visited their respective libraries and demanded support. One of the developers

in AAUL engaged in another related OSS development project because of his experience with Koha. The developer in CTITL considers the frequent release of Koha as a sign of strength and the whole implementation exercise as an entertaining and educational comparable to formal education.

## **CROSS-CASE ANALYSIS**

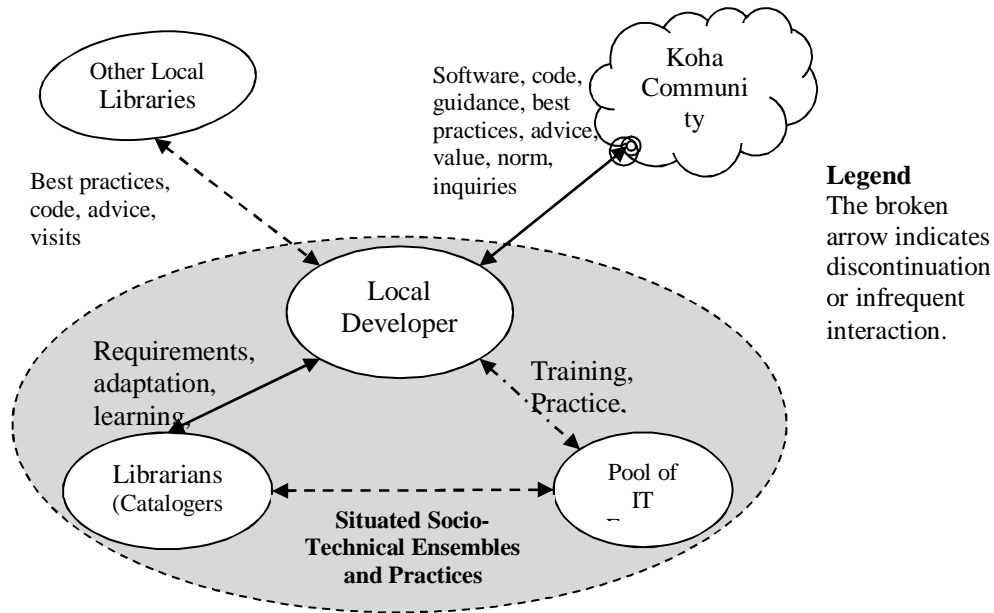
The implementation of Koha in AAUL and CTITL was the result of a collaborative effort among locally co-located and globally dispersed actors that shared experiences, knowledge, technology and value. This form of IS implementation, the community model, is contrary to the traditional practice where often vendors render installation, configuration and training services to customers in their vicinity. Unlike most widely practiced back-end OSS, a LIS is domain specific and contextual. Initially, the Librarian in CTITL and the IT people in AAUL lacked the required expertise to understand the technologies of Koha and install, configure and customize it by themselves. The need to make use of Koha in their respective libraries triggered practice-based learning processes involving local and global actors with the mediation of the information infrastructure. This section analyzes the learning and sharing processes and the elements that were shared among co-located and globally dispersed actors in the courses of implementing Koha in AAUL and CTITL.

### **Situated Socio-Technical Ensembles and Practices**

As depicted in Figure 2, there were three important co-located groups in the technical development of Koha in AAUL – the pool of IT experts (ICTDO staff), and developers and catalogers from the library. The weekly meeting of the project team brought these groups together and created an environment conducive for sharing experiences, collectively solve problems, and discuss and monitor progress. It was a learning occasion for all of the project team members as none of them had a complete knowledge of Koha, library practices and integrated system development and administration. The developers learned the tricks of installation, configuration, customization and administration of Koha through a formal practice-based training offered by ICTDO on the technologies of Koha and by working together with the ICTDO staff on the technical aspect of Koha until the end of the pilot phase.

Koha is an ensemble by itself which is composed of database management system, operating system, and inscribed work processes at a higher level of abstraction. Developers were supposed to learn each of them to make Koha up and running in AAUL. Even if the ICTDO staff was expert in IT and integrated systems, they were novice to the logic and design of Koha. Since the logic is related to library functions, developers from the library were filling in that gap. The pool of IT experts and the developers in the library shared experiences, knowledge, documents and other artifacts about the technologies, installation, configuration and management of Koha, and library practices. The developers have studied the cataloging practice with the support of the cataloging staff while documenting requirements and later on to fix bugs and streamline functions. As indicated in the figure, developers and catalogers shared the tricks of cataloging, best practices, solutions and requirements even if it did not continue throughout the project period.





