THE CHALLENGE OF "TRANSLATING" HEALTH INFORMATION SYSTEMS FROM ONE DEVELOPING COUNTRY CONTEXT TO ANOTHER: CASE STUDY FROM MOZAMBIQUE

Nhampossa, José Leopoldo, Department of Informatics, University of Oslo, Norway, leopoldo@ifi.uio.no

Abstract

What does it take for an open source, Not-for-Profit, software developed in one context to be internationalized and localized so as to be used in another context different from its origin. This question is addressed in the frame of a Health Information System application developed in and for South Africa and subsequently transferred to use in Mozambique.

Through an action research effort, five sets of key challenges to cross-country translation process have been identified: (i) language rules and lack of Portuguese equivalent terms from English, (ii) length of strings, (iii) different naming conventions, (iv) different organizational structures, and (v) inadequate knowledge. The understanding of these challenges helps us to identify the different features of translation associated with "general purpose" and "special purpose" applications. The analysis helps to address the question of how a "pragmatic balance" can be obtained between the needs for creating internationalized products on one hand, and that for providing flexibility for local adaptation on the other hand.

Keywords: Multilanguage systems, language translation, localization, internationalization, globalization, general business domain, special business domain, healthcare, Mozambique, South Africa.

1 INTRODUCTION

This paper deals with the problem of how can an open source software designed and developed for supporting Health Information Systems (HIS) in a particular developing country (South Africa) be localized and adapted for use in another developing country (Mozambique). The challenges in this localization process are multiple and complex ranging from the problem of language translation (English to Portuguese) to adapting it to the very different contexts of use in the health sector of the two countries.

Primary Health Care (PHC) in developing countries provides an interesting domain for the study of this translation process. In this paper, the term "translation" has been used more broadly than just referring to language translation but to also include adapting the software to the different contexts of use, for example related to varying health organization structures. The World Health Organization (WHO) has stipulated various standards around how HIS should be organized in various developing countries. For example, the district has been designated as the hub for all information management activities. This implies that data from the health units are sent to the district where it is aggregated and then sent to the province and national levels. Also, most countries have multiple donor-supported vertical programs operating, creating the similar need for integration of these systems to enhance effectiveness of the HISs. There is also the need for the HIS to enable the calculation of health indicators, for example of immunization coverage, some of which are standardized across countries. While this element of standardization through the WHO directives may give the impression that the HIS from one context can be easily translated to another, however, in practice this translation is very complex, given the very different socio-political-cultural contexts, which shape the HIS in distinctive ways. Having said that, however, there are certain elements in HIS which are indeed common, and can be taken from one context to another. So, while there is no need to fully "reinvent the wheel" of HIS design and development, sensitivity to the contextual differences and how they shape the HIS in different countries is essential. This need is in line with Rolland and Monteiro's (2002) argument for the need to find the "pragmatic balance" between the pressures for building global standardized system on one hand, and to allow for flexibility to localize and expand the system on the other hand. The focus of this paper is to analyse the practical challenges of translating a HIS designed and developed for South Africa to Mozambique. An analysis of this process helps to shed light on the simultaneous processes of internationalization and localization of HIS, and the tensions that exists.

The rest of the paper is organized as follows. The key theoretical concepts are presented in the second section while the following two sections describe the methodology and the case study. The fifth section describes the challenges experienced during the empirical work of software translation. The sixth section focuses on the analysis and discussion of the findings drawing upon the relevant theoretical concepts. Finally some concluding remarks are presented in section seven.

2 THEORETICAL CONSIDERATIONS

The issue concerning the design, development and implementation of software to be used across different, national and cultural boundaries can be usefully analysed with respect to processes of internalization, localization and globalisation. Internationalization refers to the process of isolating the culturally specific elements from the software and building a system for use in different countries (Russo, 1993; (O´Donnell, 1994)). Normally this process occurs in the country where the product is originally developed and is typically limited to translating text and date, time, and number formats, following specific guidelines (Russo, 1993). This processes is largely limited to language translation, or inserting some locally relevant icons and colours, but largely confined to the level of the user interface. Localization, in contrast, refers to the process of infusing a specific cultural and business context into a previously internationalized product (Taylor, 1992; O´Donnell, 1994). Like internationalization, localization is normally limited to translating text, message files, date and number

formats. Globalisation involves global corporations seeking to develop suitably standardized products and practises that can be used across the globe. These efforts at standardization are typically in tension with the need for localization and become visible in the process of translation.

