OSS DRIVEN ORGANIZATION CHANGE IN A DEVELOPING COUNTRY CONTEXT: CASE STUDY FROM THE PUBLIC SECTOR

Nigussie Tadesse Mengesha

University of Oslo, nigussim@ifi.uio.no

Follow this and additional works at: http://aisel.aisnet.org/icis2010_submissions

Recommended Citation
http://aisel.aisnet.org/icis2010_submissions/134
OSS DRIVEN ORGANIZATION CHANGE IN A DEVELOPING COUNTRY CONTEXT: CASE STUDY FROM THE PUBLIC SECTOR

Completed Research Paper

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

Abstract

This paper investigated the organizational changes that were unfolded during the implementation of an OSS-based IS in the public sector of a developing country (DC) context and the ramifications of aspects of OSS and contextual matters on the process and its outcome. To do so, the study draws upon IS literature that views technological change as an emergent property of the complex interaction between ISs and organizational contexts, and the notions of organizational routines to explore the changes. The result of the interpretative case study suggests that although the license, openness of source code, and community (open) participation in implementation offers unprecedented opportunities for developers in DCs to develop technological capacity and collaboratively customize and implement software, these aspects could also be sources of challenges and could facilitate for routines to be extended unnecessarily in turn affecting roles and relationships. The study suggests that focusing on the notions of organizational routines helps to understand and explain organization changes and the complex interaction between the social and the technical in a specific context. Finally the study recommends points to bypass some of the challenges.

Keywords: Organization change, developing country, information system implementation, open source software
Introduction

The emergence of OSS has opened up new opportunities for developing countries (DCs) to acquire, study, adapt, and redistribute software (Li et al. 2004; Weber 2003; Weerawarana and Weeratunge 2004). In principle, adoption of OSS reduces license cost and total cost of ownership (Waring and Maddocks 2005); promotes indigenous technological development (Weber 2003; Weerawarana and Weeratunge 2004); avoids being hostage to proprietary software and vendors (Weber 2003); guarantees against buried “espionage software” (Weber 2003); advances knowledge more quickly (Câmara and Fonseca 2007); promotes adaptation to changing organizational environments (Gallego et al. 2008); and helps to set up an information economy (Weber 2003; Weerawarana and Weeratunge 2004). In practice however, technology transfer to DCs in general, and implementation of ISs, often brought from the West in particular, have been challenging and the failure rate was very high (Heeks 2002; Walsham et al. 2007). IS implementation in DCs has been challenging due to the complexity of ISs (Braa et al. 1995; Heeks 2002), impediments in DCs such as lack of resources and techno-scientific capabilities (Avgouropoulos 2000; Meso et al. 2009; Odedra 1992), and country context differences which may lead to the development of ISs that are embedded with rationalizations, assumptions, and world views of the place of production demanding adaptation to use contexts (Akrich 1992; Heeks 2002; Orlikowski and Barley 2001). The implementation of OSS-based ISs in DCs is not immune to these challenges. Of course, the OSS philosophy has shifted the practice of software development and implementation from organization-based to individual-driven and from closing to opening source code (Feller et al. 2005).

The study of OSS often focuses on various aspects of development projects such as motivations, governance structure, modularity, socialization, etc (Ducheneaut 2005; von Krogh and von Hippel 2006). Little attention has been given to the implementation aspect, especially the implementation of domain specific applications such as library information system (LIS) in general, and in DCs in particular (Waring and Maddocks 2005). Given the complexities of IS implementation, hindrances in DCs, and the altered philosophy and practice of IS development/implementation as a result of OSS, how organizations in DCs cope with OSS-based IS implementation? What are the effects of the altered philosophy and practice of software development and implementation on organization change? These are some of the questions which the OSS literature did not give much attention so far and, which this research tends to address focusing on the latter. The study explored the unfolded changes and ramifications of the OSS philosophy and practice on organization change during the implementation of an OSS-based IS in a public sector organization in a DC. The study was conducted in a public sector academic library in Ethiopia called Addis Ababa University Library (AAUL). The library introduced an OSS-based LIS (OLIS) called Koha1 in 2004 and has been using the cataloging and OPAC (online public access catalog) modules ever since while additional module implementation was going on. The implementation of Koha created a new socio-technical order in AAUL that has implications both to the OSS and organization science literatures.

The rest of the paper is organized as follows. The next section explains the analytical framework adopted in this research by reviewing relevant literature on OSS in DCs and technological change in organizations. The research context and methods are presented in section three followed by a presentation on the cataloging work practices before and after the introduction of Koha and the planned cataloging in section four. Section five presents the analysis followed by presentations of the discussion and conclusion in section six and seven respectively.

OSS in Developing Countries and Organization Change

OSS in Developing Countries

Studies indicate the relevance of IT/IS to support the development agenda of DCs simultaneously emphasizing the challenges of realizing them (Heeks 2002; Heeks and Stanforth 2007; Walsham et al. 2007; Walsham and Sahay 2006). IS implementation in DCs has been challenging because of resource limitations (financial, infrastructural, and

1 Koha is an OSS distributed under the open-source General Public License (GPL), initially developed in New Zealand and first deployed in 2000. Further information is available at http://www.koha.org/ last accessed 15 April 2010.
The emergence of OSS, however, has brought lots of promises ranging from bypassing financial limitations through facilitating adaptation and integration to changing the predominately developer–consumer relationship between the developed and DCs in terms of software production and use. The OSS license dictates disclosure and dissemination of source code so that anyone with proper programming skills and motivations could study, use, modify and distribute any OSS written by anyone (Feller et al. 2005; Neumann 2005; von Hippel and von Krogh 2003). OSS development relies 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 et al. 2005). 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 a face-to-face communication, especially in the community-based OSS model, which is the concern of this research (Lakhani and von Hippel 2003; Mockus et al. 2005; von Hippel 2002; von Hippel 2005). Shorter time gap between OSS releases which not only address bugs but also incorporate additional features and functions, added with the availability of source code paves the way for continuous change.