**Figure 2: Co-located and Trans-Situated Socio-Technical Ensembles, Learning and Sharing in the Implementation of Koha in AAUL**

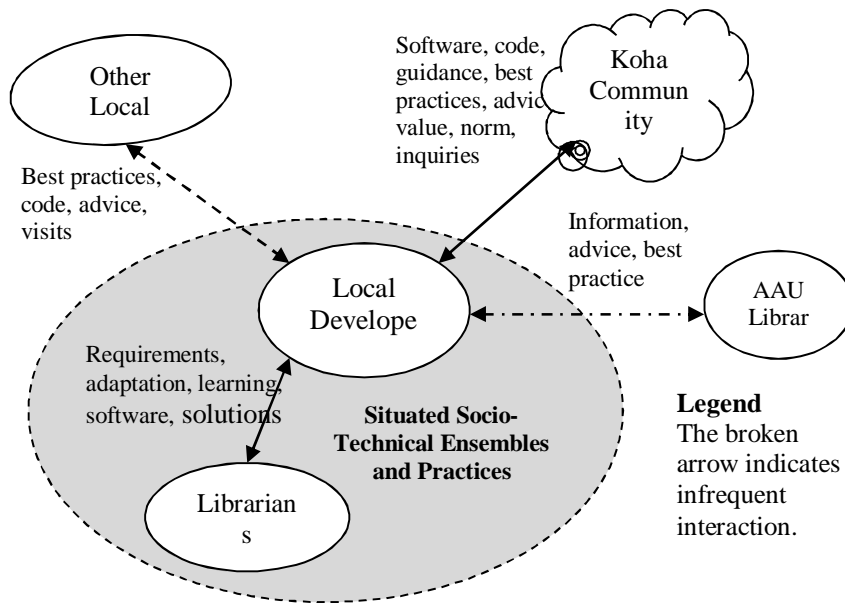
As indicated in the bigger circle in Figure 2, the developers were embedded in a local socio-technical context and dealt with various social and technical matters including project plans, requirements, technical expertise development, adaptation of Koha, dealing with opposition groups, etc. The exchanges of practice-based experiences, knowledge and artifacts among co-located local groups upgraded the knowledge and skills of developers but were insufficient to make them able to understand and implement Koha in AAUL. Disassociation of AAUL and ICTDO, and lack of comprehensive knowledge on Koha necessitated local developers to interact with the developers and users of Koha through email and to study the email archives. The next section elaborates on the interaction between the local and global actors in an attempt to fill up the required skill and knowledge gap.

In CTITL one person, who was imbedded in the local use context, was responsible for the whole process of adaptation and implementation. This individual who is labeled as developer in Figure 3 had library and IT expertise especially on related technologies to that of Koha. The developer learned about OSS, Koha, its source, best practices, etc from AAUL. Further activities such as learning more on the specific technologies of Koha and dealing with installation, configuration, customization, and administration demanded further efforts. Unlike developers in AAUL, the developer in CTITL did not benefit a lot from local partnerships rather sought advice from the developers of Koha by posting inquiries and studying email archives. This trans-situated form learning is discussed below.

### Distributed Practice-Based Learning and Sharing

The essence of OSS lies in the development of software and coordination of activities in a voluntary-based distributed environment (Ducheneut, 2005). Hence, the developers in the two libraries exploited the voluntary support of Koha developers and users that were dispersed across

space, organization and time in the course of understanding, localizing and implementing Koha in their respective libraries. The developers posted inquiries and studied archives of mailing lists to advance their understanding, improve local practice and solve problems. The mailing list archives convey past information exchanged among users and developers concerning the features and functions of Koha, bugs, solutions, suggestions, etc. As indicated in figures 2 and 3, code, guidance, best practices, advices, value, norm, inquiries and solutions were exchanged between local developers and Koha communities.