Translation of software can be of two types. One dealing with software for General Business Domain Application (GBDA), and the other for Special Business Domain Application (SBDA). GBDA refers to general purpose software like spreadsheets and text processors. In this, the functionality, content and the interface is largely decided by the software vendor, and it is relatively easily used across organizations, countries, contexts and cultures. Improvements, changes and evolution of the software are driven by the vendor to maximize the time efficiencies for testing and debugging whilst expanding the market scope.

The SBDA software is more application focussed, and thus the translation process requires a greater understanding of the business domain and the context of use. Specific meanings of terms and concepts are important to understand as they are linked to particular business rules. To enable this understanding, it is important for end-users and systems developers to work collaboratively with each other. In SBDA the development process is facilitated in such an environment where the systems developers are pulled from the context of design into the context of use, and are therefore confronted with the incompleteness of current understandings of complex cultural settings (Gregory, 1995). The translation processes of SBDAs are thus more complex and require greater time and investment of end users as compared to GBDA.

In this paper, the translation process of a SBDA software related specifically to the domain of PHC in developing countries is discussed. This translation process is different from most examples described in the literature for at least two reasons. One, the software is open source and not for profit. Two, the software is designed within the context of public sector. While these differences create challenges related to the availability of resources and skilled personnel, it also helps to open up new opportunities since project maximization is not the driving motive. This study thus has potentially important contributions to make to both the understanding of open source software development, and to the design and development of HIS in developing countries.

3 METHODOLOGY

The work presented here is based on research carried out in the Health Information System Project (HISP) in Mozambique from 2000 to 2003. HISP is an ongoing international endeavour aiming to study and introduce district-based HIS in various developing countries (Braa et al., 2001; Braa et al., 2003). DHIS is an open source software meaning that not only the user has access to the all source code, but also has the ability for free redistribution and reworking of the source code (Braa and Hedberg, 2000).

The research efforts were aimed at the design, development, and implementation of the DHIS in Mozambique, and were enabled through a 6 member multidisciplinary team for translating the DHIS. The author of this paper was a member of this team comprising of medical doctors, computer scientists, and Ministry of Health (MoH) staff. Being the computer scientist, the author was overall responsible for the entire translation process. Data collection sources included discussions within the team, participating in training workshops, study of documents, and focussed interviews with other members of the team, and also with the head of the DHIS software development team in South Africa.

The process of design and development of DHIS can be described as an action research project (Susman and Evered, 1978) consisting of a set of cyclical and interleaved steps. The focus is on the interventions that the researcher takes together with the users in order to improve a phenomenon at the same time as studying it (Kalleberg, 1995). Susman and Everd present their action research approach in a diagram as depicted in the figure 1 below.

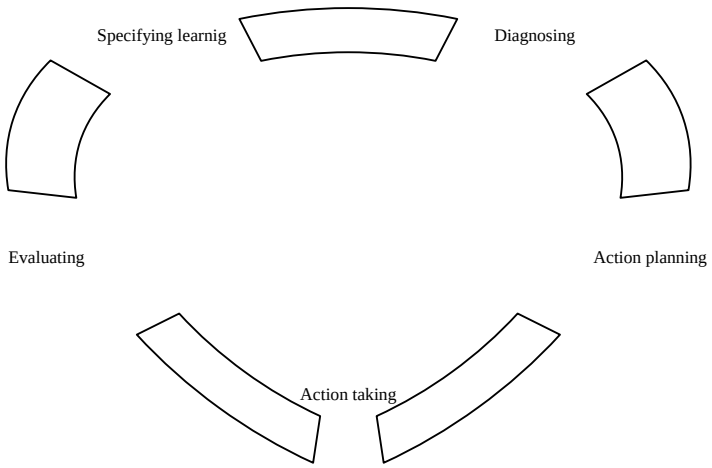

Figure 1. Susman and Evered's (1978) cyclical process of action research.