Although OSS alleviates some of the challenges of software implementation in DCs such as related to acquisition, adaptation, learning, etc, it also poses difficulty as it shifts the implementation responsibility primarily to users (or individuals local to use contexts) that have limited or no techno-scientific knowhow in the case of DCs. In Ethiopia, the context where this research was conducted the idea of OSS was little known among IT professionals until 2005 – until the establishment of EFOSSNet (Ethiopian Free and Open Source Software Network), an OSS advocacy group. A workshop which was organized by EFOSSNet and the Ethiopian IT Professionals Association (EITPA) on 27 December 2008, revealed the immaturity of the OSS idea among IT professionals three years after the establishment of EFOSSNet. The changed practice of software implementation can have different ramifications on organization change than the other models but attention has not been given to it so far. The OSS literature focused mainly on development projects, developers’ motivations, governance structure, etc of back-end applications such as operating systems, middleware, and server software. The literature in DCs focused mainly on the challenges and opportunities of OSS (Weber 2003; Weerawarana and Weeratunge 2004). Less is known about the implementation of domain specific applications in the public sector of DCs except Lungo (2008). Lungo’s study however focused on an in-house developed software and a software brought from abroad where the core developers offered field supports to local implementers and users (Lungo 2008). Lungo did not consider the community-based OSS model. Besides, as far as knowledge of the researcher is concerned, the role of OSS on organization change is not yet studied. Thus, this research investigated the changes that unfolded during the implementation of an OSS-based IS and the ramifications of the OSS philosophy and practice on organization change drawing upon the notions of organizational routines and the emergent perspective on technological change.

**Organizational Routines and Technological Change**

The IS literature lacks reliable generalization about the relationship between IT and organization change due to researchers backgrounds (Markus and Robey 1988) and selective and partial focus on the aspects of technology (Orlikowski 1992). Markus and Robey (1988), for example, identify a variety of theories based on causal agency, logical structure and levels of analysis. According to Orlikowski (1992) the knowledge about technology in organization is ambiguous and conflicting because of researchers’ selective and partial conceptualization of the scope and role of technologies. Technology has been conceptualized as “hardware” and “social technology”, and to have deterministic and emergent effects on organization change. The technology imperative research conceives of technology as the primary and relatively autonomous material cause or driver of organization change with a
deterministic effect (Monteiro 2000; Orlikowski 1996). The adoption of technology, according to this viewpoint, creates predictable changes in organizations' structures, work routines, information flows, and performance (Barrett et al. 2006; Orlikowski 1996; Orlikowski and Barley 2001). This conceptualization ignores the role of human agency in shaping the design, development and use of technology, and the proactive role of human agency in organization change (Akrich 1992; Orlikowski 1992; Orlikowski 1996; Orlikowski and Barley 2001). The constructionist perspective, on the other hand, assumes almost unlimited choice over technological options and almost unlimited control over the consequences to human action (Akrich 1992; Markus and Robey 1988; Orlikowski 1992; Orlikowski and Barley 2001). This set of research is criticized for giving more emphasis to human agency and denying the material affordances and constraints of technology. The emergent perspective questions the viability of the constructionist and materialist perspectives in light of the current organizational discourses that emphasize emergence, flexibility, and self-organization.

The emergent perspective proposes the idea that neither a strictly constructionist nor a strictly materialist stance are adequate to study technologies in the workplace. It holds that the uses and consequences of information technology emerge unpredictably from complex socio-technical interactions (Ciborra and Hanseth 2000; Cordella 2006; Markus and Robey 1988; Orlikowski 1992; Orlikowski and Barley 2001). In the contemporary world, the notion of emergence is particularly relevant because, writes Orlikowski, “unprecedented environmental, technological, and organizational developments facilitate patterns of organizing which cannot be explained or prescribed by appealing to a priori plans and intentions” (1996, p. 65) (emphasis in the original). The emergence view posits technology as an external force having impacts but where these impacts are moderated by human actors and organizational contexts. Orlikowski and Barley explain that

\[...\] adequate accounts of technological change require hybrid explanations that weave together human action and choice, the functions and features of specific technologies, and the contexts of a technology’s use in a way that attends to the micro-dynamics of situated practice (2001, p. 150).

The micro-dynamics of change and stability in organizations can best be explained by considering organizational routine as a unit of analysis (Becker 2004). Organizational routines can be defined as “repetitive, recognizable patterns of interdependent actions, carried out by multiple actors” (Feldman and Pentland, 2003 p. 95). Routines are recurrent patterns that consist of action, activity, behavior, and interaction, which also involve multiple actors (see Becker 2004 for a comprehensive review). Feldman and Pentland unpacked the constituent parts of routines to have an ostensive (or structural) and performative (or agentic) aspects following the notions of the structuration theory and examined the changeability of routines (Feldman 2000; Feldman and Pentland 2003; Pentland and Feldman 2005). The ostensive aspect is the “abstract, generalized idea of the routine, or the routine in principle” (Feldman and Pentland, 2003, p. 101), while the performative aspect “consists of specific actions, by specific people, in specific places and time … the routine in practice” (ibid). Later works of Feldman and Pentland included the notion of IT but as part of the context within which routines are executed (Pentland and Feldman 2005; Volkoff et al. 2007). The investigation into changes in organizational routines led Volkoff, Strong, and Elmes to identify the existence of a material aspect of routines that is embedded in technologies (Volkoff et al. 2007). Unlike other artifacts, IT is an integral part of routines and gives material aspect to them and as a result could change roles, data, relationships and mindset and organizational culture (Volkoff et al. 2007). Therefore, in order to analyze the changes that were emerged as a result of the implementation of an OSS-based IS, this study draws up on the notions of organizational routines and considers the three aspects such as ostensive, performative and material as described here.

**Research Context and Methods**

**The Context**

Addis Ababa University Library (AAUL) is a public academic Library composed of more than 19 branches. It was the major information resource center in Ethiopia, particularly, for research and academic purposes. The Library serves mainly AAU, the oldest and the largest higher learning organization in Ethiopia, and extends services to external organization and individual users. The Library was organized in a centralized manner, i.e., it performs technical and managerial activities and placed the card catalogs of all of the branch libraries (or the union catalog) centrally in the main library. A university librarian was responsible for the overall operation of the Library with the help of two assistant librarians each responsible for the technical and public services. The technical services include acquisition and cataloging while the public services include circulation, reference, departments and branch libraries.
The acquisition department acquires books and other materials through purchase, gift, exchange, and donation and sends them to the cataloging department after making all the necessary checks and putting property stamps on each material. The cataloging department then catalogs and dispatches them to branch libraries. Branch libraries collect cataloged materials along with corresponding catalog cards; shelve cards and books in the right shelf in the correct order; and offer circulation and reference services to users. They are also responsible for maintaining catalog cards and ordering books and other materials for purchase in consultation with faculties. Most of these activities were performed manually and had limitations to satisfy users, and facilitate operations and communications within and outside of the library (Mengesha in press). In order to alleviate some of these problems, the library tried to acquire and introduce computer-based LIS through purchase and in-house development at different times but did not succeed. Finally, the library started implementing Koha, an OLIS, at the end of 2004 and, by the end of 2008 it was making use of the cataloging and OPAC modules. The OPAC allows users of the library to search for and locate library materials, while the cataloging module aspires to automate the cataloging routine. The cataloging function was the focus of this study as it was the only module of Koha that was serving the library staff. The cataloging function included cataloging books2, maintaining the union catalog and the shelf list, inventorying books, announcing newly cataloged books each month, and providing information on plans and performances of the department to relevant authorities (section four explains the cataloging function in detail).