**Figure 3: Co-located and Trans-Situated Socio-Technical Ensembles, Learning and Sharing in the Implementation of Koha in CTITL**

Analyses of the postings from and to the local developers suggest that the locals have learned the technologies of Koha, and the tricks of customization, implementation and enhancement through time. The level of detail, relevance and completeness of inquiries that the local developers posted were also improved over time. The solutions posted to inquiries of the local developers asked further detail on the nature of a problem, directed to solutions, detailed best practices, provided detailed solutions, suggestions, advices, etc. The first ever posting from AAUL, for example, inquires for Koha's database design documentation, which was basic to deal technically with Koha. Succeeding inquiries however dealt about interfacing Koha with other databases, migrating to new operating systems, making use of new releases and fixing bugs that arose due to new release implementation. Furthermore, a developer from AAUL posted detailed practical and localized solutions to inquiries from another local librarian that followed the footsteps of AAUL. These exchanges mark improvements in the knowledge and skill of local developers and were manifested in the customization, implementation and enhancement of Koha in AAUL.

However, there are advices and suggestions that the AAUL developers did not implement. For example, interfacing Koha with the cataloging database was essential to facilitating cataloging but not implemented despite advices from the community. The cataloging module, which was one of the modules in use so far, was not properly configured to support the cataloging practice as required by catalogers. Even if Koha was changed 11 times after its introduction in AAUL,

AAUL has managed to upgrade five times only from version 2.2.0 to 2.2.4 while the new release at the time was 3.0.2. AAUL has finally announced to the Koha community about its intension of outsourcing the tasks of customization and migration.

The first ever posting from CTITL to the Koha user community was an empty message with the term help appearing in the subject field. Later postings however dealt about migration to new releases, fixing bugs that arose due to new release implementation, adding full text besides bibliographic data and translating Koha into one of the local languages. CTITL has been utilizing all functions of Koha in the Ethiopian context even by including full text materials and customizing Koha as its own web site. It was keeping abreast of new releases relatively faster than AAUL. Translation of Koha into one of the local languages, which was started by the developer, attracted other developers that were not based in either of the libraries but the progress was limited. The developer finally learned how to post relevant inquires, was able to understand and implement fixes, customized Koha according to local needs, implemented Koha, incorporated additional features, changed the look and feel of Koha, and migrated to new releases whenever new versions were out. He also advised some local libraries on best practices, technologies, and OSS in general although it was in a limited scale. The exchanges of accounts and resources among locally situated and globally dispersed developers were possible because of the information infrastructure – the Internet.

## DISCUSSION

The developers who were imbedded in the local context acquired an OSS, studied the code, and upgraded their knowledge and skill through the voluntary support of developers and users dispersed across geography, organization and time because of the OSS license and the community-based OSS implementation model. The association of local and international partners and the sharing of experiences, knowledge, code and best practices were crucial to making Koha up and running in the two libraries. The noninvolvement of license cost for OSS opened up new opportunities to acquire, study, modify and utilize software (Gallego et al., 2008; von Hippel & von Krogh, 2003), specifically a LIS in AAUL and CTITL. As it does not involve cost, adoption of OSS avoids the often lengthy and hectic bureaucratic processes, negotiations and associated corruptions apparent in public sector organizations. Even if the noninvolvement of license cost allows organizations in DCs to easily acquire software, local circumstances including the existence (not) of expertise and an environment conducive for its assimilation impact implementation and subsequent action. As various studies have explained, implementation, utilization, scalability and sustainability of technologies brought from abroad depend very much on the existence or the development of local capacity and the networking of sites where success is achieved to the new ones (Braa et al., 1995; Braa et al., 2004; Odedra, 1992).