The localization and translation processes described in this paper are similar to the steps described by Susman and Evered. In the case, the diagnosing phase consisted of understanding the health information needs of the health department and how the DHIS could help support it. The action planning stage involved setting up the multidisciplinary team and establishing the strategy for DHIS localization. The action taking phase involved using the prototyping methodology for developing and using various translated versions of the software. Evaluation consisted of testing the DHIS in the pilot sites, and taking necessary correction action in "the specifying learning" phase. Through these iterative cycles there was thus learning by doing and changes emerged as a result. This approach is different from traditional software process models, like the waterfall approach that do not allow for construction until the specification phase is completed. Such approaches are problematic in conditions where specifications are unstructured, ambiguous, keeps changing, and the requirements are very user dependent rather than being system dependent. The PHC sector represents such an unstructured domain.

The starting point of the localization process was the DHIS software designed in South Africa for the needs of the South African health system. The system and all relevant documentation were in English. For example the user interface for the South African DHIS is depicted in figure 2 below.

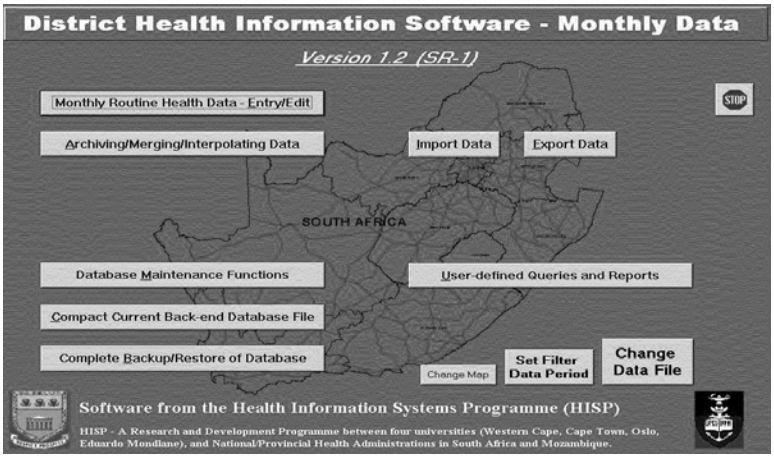

Figure 2. The original user interface of the DHIS software.

The DHIS consists of several modules and data files. The "Routine monthly data" module – allows data entry, verification and analysis of monthly data from the PHC and hospital services. The tuberculosis (TB) module enables entry of routine TB data entry, verification and analysis. The Report Generator module is used for generating and accessing reports on any of the data files. The Client Satisfaction Survey module is used to capture and analyze the client satisfaction survey data in FRONTENDS. These modules represent the different user interfaces for monthly routine health data, tuberculosis data, client satisfaction survey and data reporting.

The database files, representing the BACKEND of the DHIS stores information about data elements, data element categories, indicators, definitions, organizational unit data, and semi-permanent data for example related to population. The database files can be classified according to the organizational structure of the health system. In Mozambique, the structure consists of four levels, and the database files correspondently have four instances: National, Provincial, District and Health Unit.

For each database module there is (or can be generated) a corresponding MS-Excel spreadsheet module-pivot table that is used for data analysis. Here data can be visualized and handled in pivot tables and used to build standard or customized graphs for different purposes, for example to see the immunization coverage for different time periods.

DHIS contains tools for data quality control which allows for the checking of the quality of data entered by setting the minimum and maximum ranges for all the data elements. A validation check can be done once a facility's data has been entered for the month.

Indicators, community data, and data dictionary are three other features of DHIS. Indicators are defined and handled according to the numerator, denominator and indicator type. These form different indicator groups (for example, district or province level indicators) that are the source for the pivot tables in the Excel spreadsheet. Community data features provide the definitions for the different semi-permanent data elements, such as population groups and targets. Population data is entered for each district catchment area according to the different population age categories. The Data Dictionary is a Web based application storing the nationally approved names and definitions for all the data elements that are in common use throughout the country.

The DHIS installation CD includes the user manual, additional support software tools including various Microsoft service releases/packs. Basically all the complementary and supplementary resources available on the DHIS setup CD are in English and for English versions of Microsoft Windows operating system and Office.

HISPML, the HISP multilanguage library, is a separate database module storing the text strings visualized in different user-interface screens of the current FRONTENDS. This module makes it

possible to translate DHIS to all MS WINDOWS 2000/XP supported languages, whereby by selecting the locale, the text strings are automatically visualized in the different screens. The text strings are basically stored in three categories of sources: GLOBAL DICTIONARY (1), BACKENDS (1 or more) and MS-ACCESS FORMS (192 or more).