AAUL was introduced to the OSS phenomenon through a failed collaboration among one university in the USA (a professor and his students), an NGO (co-established by the professor), AAUL and ICTDO (ICT Development Office) of AAU. The aim of the collaboration was to customize and implement an OSS LIS called OpenBiblio but it discontinued after two months of operation (Aug – Sept 2004) without bearing the desired result. The occasion however introduced the local team to the world of OSS, which was not well known among IT professionals in the country at the time and even until 2008. It was under these circumstances that AAUL implemented Koha; in fact AAUL has been considered to be the pioneer in implementing OSS in Ethiopia.

Methods

The research adopted the qualitative research approach (Silverman 2005) with the underlying epistemological and ontological notions of the interpretive philosophy (Klein and Myers 1999; Orlikowski and Baroudi 2002; Walsham 2002) to collect and analyze data. In order to make sense of the changes, it was desirable to consider the previous cataloging routine, the envisaged cataloging, features and functions of the new library system (Koha), the implementation context and the resulted cataloging routine. To do so, detailed data collection was conducted through semi-structured interviews, review of reports and memos, and observations at the research sites. The researcher was employee of the Library for about eight years until the end of 2002 and served as manager of the library automation project from August 2000 to August 2004. The membership allowed him to gain access to all recorded documents such as reports, minutes and memos. The researcher’s library science and IT background by training and experience facilitated the sense making processes of the cataloging routine, feature and functions of Koha and their interplay. Besides, he conducted interviews and observations from February to April 2007 and from November 2007 to February 2008 for a different research. The previous research revealed the existence of perception differences among key stakeholder groups in the library with regard to the library’s adoption of OSS (Mengesha in press). The management and technologists perceived OSS as an opportunity that would minimize total cost of software ownership and a means for building knowledge hub that would serve AAUL and other libraries in Ethiopia. Users (catalogers and other librarians) however perceived it from the perspective of expertise requirement and third party involvement in the implementation process, and were against the adoption of OSS. These findings led to further investigations into the learning/sharing and collaboration processes and the changes that could emerge as a result of OSS implementation (this research).

Thus, a fieldwork specifically aimed at this research was carried out from September to December 2008. During this period a total of 16 interviews were conducted with 10 respondents, their duration varying from 1 to 1:30 hours. Most of the interviews (ten of them) were tape-recorded, and detailed notes were taken for the rest of the interviews. All of the respondents were either users of the new system (catalogers), technologists/developers, managers of the Library or project managers that had significant influence on the implementation or operation of Koha. The data

---

2 Besides books, the library acquires periodicals, maps, audiovisual materials, etc but their total size is less than 5 percent of the whole collection and the procedure followed to catalog them is almost similar except few differences. As a result, the process of cataloging books is considered in this case study.
collection was meant for exploring the initial assumptions of technologists/developers, users and managers concerning the planned role of Koha, the cataloging routine before and after the introduction of Koha, and exploring the implementation process. Through iterative reading, the collected data was classified to understand the cataloging routine that was practiced before and after the introduction of Koha, the planned cataloging, the cataloging routine inscribed into Koha, and the unintended consequences of implementing Koha in AAUL. The contents of these routines were analyzed and compared with one another to understand their similarities and differences and, hence, the changes. These similarities and differences were interpreted in terms of the three aspects of organizational routines (ostensive, performative, and material). The findings were further analyzed to understand the relationship between the changes, the implementation context and the aspects of OSS such as license, code openness, and the implementation model. 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.

**The Old and the New Cataloging Routines**

The cataloging department catalogs all library materials that the library acquires centrally in the main Library and dispatches catalog cards and books to branch libraries for use. Both the cataloging department and branches perform subsequent activities of shelving and maintaining cards similarly. The cataloging department had four categories of staff – bibliographic checkers, catalogers, typists and pastors. The head of the department was responsible for overall operation and management of the department and reports to the assistant university librarian in charge of technical services on plans and performance of the staff and the department at different frequencies. Branches had at least one person responsible for shelving and maintaining catalog and shelf list cards both in the shelf list and public catalog in each branch. The shelf list card is a catalog card produced for internal use by catalogers, it is not accessible to the public, and each title has one shelf list card if it belongs to the main library, two otherwise. The catalog cards are produced for public use, to facilitate access to collections of the library. The next sections discuss the cataloging routine before and after the introduction of Koha and the routine that was envisaged by the project.

**The Manual Cataloging – the Old Cataloging**

After receiving books from the Acquisitions department, Bibliographic Checkers (BCs) make sure that they all bear the authority stamp of the Library and are not mutilated. Then they verify whether a book is new to the library or an already existing one (table 1 describes the old cataloging in detail). To do so, BCs bring books in trolleys to the public area where the union catalog was located (where users of the library also search for books) and look for a matching bibliographic record. If a matching record was found, BCs write an “add to” message on the book to guide the typist to add copy number and receiving date on the shelf list card later. If a matching record was not found, the book was regarded as new and undergoes either copy or original cataloging process. Following the verification, BCs handover books to catalogers and typists accordingly.

Copy cataloging involves adapting an already existing bibliographic data of a book from a given source to requirements of a library. Due to space and financial constraints to purchase catalog cards from abroad, AAUL limits the size of data in a card and the number of cards per book. The library subscribes to a database called Bibliofile from the Library of Congress in the USA for copy cataloging. Bibliofile contains bibliographic details of newly published books and journals in CDs; it is updated every year and offers various functions including searching, displaying, editing, printing, exporting data, etc. The major tasks of catalogers would be to search for and locate books’ data in the database, localize it according to AAUL policy and print the required number of shelf list and public catalog cards, if found. Otherwise, the second option would be to feed the CIP (cataloging in publication) data of a book into an empty data entry form in Bibliofile and produce appropriate number of cards, if the book has a CIP data. If the data of a book is not available in Bibliofile and if it does not have CIP data, catalogers perform original cataloging, i.e., they prepare catalog cards from scratch by analyzing the physical and content of a book. They fill out the data of a book in an empty paper form by analyzing a book and later feed it into an empty data entry form in Bibliofile and produce appropriate number of cards.

Once cards are produced, catalogers verify books’ data against corresponding catalog cards, correct them in case of errors, and finally deliver them to typists for physical preparation. Typists prepare book label, book pocket, book card, and due date slip and a pastor pasts them on each book. Finally, the book is ready for dispatch to branch
libraries and the cataloging department informs branches to that effect. Usually representatives from branches and the acquisitions department fill out the book exit form together, crosscheck and deliver books to respective libraries.