The responsibilities of implementation and subsequent activities lie on the hands of clients in the case of community-based implementation. This form of implementation opens up new opportunities for learning and developing capacity through practice being imbedded in the local context. To be imbedded in the local context paves the way for understanding local conditions of practice. Individuals can learn by participating in shared activities in situated and trans-situated contexts both within, and beyond, the contexts of formal schooling, education and training, especially in the workplace and in occupational communities (Fox, 2000; Vaast & Walsham,

2009). The two cases disclose that by participating in shared OSS implementation activities, local developers can learn technologies and the tricks of adaptation, implementation and enhancement, i.e., the study shows the possibility of technological capacity development through a distributed loosely coordinated OSS implementation environment. The local developers acquired knowledge and skill through a situated participation in a local context, by accessing the resources of other communities that are engaged in related practices, and by directly accessing members of other CoP. The participation of the local developers in the development process was shifted from the periphery to the center through practice, learning and interaction with the old timers. However, the level of centrality was different for the two groups of developers which can be apparent considering adaptation and implementation levels of Koha and the reactions of users in the two libraries. Although AAUL started the implementation process earlier than CTITL, it managed to implement only two modules of Koha; even one of the modules did not meet users' requirements. CTITL however implemented Koha in full, localized it further and met the requirements of users. As explained by Heeks (2002) one of the obstacles of IS development in DCs is lack of hybrid expertise that can understand both the context and the technology. The cases show that CTITL was benefited from the hybrid expertise of the developer who was librarian and IT expert by training.

Furthermore, the cases disclose that even if OSS implementation facilitates the development of indigenous technological capacity, it demands a certain level of previous knowledge and skill. The previous knowledge and skill of the developer in CTITL in related technologies helped him to learn the technologies of Koha faster than developers in AAUL. This suggests the importance of giving due attention to develop local capacity that can be able to absorb and work with local and global peers. The arrangement in AAUL to make use of local IT expertise was a good strategy to introduce developers to the technologies of Koha through formal training and practice. Besides making use of Koha in their respective libraries, the local developers were serving as knowledge hub for local libraries, although the sharing of experiences and resources among the locals was limited.

The two cases indicate the importance of participation of local developers both in the global and local arenas to facilitate implementation and develop technical capacity. Localization, contextualization and further deployment could be possible and easier if more local developers are involved in the process, and shared their experiences, knowledge and solutions both among themselves and with the global community. Networking successful actions facilitates the movement of best practices, knowledge, learning, technology, value and norm (Braa et al., 2004) paving the way for successful implementation as well as indigenous technological capacity development. The role of the information infrastructure was paramount in supporting the process. The infrastructure facilitated for the development of a repository of email exchanges as well as access and further communication with peers around the globe.

## CONCLUSION

The research reported in this article explored the trajectories of two OSS implementation projects in the public higher education sector in Ethiopia to uncover the practice-based learning and resource sharing practices evident among locally situated and globally dispersed developers and users. The study was aimed at exploring the learning and resource sharing dynamics among and between locally co-located and globally dispersed users and developers drawing upon the notions

of trans-situated learning. The findings of the research designed as an interpretive qualitative case study show that the OSS license facilitates resource constrained organizations to acquire software free of charge along with the freedom to study, modify, and re-distribute it, bypassing the often lengthy bureaucratic process and corruption apparent in the public sector. Although the license facilitates for software acquisition, the community-based OSS implementation model shifts the responsibilities of customization and implementation to less experienced and unskilled local developers impeding subsequent activities. Nevertheless, the implementation model itself creates an environment conducive for local developers to learn the technologies of the specific software and the tricks of installation, configuration, customization and enhancement. The implementation brings together voluntary developers/users around the world transcending organizational, geographical, and functional boundaries paving the way for sharing advices, best practices, products, information, etc. The participation of local developers in a locally situated implementation activity, their access to email repositories held elsewhere and the possibility to exchange ideas, advices, solutions, and values with communities engaged in similar practices regardless of geography, organization, and function, facilitates implementation as well as the development of technical capacity of the local developers. The information infrastructure plays an important role in supporting these distributed processes. The Internet facilitates and realizes the distributed implementation and learning processes by enabling the development of and access to email repositories which accounts for the experiences of developers and users while developing, implementing and utilizing systems in different parts of the world. The study, in general, shows the processes in which a practice-based learning emerges from the sharing of implementation accounts and artifacts without sharing the same context of work.