The entire development of the DHIS is being conducted by a development team based in South Africa. Along with producing new versions of the DHIS as they are continuously developed, the team is also responsible for providing technical support to the team engaged in localization in different countries, as in Mozambique.

After this brief description of the technical features of the DHIS, and how the development is organized, the case study concerning its localization process to the Mozambique context is now described.

4 THE CASE: TRANSLATION AND ADAPTATION OF DHIS

There was consensus among the HISP team in Mozambique to initially implement for testing purposes only the Routine Monthly Data module of the DHIS. This module was seen to be critical as it provides the set of data collection and reporting procedures to handle monthly routine health data. So the initial use of DHIS was to computerize the existing paper system and basic procedures from the current National Health Information System, called SIS. In doing the translation, skills and expertise was required from the domain of computer science, medicine, and public health. Although none of the HISP team members were native English speakers, and neither did they possess any prior experience in language translation in general and of software in particular, the first prototype was primarily an output of the combined effort of this team, supported by officials from the Ministry of Health.

The adaptation process was conceptualized in two parallel tasks: Language translation and development of the Mozambican BACKEND. A CD with the first monolanguage version of the DHIS software was installed on a portable computer. Translating the software interface was the starting point of the DHIS so as to provide the users with a greater sense of familiarity and with it more ownership of the system.

The fact that the strings were part of the code made it difficult if not impossible to install the Portuguese version of DHIS on other computers, meaning the translation process was restricted only to the hosting computer where the piloting process was being carried out. During a training session, for example, all the facilitators and available participants had to sit around the one portable computer that hosted the Portuguese version of the DHIS.

Basically all the changes to DHIS software were introduced in the same portable computer, which was then needed to be sent back to the South African development team for the production of the setup CD. The new DHIS was then sent back to Mozambique, making it possible to distribute it among different people, and install on other computers for use and further testing.

The number of language problems reported magnified as the number of users increased. The strings inscribed in DHIS software were generated in South Africa and were not context free. The translation in a different context is naturally complex, requiring greater face-to-face interaction between Mozambique and South Africa. However, this was difficult to achieve, due to geographical distance and limited resources. Therefore the interaction with the South African team was limited to electronic mail and some limited telephone calls.

As an example of the kind of electronic communication exchanged, is provided below this excerpt of an interaction between Mozambique and South Africa.

Communication facilitated through e-mail, between the head of DHIS software development team in SOUTH AFRICA and myself:	
Mozambican side	South African side
• Thanks for sending me the updated version of the standard hardware/software for HMIS document. • I find it a very useful document indeed. Do you mind if we translate part of the doc and use it to update the MoH policy document regarding this issue? If yes we aim to distribute it to all provinces and district health authorities.	• Feel free to translate, use, or re-distribute it as you see fit. When translating it into Portuguese, I would also suggest you adapt it to fit Mozambique (replace some SA-specific comments, use local prices etc). • In particular, I would insert a few extra paragraphs: Emphasise the need for Portuguese keyboards. I see you are using XP, and that the message is "failure to create registry key". It's too late for me to test it on my XP disk now, but check two things: 1. Verify that the decimal separator is set to. (ot) and the thousand separator to a comma. 2. Verify that you have logged into windows XP as a user with admin privileges (or at least have authority to make modifications to the registry).

Table 1. Electronic communication between development and localization teams

The new version received from South Africa in Portuguese was not compatible with the existing local Mozambican database/BACKEND that contained the data elements, indicators, organizational unit data, definitions, etc. So using the new version required the database to be built from scratch. One more language translation cycle took place, and the required changes were introduced in the portable computer and, to produce the setup CD the computer was sent again to South Africa. The implementation of every single change was synonymous to totally deleting the previous work since the new versions were not compatible with the local backend. These repeated cycles of changes, transfers to South Africa and not optimal design significantly impeded the progress of the project. These problems led to feedback to the head of development team in South Africa to separate the text strings from the software and associate them with a particular language module. In parallel, a checklist with a set of DHIS problems or bugs and proposals for improvement was developed in Mozambique and submitted to the South Africa development team. This formed the starting point for the birth of the idea of shifting DHIS from mono to a multilanguage platform.