Besides cataloging, maintaining catalog cards that involves filing (alphabetically by author, title and call number in different card catalog files), replacing worn out cards, correcting erroneous cards, removing cards when the book is discarded and relocating them whenever there is relocation of books are the duties of catalogers. Each branch library also performs these activities separately. Each cataloger also produces a cataloger report that shows the number of books and periodicals he/she has cataloged on a monthly basis. Typists also produce reports that show the number of new title, added copies, number of card, and number of pockets produced by each typist. The department produces the list of newly acquired and catalogued books each month arranging by faculty, which is called accession list, and inventory the stock at different frequencies. Each cataloger was responsible for cataloging the books of a given faculty though their size and the size of books they purchase are different.

Table 1: Original, Planned and Emerged Cataloging Routines due to the Implementation of Koha in AAUL

<table>
<thead>
<tr>
<th>ROLES</th>
<th>ORIGINAL ROUTINE</th>
<th>PLANNED ROUTINE</th>
<th>EMERGED ROUTINE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bibliographic checkers</td>
<td>Receive books from acquisitions and verify their newness to the Library. To do that carry books to the public area and check their bibliographic detail against the union catalog one by one.</td>
<td>Catalogers carry out the activity using Koha without bibliographic checkers.</td>
<td>Bibliographic checkers have been checking the availability of a book using Koha.</td>
</tr>
<tr>
<td>Search for matching records in Bibliofile, adapt data and print cards according to AAUL policy. Proofread cards and correct errors if any. Send books and cards to typists for physical preparation. Verify cards for error or mismatch of data after physical preparation and correct it if any.</td>
<td>Avoid production of card catalog and automatically feed Koha from Bibliofile.</td>
<td>Besides the original routine, catalogers add the detail of each processed book into an intermediary database, later import into Koha, add more information such as assign barcode and holding data, verify and confirm the entered data, and save into Koha database. Print barcodes and paste them on to books, and deal with barcode generation problems and the problems of the cataloging module.</td>
<td></td>
</tr>
<tr>
<td>If a matching record is not found in Bibliofile and the book has CIP data, copy the data into an empty Bibliofile form and conduct the rest procedure similar to the copy cataloging stated above.</td>
<td>Directly feed the CIP data into Koha, if the electronic data of a book is not found in Bibliofile.</td>
<td>The same except where the data will be stored. Instead of entering the data into the Bibliofile empty form, it will be entered into Koha’s form.</td>
<td></td>
</tr>
<tr>
<td>Perform original cataloging if bibliographic data of a book is not found in Bibliofile and if it does not have a CIP data. Scan the physical and content of the book; consult the Library of Congress Subject Heading (LCSH) list for assigning subject headings and the Library of Congress classification scheme for determining the call number; fill out the cataloging slip (a piece of paper prepared for the purpose); and copy the data into an empty Bibliofile form and conduct the rest procedure similar to the copy cataloging procedure stated above.</td>
<td>The same as the original.</td>
<td>The same except where the data will be stored. Instead of entering the data into the Bibliofile empty form, it will be entered into Koha’s form.</td>
<td></td>
</tr>
<tr>
<td>Inform receiving libraries when reasonable number of books is ready for collection. Fill out part of the material exit form and handover the books to the person assigned for the collection.</td>
<td>The same as the original.</td>
<td>The same as the original.</td>
<td></td>
</tr>
<tr>
<td>File cards alphabetically by author, title, subject or call number in public catalog and shelf list shelves. Replace worn out cards, correct erroneous cards, and remove and relocate cards when a book is discarded and relocated.</td>
<td>By abandoning card catalog use avoid the tasks of filing and maintaining cards.</td>
<td>The same as the original.</td>
<td></td>
</tr>
<tr>
<td>Produce a report listing the title of books cataloged per month. Inventory the Library’s collection as schedule by the Library.</td>
<td>Automatically produce such reports with few mouse clicks.</td>
<td>Cut required data from Bibliofile and paste into an excel worksheet or word.</td>
<td></td>
</tr>
</tbody>
</table>
**The Planned Cataloging**

The automation project targeted at minimizing or avoiding the time and effort required to verify the availability of books in the library, edit bibliographic records, produce accession list, and maintain card catalogs both at the public area and the shelf list (table 1 describes the planned cataloging in detail). Card catalog maintenance such as filing, identifying, correcting, removing and relocating was cumbersome, error prone and time consuming. The project also planned to automate the works of typists, which was to select data from the bibliographic data of a book and type it on separate papers called book pocket, book card, and book label.

A book pocket and a book card are specifically designed to facilitate the manual processes of borrowing/lending books. In a computerized library system, book pocket and book card may not be required. The project planned to abandon all activities pertaining to card catalog – such as production, filing and maintenance. Because, once the