This study has implication to resource constrained settings in general, and DCs in particular, as they are suffering from similar challenges. The study suggests DCs to embrace OSS, focus beyond the license, and device mechanisms for local developers to participate both locally among co-located peers and globally dispersed developers. The community-based distributed and loosely coordinated OSS implementation model is a viable means to develop indigenous technological capacity simultaneously facilitating IS implementation in DCs. Strong and vibrant participation of local developers both in the global OSS development arena, and local implementation and networking aspects is a requirement and it increases the prospect of realizing the potential of OSS in DCs. The study suggests OSS practitioners in DCs to devise appropriate mechanisms towards this end.

The study also shows the relevance of the trans-situated learning model as an appropriate lens to understand the learning dynamics in OSS implementation projects. The model offers us with the vocabulary both to explore and analyze the learning dynamics evident among loosely coordinated distributed developers/users that are engaged in similar practices. However, further research is required to clearly understand the local universality and embeddedness of the information infrastructure with other infrastructures, as it was not evident in this study and fine-tune it further.

## **ACKNOWLEDGEMENTS**

I am grateful to the anonymous reviewer, Petter Nielsen, and Margunn Aanestad for the constructive comments on earlier versions of this article. I am indebted to HISP and those staff members of AAU and CTIT libraries that participated in this research.

## REFERENCES

- Avgerou, C. (2000). Recognising alternative rationalities in the deployment of information systems. *Electronic Journal of Information System in Developing Countries*, 3(7), 1-15.
- Avgerou, C. (2008). Information systems in developing countries: a critical research review *Journal of Information Technology*, 23(3), 133-146.
- Benbasat, I., Goldstein, D. K., & Mead, M. (2002). The case research strategy in studies of information systems. In M. D. Myers & D. Avison (Eds.), *Qualitative research in information systems: A reader* (pp. 79-99). London: SAGE Publications.
- Braa, J., Monteiro, E., & Reinert, E. S. (1995). Technology Transfer vs Technological Learning: IT - Infrastructure and Health Care in Developing Countries. *Information Technology for Development*, 6(1), 15-23.
- Braa, J., Monteiro, E., & Sahay, S. (2004). Networks of Action: Sustainable Health Information Systems Across Developing Countries. *MIS Quarterly*, 28(3), 337-368.
- bridges.org (2005a). Comparison Study of Free/Open Source and Proprietary Software in an African Context Implementation and Policy-Making to Optimise Public-Access to ICT. Retrieved from [http://www.bridges.org/software\\_comparison/SoftComp\\_Final\\_24May05.pdf](http://www.bridges.org/software_comparison/SoftComp_Final_24May05.pdf)
- bridges.org (2005b). Free/open source software (FOSS) policy in Africa: A toolkit for policy-makers and practitioners. Retrieved from [http://www.bridges.org/files/active/0/FOSSPolicyToolkit\\_10Aug05r.pdf](http://www.bridges.org/files/active/0/FOSSPolicyToolkit_10Aug05r.pdf)
- Brown, J. S., & Duguid, P. (2001). Knowledge and Organization: A Social-Practice Perspective. *Organization Science*, 12(2), 198-213.
- Câmara, G., & Fonseca, F. (2007). Information policies and open source software in developing countries. *Journal of the American Society for Information Science and Technology*, 58(1), 121-132.
- Chan, A. S. (2007). Retiring the Network Spokesman: The Poly-Vocality of Free Software Networks in Peru. *Science Studies*, 20(2), 78-99.
- Ducheneut, N. (2005). Socialization in an Open Source Software Community: A Socio-Technical Analysis. *Computer Supported Cooperative Work (CSCW)*, 14(4), 323-368.
- Feller, J., Fitzgerald, B., Hissam, S. A., & Lakhani, K. R. (Eds.). (2005). *Perspectives on Free and Open Source Software*. Cambridge, MA: The MIT Press.
- Fitzgerald, B., & Kenny, T. (2004). Developing an Information Systems Infrastructure with Open Source Software. *IEEE Software*(January/February), 50-55.
- FMOE – Ethiopia (2010). *Education statistics annual abstract 2001 E.C. /2008-2009 G.C./*. Retrieved 5 July 2010. from <http://www.moe.gov.et>.
- Fox, S. (2000). Communities Of Practice, Foucault And Actor-Network Theory. *Journal of Management Studies*, 37(6), 853-868.
- Gallego, M. D., Luna, P., & Bueno, S. (2008). User acceptance model of open source software. *Computers in Human Behavior*, 24(5), 2199-2216.
- Galliers, R. D., & Land, F. F. (2002). Choosing appropriate information systems research methodologies. In M. D. Myers & D. Avison (Eds.), *Qualitative research in information systems: A reader* (pp. 13-17). London: SAGE Publications.
- Gupta, S. (1995). Development of libraries, documentation and information centres in Ethiopia in the twentieth century. *International Information and Library Review*, 27(4), 317-331.