A multilanguage version of DHIS was subsequently developed in South Africa which was capable of supporting several languages including Portuguese, Norwegian, Russian, and Spanish. In the multilingual DHIS, the caption strings, data definitions, and indicator strings could be translated and used with new coming versions. This new version of the DHIS was obtained from South Africa and installed in various personal computers in Mozambique. The HISP team verified to what extent had the problems listed in the checklist been fixed in the multilanguage version of DHIS. It was seen that the organizational schema of the strings suffered from great alterations, creating the need to correspondingly change the translation approach along with the adaptation. The strings in the FRONTEND – representing the visible part of the DHIS - and the ones in the BACKEND – representing the invisible part of the DHIS were approached differently when translating. The strings in the HISPML library were divided among the team members and translated to Portuguese, while the BACKEND was translated or rebuilt in parallel. To refine the translation, and make sure inputs from everyone involved in the localization process was taken into account, several stakeholder meetings were organized to discuss in detail, string-by-string and line-by-line the meanings and appropriateness of the strings or sub-strings used. It included sometimes starting up the computer and the DHIS software and clicking on the strings corresponding buttons to see the linkage to the functions and concepts behind the visualized text in the screen.

5 CHALLENGES EXPERIENCED DURING THE TRANSLATION PROCESS

As the DHIS was translated into Portuguese, it was tested in the pilot sites for the first time. Feedback on problems were reported to the HISP Mozambique (HISPMZ) team by the users during the piloting exercises and training sessions organized at various sites. Five sets of key challenges were experienced:

- Language rules and lack of Portuguese equivalent (see Griffiths et al., 1994) terms from English,
- Length of Strings,
- Different naming conventions,
- Different organizational structures, and
- Inadequate knowledge.

These challenges are now described separatelly.

5.1 Language rules and lack of Portuguese equivalent terms from English

The present English language computer vocabulary was invented when the need arose when the item or concept was created (Barbour, 1996). Terms like backup, zoom, and data mart do not have direct translation in Portuguese. In this case, the team was forced to perform a partial or intermediate translation, mixing English and Portuguese text. This hybridization of terms often created problems of interpretation for the users.

The dictionary (as a starting point) can be a good input for the translation if aligned with knowledge about concepts, meaning, language rules (which are specific to languages) and contexts of use. These meanings can be effectively provided only by people who are conversant with the language rules, cultural, context and business rules. The dictionary or the machine can only provide for translation of the standard concepts but not of the culturally specific meanings. The dictionary thus can help in the translation of simple strings but not of strings of strings. Since the meanings of strings are linked to broader cultural and business understandings, the dictionary provided with the installation CD was inadequate.

For example, consider the command: printf("%s %s", string1,string2);

The machine will access the corresponding text for strings 1 and 2 in the string tables and translate automatically. But if the text order has to be changed there will be a problem.

If string1 is for “yellow” and string2 is for “house” the translation for Portuguese could result in “amarela casa” instead of “casa amarela,” as is the rule in Portuguese.

The lack of Portuguese equivalent terms is also illustrated in the following excerpt related to what I call “partial” or “intermediate” translation.

> *DHIS users of non-English versions of Windows and Office should download the correct Service Releases/Packs from the relevant Microsoft web-site.*

This quote is from the DHIS user manual. The decision whether or not to use the English or Portuguese version of MS Access is needed to be taken locally based upon the ease to download the correct service releases/packs from the internet. Accessing the releases/packs from the internet requires (i) availability of internet connection and a (ii) reasonably fast link. Such a downloading exercise could last for hours or even days in Mozambique. Therefore, in practical terms, we decided to install the translated Portuguese DHIS version using the English MS Access, as visualized in the Figure 3 below. This intermediate or partial translation again created confusion and ambiguity for the users.

When prompted, the user had to click on the Tools menu, Database Utilities and Compact ad Repair Database batons. The translated Portuguese version appeared as follow: *Clicar no* Tools, *Ferramentas*

de base de dados, Compactar base de dados. So the user is confronted with a situation where she or he is prompted to click buttons which are not visible on the screen. In this case, the translation is adding more confusion instead of facilitating the dialog between the user and the system. This situation remains the same even in the English version of MS Access because of the absence of specific technical terms in languages other than the original, in this case English. This ambiguity is reflected in one of the questions asked to the HISP team about the possibility of using the Portuguese version of WINDOWS and OFFICE. The answer was:

Ideally yes, but because of possible bugs on the Portuguese versions of MS Access, no.

The DHIS functions in the Microsoft Access environment. It would be thus advisable to use the Microsoft Access in Portuguese in Mozambique, and other countries were Portuguese is the official language to avoid the appearance of English and Portuguese strings in the same screen.