<table>
<thead>
<tr>
<th>ROLES</th>
<th>ORIGINAL ROUTINE</th>
<th>PLANNED ROUTINE</th>
<th>EMERGED ROUTINE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Typists</td>
<td>For an already existing book, fetch the shelf list card, and add copy number and receiving date. For new books, on all cards add the first letters of the first and second names of the cataloger and the typist, the date in which the material is cataloged, type of material: “per” for periodical and “ref” for reference, and a letter “S” and copy number if the book has more than one copies on shelf list cards. Prepare book pocket, book card, book label and due date slip based on the data on the shelf list card. Produce statistical report that indicates the number of new titles, added copies, book pocket, book card, book label and due date slip produced on a weekly basis.</td>
<td>The role of the typist is not required in a fully automated cataloging; therefore phase out the typist tasks step-by-step.</td>
<td>As usual for the typist but catalogers were required to add similar data on the new system to reflect the change in the number of copies.</td>
</tr>
<tr>
<td>Pastors</td>
<td>Paste book pocket at the back verso of the material, due date slip on the page opposite to where the book pocket is pasted, and the book label on the spine or the bottom left corner of the book. Insert the book card into the book pocket.</td>
<td>In a computerized library system, book pocket and book card may not be required, and hence, their production.</td>
<td>The same as the original routine.</td>
</tr>
<tr>
<td>Head, Cataloging Department</td>
<td>Produce accession list that lists all the materials acquired and cataloged by the department disaggregated by Faculty and subject on a monthly and yearly basis. Produce annual report concerning the accomplishment of the department.</td>
<td>Allow head of the cataloging department to generate accession list automatically.</td>
<td>The same as the original routine.</td>
</tr>
<tr>
<td>End Users (concerning card catalog usage)</td>
<td>End users should go to the public card catalog area of each library to search for books. Users in the main library can search for a book in all of the libraries but they are required to visit the desired library to fetch the required book. If the search is successful, users go to the stack area, handover the detail of a book to a staff, and a staff fetches the book if it is available. Otherwise, users should ask the circulation desk if the book is on loan or it is under maintenance. The required book might be lost or misplaced if still not found. If you find a book, users should go to the circulation desk for borrowing the book.</td>
<td>Search and locate materials in any of the AAU libraries 24/7 from a desktop using keywords, subject headings, title, authors, editors, etc or by combining them. When the circulation module is up and running users will check the status of a book and perform some circulation activities.</td>
<td>Users interact with computer to search and locate books. They can also search for a book either in all or one of the libraries using a number of access points or a combination of them. The detail of data provided to users on a given book has also increased.</td>
</tr>
<tr>
<td>Deployment team/ Computer Center</td>
<td>Provide IT related support the Library and conduct IT R&amp;D activities.</td>
<td>Customize, implement, and maintain Koha.</td>
<td>Assure Koha runs smoothly, enhance and localize it, &amp; implement the rest modules.</td>
</tr>
<tr>
<td>Branches (regarding cataloging)</td>
<td>File cards alphabetically by author, title, subject or call number in public catalog and shelf list shelves. Replace worn out cards, correct erroneous cards, and remove and relocate cards when a book is discarded and relocated.</td>
<td>By abandoning card catalog use avoid the tasks of filing and maintaining cards and this role from branches.</td>
<td>The same as the original.</td>
</tr>
</tbody>
</table>
The implementation of Koha in AAUL changed the practice of cataloging and the way users interact with the library would also be carried out easily from a terminal. By avoiding card catalog, the project planned to evade the amount of hard currency incurred to purchase catalog cards, book label and other papers from abroad, abandon the roles of card catalog filing and maintenance from branch libraries, and reduce the time and effort of catalogers and the error thereof. Editing bibliographic records in the Bibliofile involves manually deleting and adding similar data elements for each book. The project planned to automate this process so that selected data could automatically be fed into Koha.

Since every branch library houses the catalog that reflects its own holdings except the main library that houses the union catalog, it was difficult for users to know the holdings of other libraries unless they go to that specific library and search the catalog. With the emergence of interdisciplinary and multidisciplinary programs, the demand for visiting multiple libraries had increased. Besides the limited number of access points (such as subject, author, title, etc) and the number of cards printed per book, the card catalog lacks the flexibility to combine access points. There is a general understanding that the higher the number of access points and the possibility to combine them for searching, the higher the chance for retrieving/fetching the relevant book from the entire collection.

Deployment of an ILS planned to offer the ability of searching for and locating materials in any of the branches 24/7, increasing the number of access points such as keywords, subject headings, title, authors, editors, etc and the ability of combining them together and perform some circulation activities to end users. The project also planned to automatically produce statistical reports that the catalogers and typists used to compile manually – such as accession list, cataloger report, and typist report.

**Cataloging with KOHA – the New Cataloging**

The library started customizing Koha after previous attempts to purchase proprietary software, develop in-house and adapt another OILS in collaboration with another partner were failed to bear fruits. The justification behind switching from purchase to OSS adoption were 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 an LIS that would run on 25 workstations each for staff and users. The quoted price excludes adaptation costs that may be incurred to make the system able to accommodate materials written in Amharic script.

In the end of 2004 the Library in collaboration with the ICT development office (ICTDO) of the University embarked on a joint OILS adoption project and organized a project team comprised of the library and ICTDO staff. The alliance sought to exploit the technical expertise of the IT personnel (of ICTDO) both in customization and implementation, and developing the technical capacity of the library staff. However, the partnership was discontinued after the pilot phase and ICTDO trained some of the computer center staff of the Library on the technologies of Koha. These trained people (developers hereafter) were later reorganized as the deployment team members and took the responsibility of customization, installation and configuration of Koha in AAUL. The disassociation of ICTDO from the project caused reinstallation and reconfiguration of Koha on the library’s server that was 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 overflow by tasks related to database management system, operating system and Koha itself.

The developers customized and implemented Koha with the support of the Koha community and its documentations. The developers posted inquiries about the design of Koha, installation and configuration issues, fixing bugs, switching operating systems, upgrading to new releases, and interfacing Koha with other systems among others and, made use of the responses. They also studied the email archives of Koha’s developers and users mailing lists and other documents to implement Koha in AAUL.

The implementation of Koha in AAUL changed the practice of cataloging and the way users interact with the Library (Table 1 presents the new routine in detail). Users of the Library could search for books using computers (and Koha) either in one or all of the libraries using a number of access points or a combination of them. The detail of data provided to users on a given book was increased because of the insignificant cost of adding further data per book. The bibliographic checkers no more carry books to the public card catalog area and manually compare records to verify whether a book is new to the library or not. They were doing it from their desktops using Koha.
The implementation of Koha increased the workload of catalogers and nothing was changed for the typists. Besides conducting the previous cataloging routine, catalogers were required to feed Koha with bibliographic data, assign barcode numbers, add holding data, and verify and confirm entered data. After producing card catalogs using the old system, catalogers add the detail of each processed book into an intermediary database, and later import and save into Koha database. As an additional task, they print barcodes, paste them on to books and deal with barcode generation problems and the problems (bugs, improvements, missing functions, etc) of the cataloging module of Koha in general.

Koha did not help catalogers to produce statistical reports such as the titles of newly cataloged books in a given month, the number of books cataloged by a cataloger, and accession list. They were aware of the possibility and flexibility of producing reports once data is captured. Moreover, they were not getting favorable responses to bugs; often bugs remained unfixed altogether; they operated another function of Koha that they were not allowed to do so; etc. All of the catalogers had information about the smooth running of Koha in another local library and some of them had paid a visit. Most of the catalogers were doubtful of the smooth running of Koha in AAUL and insist on using the old system instead. The following two excerpts exemplify their frustrations:

The good side of Koha is that you cannot leave the call number field blank; in the rest, 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. ….

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.

The working relationship between the cataloging department staff and the developers started to deteriorate when inquiries of the catalogers failed to get favorable answers. As a result catalogers started to question the knowledge and expertise of the developers. The following excerpt reflects some of their anxieties:

… we were individually approaching the deployment team members for solutions when we encounter any problem. … Due, in part, to the negative reactions of the developers and delayed solutions, if solved at all, we abandoned directly reporting problems to them but through our coordinator.

I used to report problems with the view to improving Koha but I stopped doing it after I was regarded as a person who is trying to ruin the implementation. … The error reporting process destroyed the smooth relationship between us and those who are responsible for debugging.