- Heeks, R. (2002). Information Systems and Developing Countries: Failure, Success, and Local Improvisations. *The Information Society*, 18(2), 101-112.
- Heeks, R., & Stanforth, C. (2007). Understanding e-Government project trajectories from an actor-network perspective. *European Journal of Information Systems*, 16(2), 165-177.
- Hossain, L., & Zhu, D. (2009). Social networks and coordination performance of distributed software development teams. *The Journal of High Technology Management Research*, 20(1), 52-61.
- Klein, H. K., & Myers, M. D. (1999). A set of principles for conducting and evaluating interpretive field studies in information systems. *MIS Quarterly*, 23(1), 67-94.
- Kogut, B., & Metiu, A. (2001). Open-Source Software Development and Distributed Innovation. *Oxf Rev Econ Policy*, 17(2), 248-264.
- Lakhani, K. R., & von Hippel, E. (2003). How Open Source software works: "Free" user-to-user assistance. *Research Policy*, 32, 923-943.
- Lee, A. S. (1989). A scientific methodology for MIS case studies. *MIS Quarterly*, 13(1), 33-52.
- Lungo, J. H. (2006). *Critical Issues Associated with Adoption and Use of Open Source Software in Public Sector: Insights from Tanzania*. Paper presented at the Proceedings of the Fourteenth European Conference on Information Systems.
- Markus, M. L., & Robey, D. (1988). Information Technology and Organizational Change: Casual Structure in Theory and Research. *Management Science*, 34(5), 583-598.
- Mockus, A., Fielding, R. T., & Herbsleb, J. D. (2005). Two Case Studies of Open Source Software Development: Apache and Mozilla. In J. Feller, B. Fitzgerald, S. A. Hissam & K. R. Lakhani (Eds.), *Perspectives on Free and Open Source Software* (pp. 163-210). Cambridge: The MIT Press.
- Monteiro, E. (2000). Actor-Network Theory and Information Infrastructure. In C. U. Ciborra, A. Cordella, B. Dahlbom, A. Failla, O. Hanseth, V. Hepsø, J. Ljungberg, E. Monteiro & K. A. Simon (Eds.), *From Control to Drift: The Dynamics of Corporate Information Infrastructures*. New York: Oxford University Press.
- Myers, M. D., & Avison, D. (Eds.). (2002). *Qualitative research in Information Systems: A reader*. London: SAGE Publications.
- Neumann, P. G. (2005). Attaining Robust Open Source Software. In J. Feller, B. Fitzgerald, S. A. Hissam & K. R. Lakhani (Eds.), *Perspectives on Free and Open Source Software* (pp. 123-126). Cambridge: The MIT Press.
- Nhampossa, J. L. (2005). *Re-Thinking Technology Transfer as Technology Translation: A Case Study of Health Information System in Mozambique* University of Oslo, Oslo.
- Odedra, M. (1992). Is Information Technology Really Being Transferred to Developing Countries? In G. Cyranek & S. C. Bhatnagar (Eds.), *Technology Transfer for Development: The Prospects and Limits of Information Technology* (pp. 47-58). Mumbai: Tata McGraw-Hill.
- Orlikowski, W. J. (1992). The duality of technology: rethinking the concept of technology in organizations. *Organization Science*, 3(3), 398-427.
- Orlikowski, W. J. (1996). Improvising Organizational Transformation Over Time: A Situated Change Perspective. *Information Systems Research*, 7(1), 63-92.
- Orlikowski, W. J., & Baroudi, J. J. (1991). Studying Information Technology in Organizations: Research Approaches and Assumptions. *Information Systems Research*, 2(1), 1-28.
- Pankhurst, R. (1988). Libraries in post revolutionary Ethiopia. *Information Development*, 4(4), 239-245.