In addition to the complexity of translating Portuguese terms, similar challenges were experienced in finding the "right" equivalent for English-based terminology used in health, medicine or epidemiology.

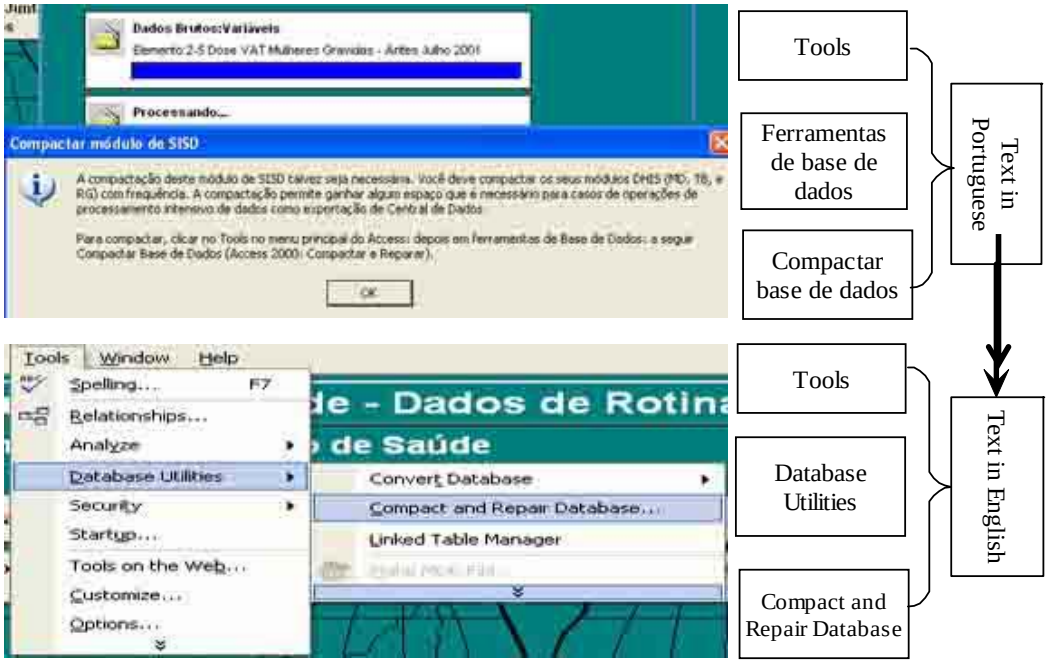

Figure 3. English version of MS Access interface development environment versus Portuguese version of DHIS leading to intermediate or partial translation: Mix of English and Portuguese text.

5.2 Length of Strings

The translation raised issues related to the length of strings as Portuguese equivalent were much longer than those used in English. This issue of length had implications for the user interfaces, the description and distribution or location of the different buttons, the layout of the screens and quality of the video adapters. Consequently, in order to keep the "correct" translation, the buttons for example had to be enlarged and located in different positions. Or the long strings had to be simplified in order to keep a reasonable layout and distributions of the buttons. For example the translation of "backup" will be *Cópia de segurança*, which did not fit in the original user interface button. In this case, the button had to be expanded with knock down implications on the design of the Menu.

5.3 Different naming conventions

The Mozambique health system suffers from the problem of different and inconsistent naming conventions of the different organizational units even though located in different provinces. For example, it is possible for a health unit in Maputo and Niassa to both have the same name of Eduardo Modlane. The naming convention thus needed to be changed, based on consultations with health authorities.

5.4 Different organizational structures

The hierarchical organizational structure of the health system in Mozambique is different from South Africa. In South Africa, there are five levels including National, Province, Region, District and the Facility. In comparison, Mozambique has one less level as there are no health regions. This implied adding a dummy organizational layer to the Mozambican organizational structure to allow for the compatibility of levels (Kaasboll and Nhampossa, 2002).

5.5 Inadequate domain knowledge

Translation required computer skills, understanding of medical terminology, knowledge of application domain and experience on translating software. The translation took place under extreme time pressure as the project needed to show quick results generated from a usable DHIS prototype to the MoH officials. For reasons of expedience, the translation of the monolanguage DHIS was therefore performed focusing more on the technical terminology and aspects from the computer point of view, rather than on developing the "correct" meanings of technical health terminology. As the English strings were hard coded, the translation took place in a traditional way, using the tools available in the MS-Access editor (cut, copy, replace, paste). However, this technical focus led to improper meanings of terms that caused problems for the users.