I think the technical people did not have the required knowledge and skill. That is why they refuse to give us favorable response to our requests. … We know that Koha is running smoothly in another library and I would suggest them to ask for support.

Furthermore, Koha did not favorably support collaboration between catalogers. If a cataloger assists fellow cataloger, the assisted had to collect catalog cards and then either type or retrieve the detail of each cataloged book from Koha and cut and paste it on MS word or Excel worksheet to prepare reports. There was a time gap between making the detail of a book available online and making it ready for service in a given library because the place where the book is processed and used were separate. It took some time before reaching the user library and nothing denotes this time gap. This might misinform users as the detail of a book is visible to users despite readiness the book for use. In another respect Koha has been policing unnecessary requests. Branch libraries occasionally requested for the re-cataloging of books and complain to users that they did not receive books while they actually have it. Koha arbitrates such conflicts mostly in favor of catalogers.

Due to the above reported problems catalogers developed negative attitude towards Koha, the developers and the implementation process and ultimately the full-fledged utilization of Koha. The implementation process also affected the working relationship between branch library heads and the project manager who later was promoted to the assistant university librarian position. Some of the branch heads regarded their usual duties as par-time and engaged in other activities that were nothing to do with the duty they were paid for. The relationship made them not to be accountable to their responsibilities. One of the branch heads said “some of us considered the other work as our primary duty. It was a good opportunity for us to increase our income but heavily impacted the library’s work.”

For the developers and the project manager, some of the functions that the catalogers required such as producing card catalog will soon be obsolete and including it in Koha was not advisable. They explain that even the use of Bibliofile will cease when the Internet bandwidth allows for faster downloading and uploading over time.
Furthermore, the circulation module was not yet automated and demands production of card catalog and related cards. While implementation of Koha was going on, the university annexed independent public colleges and institutes that dictated continuation of previous work in parallel with Koha beyond the expected duration according to the project manager. He also relates interfacing and other technical problems to the knowledge and skill of the developers. New stable versions of Koha were released 11 times since it was introduced in AAUL but AAUL managed to change three times only (from 2.2.9 to 2.2.4). Since the middle of 2007 the project was ceased and dealing with Koha became the normal routine of the computer center that was reorganized and staffed for the purpose (see table 1 for the detail).

**Analysis**

The implementation of Koha in AAUL created a new socio-technical order that was unfolded overtime as a result of the interaction between Koha and the socio-technical ensembles in AAUL. The cataloging routine that was practiced after the introduction of Koha was different from the old, the envisaged and the inscribed (into Koha) cataloging as described in table 1. The implementation of Koha changed the cataloging routine, relationships and mindsets of the staff. This section analyzes these changes and the roles of aspects of OSS (license, openness of source code, community-based implementation model) and contextual factors in shaping the unfolded changes.

**Changes in the Cataloging Routine**

The implementation of Koha in AAUL required the harmonization of contents of the cataloging routine inscribed into Koha, the actual work practice and the planned cataloging. The higher the gap among the contents of these routines, the higher the change and the risk of failure would be especially in a DC context (Heeks 2002). In AAUL, the ostensive aspect of the cataloging routine, the routine in principle, i.e., the process of making books easily accessible to users remained stable. As usual, cataloging receives books from acquisitions, prepares means of access, delivers to branches, maintains the means of access on a regular basis and produces various reports. However, the specific practices of making books accessible to users, the performative aspect of the cataloging routine, were changed. Koha changed the old performative aspect of cataloging by giving material forms to the different roles and activities of the cataloging routine, digitizing the paper-based data, and streamlining embedded activities with the influence of contextual elements and the nature of the software.

The planned routine abandoned the use of cards, and hence, filing and maintenance activities and the staff involved in these activities both at the main and branch libraries by giving material forms to these activities and roles. BCs could verify the existence of a book’s detail in the library from a desktop without the need for carrying them to the public catalog area. Doing so saves money and resources that would otherwise be spent on cards, personnel, equipments and related activities. The routine inscribed into Koha also considered this model of cataloging except the mechanism of capturing data. Koha automatically fed its database by fetching records from online sources and required an Internet connection with a reasonable speed while the planned cataloging demanded interfacing with Bibliofile, i.e., except the data sources the two routines gave material forms to the activities of searching and importing records. The role of a cataloger upon receiving a book would be to retrieve its record, adapt the record according to the AAUL policy, produce barcode and paste it on the book, and add the record into Koha’s database. For books the record of which was not available in Koha’s bibliographic database, Koha was supposed to automatically search and download the record from online sources (in case of Koha) or from Bibliofile (in case of the plan). The procedure for adding the record of a book into Koha was the same in both cases if the record of a book was not found either in online sources or in Bibliofile – it is a manual process in both cases. The plan demanded the production of different reports, while Koha was able to produce only some of them. Both the planned cataloging and Koha gave material forms to the roles of BCs, typists, and pastors.

The difference between the routine inscribed into Koha and the planned cataloging therefore was small. However, there was a huge difference between the two systems (Koha and the plan) and the old cataloging routine. As described in table 1, BCs verified the existence of records of a book by carrying them to the public catalog area and searching for a matching record of each book. Then, catalogers search and download the record of a book from Bibliofile (by inserting and removing at least two Bibliofile CDs into a PC CD drive for a book), adapt it according to the AAUL policy and print the required number of cards if the record was found. Otherwise, they copy cataloging in publication (CIP) data of a book into an empty Bibliofile form and perform the rest activities as the previous one. In case a book did not have such detail, catalogers analyze the physical and content of the book and fill out the detail on a sheet of paper, then input to an empty Bibliofile form. Typists prepare book pocket, book card, book label and
due date slip for each book to facilitate circulation, while pastors paste them in an appropriate location in the book. The cataloging department staff produces report on individual activities at different frequencies to the head of the department who in turn produces summarized reports to the assistant university librarian.

Although the difference between the old and the other two routines (Koha and the plan) was huge, it would be of great advantage to the efficiency and effectiveness of the routine had either Koha or the plan were implemented properly. However, the resulted routine, the performative cataloging was a mixture of Koha and the old routine, or simply, Koha was running on top of the old routine without any automatic interaction with artifacts and activities of the old routine. For example, Koha was not able to deal with card catalog production; did not automatically interact with Bibliofile; and did not produce required reports. The failure of Koha to give material forms to these activities made it neither stand by itself nor able to automatically interact with the old routine. The failure of Koha to produce card catalog necessitated for the use of Bibliofile, the old system, and its inability to automatically interact with Bibliofile necessitated for the intervention of catalogers (manual activities). The interaction between Koha and Bibliofile was mediated by an intermediary database with manual activities of exporting and importing data to and from these databases.