- Pollock, N., Williams, R., & D'Adderio, L. (2007). Global Software and its Provenance: Generification Work in the Production of Organizational Software Packages. *Social Studies of Science*, 37(2), 254–280.
- Raymond, E. S. (2000). The Cathedral and the Bazaar. Retrieved from <http://gnuwin.epfl.ch/articles/en/cathedralbazaar/cathedral-bazaar.pdf>
- Sahay, S., & Avgerou, C. (2002). Introducing the Special Issue on Information and Communication Technologies in Developing Countries. *The Information Society*, 18(2), 73-76.
- Scacchi, W. (2005). Socio-Technical Interaction Networks in Free/Open Source Software Development Processes. In S. T. Acuña & N. Juristo (Eds.), *Software Process Modeling* (pp. 1-27). New York: Springer Science+Business Media Inc.,.
- Silverman, D. (1998). Qualitative Research: Meanings or Practices? *Information Systems Journal*, 8, 3-20.
- Silverman, D. (2005). *Doing qualitative research: a practical handbook* (2nd ed.). London: Sage.
- Subramanyam, R., & Xia, M. (2008). Free/Libre Open Source Software development in developing and developed countries: A conceptual framework with an exploratory study. *Decision Support Systems*, 46(1), 173–186.
- Vaast, E., & Walsham, G. (2009). Trans-Situated Learning: Supporting a Network of Practice with an Information Infrastructure. *Information Systems Research*, 20(4), 547-564.
- von Hippel, E. (2002). Open source software projects as user innovation networks.
- von Hippel, E. (2005). Open Source Software Projects as User Innovation Networks. In J. Feller, B. Fitzgerald, S. A. Hissam & K. R. Lakhani (Eds.), *Perspectives on Free and Open Source Software* (pp. 267-278). Cambridge: The MIT Press.
- von Hippel, E., & von Krogh, G. (2003). Open source software and the "private-collective" innovation model: Issues for organization science. *Organization Science*, 14(2), 209.
- von Krogh, G. (2002). The communal resource and information systems. *The Journal of Strategic Information Systems*, 11(2), 85-107.
- von Krogh, G., & von Hippel, E. (2006). The Promise of Research on Open Source Software. *Management Science*, 52(7), 975-983.
- Walsham, G. (1993). *Interpreting Information Systems in Organizations*. Chichester: John Wiley & Sons.
- Walsham, G. (1995). Interpretive case studies in IS research: Nature and method. *European Journal of Information Systems*, 4(2), 74-81.
- Walsham, G. (2006). Doing interpretive research. *European Journal of Information Systems*, 15(3), 320-330.
- Walsham, G., Robey, D., & Sahay, S. (2007). Foreword: Special Issue on Information Systems in Developing Countries. *MIS Quarterly*, 31(2), 317-326.
- Waring, T., & Maddocks, P. (2005). Open Source Software implementation in the UK public sector: Evidence from the field and implications for the future. *International Journal of Information Management*, 25(5), 411-428.
- Weber, S. (2003). Open Source Software in Developing Economies. Retrieved from <http://mediaresearchhub.ssrc.org/open-source-software-in-developing-economies/attachment>
- Weerawarana, S., & Weeratunge, J. (2004). Open Source in Developing Countries Available from [http://www.it-inwent.org/e2484/e3407/e3431/e3432/opensource\\_in\\_developing\\_countries\\_eng.pdf](http://www.it-inwent.org/e2484/e3407/e3431/e3432/opensource_in_developing_countries_eng.pdf)
- Yin, R. K. (2003). *Case Study Research Design and Methods* (3 ed. Vol. 5). London: SAGE Publications.



Zewde, B. (2002). *Pioneers of change in Ethiopia: The reformist intellectuals of the early twentieth century*. Addis Ababa: Addis Ababa University Press.