Language problems were the most critical due to lack of understanding of the terms visualized on the user interface and linked to specific functions of the DHIS. For example, the string data element was translated *as elemento de dados*, but on testing we found that the meaning was distorted by the pure text translation performed by people lacking expertise in technical health terminology. *Variável* for variable is in this case the correct translation according to health workers. It was possible to translate back the expression but this implied stopping the DHIS program and switching to design view, making the required changes, saving and restarting. In general this could happen several times during a training session leading to interruption of the training session.

6 ANALYSIS AND DISCUSSION

The fact that initially the DHIS strings were hard-coded significantly contributed to the challenges in translation. This is because the DHIS was not originally designed with features and code to accept international conventions, foreign data, and format processing. Miller (1994) argues that building internationalization into a [software] minimizes or eliminates the need for engineering revisions [as happened with DHIS monolanguage], and greatly simplifies the localization process and reduces the time lags inherent in localizing software for other contexts. For example, it was assumed and expected that the translation would be trustworthy if it was carried through in a team spirit and the result would be a product of conjugation of computer science, medical and health management expertise. Unfortunately the materialization of the idea was hindered by the inadequate design for DHIS which did not have the prior aim of internationalization.

The DHIS described earlier is composed by a set of modules, each with specific functionality. Viewed from the perspective of a SBDA, the DHIS gives space for specific countries or organizations to

decide, according to their needs and priorities, what do adopt and localize first. For example in Mozambique it was agreed to focus first on the localization and translation of the monthly data module (FRONTEND), which still allowed us to use the software. Similarly we had the option to translate or not the documentation of DHIS. However, given the relatively advanced nature of the DHIS implementation process in South Africa, we found the user manual provided with the software was developed for users with high computer skills, which was definitively not the case in Mozambique, especially in the rural areas.

In contrast, localizing a GBDA implies taking care of primarily all the engineering aspects before the software is released for distribution. For example, in this case the functionality present can be the base to categorize the users as beginners or advanced, according to their confidence in using the software. However, a primary focus on the technical aspects implies that the context specific meanings of use are ignored. As a result, the users can feel alienated and resist the system.

In the SBDA, the source (developers in South Africa) and the target (localization team in Mozambique) teams are expected to co-develop the software, and the source counterpart acts as a supervisor, ensuring all the desired features at the required standard are included (Uren, 1993). However, such a development model assumes the existence of local capacity, and not only about the technical features of the software, but also of domain (medical) and context of use (public health). This assumption, while initially incorrect, however has positive implications in the long run. As through the practical experience of the translation, local capacity gets enhanced, the potential for longer term sustainability of the system is incremental (Korpela, 1998).

Our analysis helps to identify the points of differences in translating software for GBDA and SBDA systems.

	GBDA	SBDA
What should be translated	• Translate interface • Keep functionality	• Translate interface, documentation • Localize/translate functionality
Competence needed	• Computer skills • Technical expertise relating to formats, standards • Professional language and software translation	• Computer skills • Application domain experience & knowledge • Contextual language & software translation
How	• Long-term iterative process of translation according to INTERNATIONAL market requirements • Separate in time and space interface translation from functionality	• Short-term iterative process of translation according to LOCAL requirements • Not separate in time and space interface translation from functionality
Localization	• Technically enable the software to support foreign languages and basic formatting required • Vendor or corporate-driven • Provide tools and utilities for local customization of interface	• Technically enable the software to support foreign languages, the required formatting and the meanings, e.g. symbols, colours • Require more time and effort for understanding the

	• Through localizers	culture, meanings • In-house, contextual process of development • Multidisciplinary team representing different knowledge domain • Full involvement of potential users • Provide tools and utilities for local customization of interface and functionality
Internationalization	• Semantic factors most important • The commercial restriction does not allow for providing the source code • Through internationalizers	• Context factors most important • Internationalization as first step • Provide the application and source code
Globalization	• Applicable • Same standard across countries and cultures • Through globalizers	• Not applicable

Table 2. Summary of semantic and context matters

Barbour (1996) argues that separating the functionality from the interface will help accelerate the internationalization of software. However, as this case emphasizes, translating the DHIS and creating the backend are not two separable exercises, and are linked to the functionality of the software. As a result, they need to be performed simultaneously, involving people knowledgeable about the application domain, public health and of the context in which the system is intended to be used. Translation is not merely a technical exercise (Barbour *at el*., 1996), but involves mastering the domain of system use, the content of the application, new (technical) terms, use of shortcuts in menus, names of objects (trash can, dustbin, period – US, full-stop - UK), leaving space for text expansion, and maintaining consistency in terminology and documentation.