The manual procedure of exporting and importing data from and to the three databases, and performing both the old and the new systems in parallel extended the practice of cataloging, demanded extra efforts of catalogers, and affected their mindset. Furthermore, bugs in the cataloging module, late delivery of fixes and dissatisfaction with functions, severely affected the relationship between developers and catalogers. As a result, catalogers challenged the expertise of the developers and developed negative attitude towards the use and full-fledged implementation of Koha. However, catalogers were happy with an unprecedented function of Koha that arbitrated conflicts with branches in their favor. The implementation increased the size and type of data that can be captured per individual book, increased access points and changed the location of data and the type of storage device. Book data was simultaneously kept electronically and on paper.

The project tended towards modifying the old cataloging routine to resemble the routine inscribed into Koha due to the influence of the plan and the opportunity provided by Koha. For example, even if there was demand for card catalog production, developers did not make Koha able to deal with card catalog due to the plan to abandon cards in the near future and lack of expertise. Again, partly due to the opportunity provided by Koha to abandon the use of proprietary database for cataloging, developers gave less priority to interfacing Koha with Bibliofile even if it was the plan to do so. Furthermore, various contextual elements such as availability of techno-scientific knowhow, limited knowledge and experience of local developers with OSS and the specific technologies of Koha, resource and infrastructure constraints, lesson from previous failed attempts, the demand for card catalog production, etc, contributed a lot towards the result. For example, even if the plan abandoned cards, the developers tried to interface Koha with Bibliofile and make Koha able to deal with card catalog production mainly to meet the demands of catalogers but without success. The changed philosophy and practice of software implementation also affected the process and contributed its share towards the result. The following section analyzes the effects of the license, openness of the source code and the community implementation model on the unfolded changes.

The Role of aspects OSS in Organization Change

Openness of source code, the license and the community model of implementation (open participation) are the core characteristics of OSS. These characteristics had both constraining and enabling ramifications in the implementation of Koha in AAUL and shaped the implementation process and the outcome differently than what would have happened had it was closed code (proprietary) software. The OSS license offered the library to acquire Koha free of charge and gave the freedom to study, modify and implement it (Neumann 2005; von Hippel and von Krogh 2003). The community model facilitated for developers and users (of Koha) around the world to involve in the implementation of Koha in AAUL. The local implementation process attracted and made use of the involvement of local and international developers and users. The developers in AAUL downloaded Koha, studied its designs and technologies, posted inquiries to the Koha community, made use of responses, studied the email archives of both the developers and users mailing lists, upgraded their knowledge and skill of Koha and related technologies, and finally, implemented Koha. They also benefited from local IT experts that trained them. These results suggest that the license, openness of the source code and the community model of implementation set an environment conducive for local developers to freely acquire OSS, and learn technologies and the tricks of installation, configuration, customization and fixing bugs among others. These enabling aspects of OSS were also sources of challenges in AAUL.
Implementation of ISs in organizations is a socio-technical process that involves both the social and the technical and where both are supposed to evolve together (Cordella 2006; Heeks 2002; Monteiro 2000). The outcome of the process would be an order negotiated between the two in a specific context in time. The implementation outcome and reactions of users of Koha in AAUL suggest the influence of local circumstances and the aspects of OSS. The AAUL were using technologies that were obsolete in the developed world such as card catalog; it was not using an automated system; it cannot make online financial transactions because it was not yet started in Ethiopia; and the library required facilitating access to materials written in the Amharic script. Furthermore, the Internet bandwidth was so limited that downloading a record from online sources located abroad was difficult. Making Koha to download records from abroad therefore was futile and increasing the bandwidth of the Internet connection was beyond the capacity of the library. Besides, Koha did not deal with card catalog and the Amharic script, and the library could not abandon the use of cards. These requirements necessitated for further streamlining activities such as making Koha able to deal with card catalog and the Amharic script, and to automatically interact with Bibliofile. These contextual elements simply demanded giving material forms to some of the cataloging activities. Lack of techno-scientific knowhow in general, and OSS in particular among IT professionals of the country including the AAUL developers was another contextual element that contributed towards the unfolded changes. The AAUL developers were novice to OSS and the technologies of Koha. Furthermore, the philosophy and practice of OSS, especially the community-based implementation model, shifted the responsibilities of localization and implementation to these inexperienced and less skilled local developers. As a result, embedding the functions of dealing with card catalog, Amharic script and Bibliofile into Koha became difficult. Furthermore, lack of technical expertise and the plan to implement an ideal system, led the project to give less emphasis to intermediate steps of solution provision. Finally, manual activities of importing and exporting data to and from three databases, the use of the old system to deal with card catalogs, doing extra activities due to the new system such as pasting and dealing with bugs, etc extended the cataloging routine.

AAUL was using obsolete technologies; did not use automated systems; cannot conduct financial transactions online; the Internet bandwidth was limited; needed to deal with locally produced materials; lacked techno-scientific knowhow, OSS and its technologies; planned to implement an ideal system, etc. These contextual elements demanded working around the cataloging routine inscribed into Koha but due to the aspects of OSS and some contextual elements, effecting the required changes were challenging and as a result the performative aspect of the cataloging routine was extended.

**Discussion**

The implementation of Koha in AAUL resulted in a new socio-technical order that was determined neither by the social nor the technical but the complex interactions between the social and the technical with the influence of contextual elements and feature and functions of the software. Some contextual elements dictated for the adaptation of the new system to the use context while others hampered the process. For example, the need to continue working with card catalog and the limited Internet bandwidth demanded adaptation of Koha so that it automatically deals with card catalog production and interacts with Bibliofile. However, due to lack of techno-scientific knowhow, and limited knowledge and experience of local developers with OSS and the specific technologies of Koha, it was not possible to effect the changes. The nature and feature of the software played constraining and promoting roles in the implementation process and shaped the outcome accordingly.

The noninvolvement of license cost for OSS opened up new opportunities for individuals and organizations to acquire, study, modify and utilize software (Gallego et al. 2008; von Hippel and von Krogh 2003). The adoption of OSS also avoids the often lengthy bureaucratic negotiations and associated corruptions apparent in the public sector. Unlike donated ITs that ended up in locked rooms in DCs due to lack of local skilled personnel, problems with western consultants and ineffective capacity development strategies (Odedra 1992), OSS offers new opportunities and mechanisms for capacity development, localization and implementation. Openness of source code coupled with open participation paves the way for developers to study source code, learn technologies and the tricks of customization and implementation through practice and with the active involvement of developers and users around the globe regardless of geographical, organizational or time barriers. The email archives of developers and users also provide significant support to solve technical problems in the course of implementation. Openness of source code facilitates for integrating a given OSS with others as well. Furthermore, OSS makes it easy to make use of changes made elsewhere and continuously improving systems. Improvisation in OSS is supported by frequent releases that would not only solve technical bugs but also introduce new features and functions. Thus, openness of source code,
the license and the community model of implementation (open participation) open up a window of opportunity for learning, localization, overcoming financial constraints, making use of international resources, etc.

Although code openness, OSS license and open participation facilitate learning and capacity development, these could also be sources of implementation challenges depending on the context. As various studies have pointed out, the success of implementation, utilization, scalability and sustainability of technologies brought from abroad depend very much on the existence or the development of local capacity (Braa et al. 1995; Heeks 2002; Odedra 1992). Different arrangements have been made to this effect. For example, in the case of proprietary software, vendors often customize and implement systems, provide technical support after implementation, and train end users and technical experts on how to operate systems, administer them and solve technical problems. Such arrangements do not exist in the case of community-based OSS implementation as described in this case study. The community-based OSS implementation practice demands local users/developers to play the anchor roles in all activities of localization, implementation, enhancement, etc. In the context of DCs, it means that the major responsibility of customization and implementation rests upon resource constrained organizations and inexperienced developers in an environment where techno-scientific knowhow in general, and knowledge and experience with OSS (in this particular case) are lacking.

The implementation of ISs in DCs, especially those brought from the West including OSS, demand translation to use contexts due to country context differences and the gap between design assumptions and realities in implementation contexts (Akrich 1992; Heeks 2002). The translation in turn demands resources and techno-scientific knowhow. For example, the AAUL was required to adapt Koha due to some peculiar characteristics such as, for example, the need to deal with card catalogs and limitation in the Internet bandwidth. However, due to lack of expertise and the plan to move directly to an ideal system by phasing out the old system, materializing the required adaptation was not materialized. These contextual elements (with the support of the changed implementation practice) dictated for the old and the new systems to run in parallel with each other for longer periods. As a result, the cataloging routine was extended to fulfill the requirements of the old and the new systems and interaction between them, i.e., the introduction of Koha extended performative aspect of the cataloging routine.

The performative aspect of the cataloging routine was extended (changed) partly due to the inability of developers to give material forms to some of the required activities such as interfacing, report production and dealing with card catalogs. Some of the activities are core and center in the cataloging routine while others are in the periphery accounting to prototypical and non prototypical categories of routines as suggested by Tsoukas and Chia (2002). Conceptual categories have radially structured members where the center is occupied by more representative, stable and prototypical members that account for the stability with which the category is often applied (Tsoukas and Chia 2002). There is an unstable part, consisting of nonprototypical members, which accounts for the potential change in a category. For example, robins are more central to our understanding of the category "bird," than ostriches are. Similarly, routines in organizations have stable core and peripheries that derive their meaning from the broader web of background assumptions, experiences, and understandings shared in a culture (Feldman 2000; Tsoukas and Chia 2002). The extended performative cataloging routine included activities that lied in the periphery, non prototypical activities such as manually importing and exporting data to and from databases. The extended performative routine affected the relationship between developers and those who were performing the activities. The catalogers undermined the knowledge and skills of the developers and developed negative attitude towards the full-fledged implementation and use of the new system. These changes and the implementation also affected roles and relationships.

Even if, in principle, OSS paves the way for continuous change, it could be hampered by local circumstances such as availability (or not) of expertise. Heeks (2002) argues that developing countries may not be able to improvise systems and organizations mainly due to resource limitations and local circumstances and insists upon onetime appropriation and implementation. However, the onetime implementation could be problematic in an environment where the interaction between the technology and its context is a continuous process. This study demonstrates the difficulty of implementing ISs at once leaving the rest aside as stated by Heeks (2002), especially in a resource constrained setting.

**Conclusion**

The research reported in this paper investigated the organizational changes that were unfolded due to the implementation of an OSS-based IS in a DC context and the ramifications of aspects of OSS and contextual matters on the process and its outcome. To do so, the research draws upon the emergent perspective on the role of
technology in organizations and the notions of organizational routines. The research designed as an interpretative case study analyzed the contents of the old, the planned, the inscribed (into technology), and the emerged routines, the implementation context and the nature and functions of the software in order to make sense of the changes, and their ramifications. By attaining to the interaction among human action and choice, functions and features of the system, the implementation context and the resulted socio-technical order, the study indicates that aspects of OSS (code openness, license and community-based implementation) could both promote and constrain implementation in DCs. These aspects of OSS set an environment conducive for local developers to freely acquire OSS, learn technologies, solve local problems in collaboration with developers around the globe, and implement software. However, the community-based implementation makes resource constrained organizations and unskilled and inexperienced developers in an environment where the idea of OSS was not well known (at least in the case at hand), to be in charge of customization and implementation processes. The shift in responsibility together with the requirement for adaptation of the system to the use context coupled with lack of techno-scientific knowhow unnecessarily extended the existed routine (work practice).

The emerged changes were mainly related to the difficulty of giving material forms to some of the activities, data and roles, i.e., embedding them into the technology, and streamlining the existing ones. As a result of the difficulty, the specific activities of the cataloging routine were extended to include additional ones that can be considered as unnecessary. The extensions affected roles, relationships and mindset of those who were performing activities. Further to the skill problem, the plan that tended towards implementing a radical change without considering a step-by-step transformation played significant roles in the process. The organizational changes that resulted due to the implementation of OSS-based IS, following the community-based implementation model in a DC context where resources and techno-scientific knowhow is limited could be different from other models, for example, for proprietary system implementation because of skill and expertise differences, and differences in implementation approach.

This study shows both the facilitating and constraining ramifications of aspects of OSS on OSS-based IS implementation processes and organizational changes in a DC context. The challenge would be to overcome its constraining effects and making the best out of it. The study also enriches the OSS literature by considering the under explored implementation aspect of end-user OSS-based IS. Furthermore, it empirically shows the relevance of the notions of routines in investigating organizational changes and the complex interactions between the social and the technical in a certain context. Considering organizational routines helps to understand the micro-dynamics of situated practices thereby helping to grasp the complex interaction between the social and the technical in a specific context with the influence of the nature and function of the software.

Acknowledgement

I would like to thank the track chair, the associate editor, the two anonymous reviewers, Eric Monteiro, and Margunn Aanestad for their constructive comments on earlier versions of this article. I am also indebted to HISP and those staff members of AAU library that participated in this research.

References

Mengesha, N.T. "The Role of Technological Frames of Key Groups‘ in Open Source Software Implementation in a Developing Country Context," *Electronic Journal of Information System in Developing Countries* in press.


von Hippel, E. "Open source software projects as user innovation networks," 2002.