In developing application specific software to be used across countries it is not enough to cover only technical issues (Barbour *at el*., 1996), but requires strategies to separating the user interface from the functionality of the application software. This provides space for individual user interface development and translation using the experience of local teams. In providing these insights based on the practical experience of translation, it has been attempted to answer the challenge posed by Rolland and Monteiro (2002) on how to find the "pragmatic balance".

7 CONCLUDING REMARKS AND RECOMMENDATIONS

In this study, the primary concern was about understanding the processes involved in translating HIS in the context of developing countries. Specifically, the focus of the paper was to analyse the practical challenges of translating a HIS designed and developed in South Africa to be subsequently used in Mozambique.

Two application domain perspectives were distinguished, GBDA and SBDA, and the differences in their internationalization and localization processes were identified.

The evident tensions between the needs for internationalization and localization models were highlighted and five specific challenges were identified. These challenges need to be considered for purposes of both the language translation and adapting the HIS in varying contexts of use.

However, there are certain elements in HIS which are indeed common, and can be taken from one context to another as starting point, so as to avoid re-inventing the wheel. So, while there is no need to fully "reinvent the wheel" sensitiveness to contextual differences must be taken into account when designing, developing or implementing systems. As the HISP initiative is underway in different developing countries, lessons from this translation experience can be useful to guide localization exercises in other contexts, for example to Swahili in Tanzania and to Telugu in Andhra Pradesh, India.

8 ACKNOWLEDGEMENTS

The author wishes to thank Professor Sundeep Sahay and Professor Jens Kaasbøll who guided the preparation of this article. Sincere thanks are due to Jyotsna Sahay, the HISP team and the officials in the Ministry of Health, Mozambique.

References

Barbour, R., H., Yeo, A., 1996. Bilingual/Multi-Lingual Business Software: The Spreadsheet. Proceedings of Information Systems Conference of New Zealand. IEEE Computer Society Press, 63-71, Los Alamitos, California.

Braa, J. and Hedberg, C., 2002. The struggle for district-based health information systems in South Africa. The information society, 18 (2), 113-128.

Braa, J., Macome, E., Da Costa, J. L., Mavimbe, J. C., Nhampossa, J. L., José, B., Manave, A., and Sitói, A., 2001. A study of the actual and potential usage of information and communication technology at district and provincial levels in Mozambique with a focus on the health sector. Electronic journal of information systems in developing countries, 5(2).

Braa, J., Monteiro, E. and Sahay, S., 2003. Networks of action: sustainable health information systems across developing countries, (accepted for publication in MIS Quarterly, special issue on Action research, to appear 2004).

Gregory, J., Mattison, J. L., 1995. Naming notes: Transition from free text to structured entry. Methods of Information in Medicine. 34, 1/2. 57-67.

Griffiths, D., Heppell, S., Millwood, R., and Mladenova, G., 1994. translating software: What it means and what it costs for small cultures and large cultures. Computers Educ. 22, 1/2, 9-17.

Kaasbøll, J.J. and Nhampossa, J. L., 2002. Transfer of public sector information systems between developing countries: south-south cooperation. Proceedings of IFIP 9.4. "Social Implications of Computers in Developing Countries", 507-517. Bangalore, India.

Kalleberg, R., 1995. Action research as science and profession in the discipline of sociology, Draft.

Korpela, M., Soriyan, H. A., Olufokunbi, K. C., and Mursu, A., 1998. Blueprint for an African Systems Development Methodology: An Action Research Project in the Health Sector. In: C. Avgerou (ed.), Implementation and evaluation of information systems in Developing countries, 173-285. International Federation for Information Processing: Vienna.

Miller, C. L., 1994. Transborders Tips and Traps. BYTE. 19 (6). 93-102.

O´Donnell, S. M. (1994) Programming for the World, PTR Prentice Hall, Englewood Cliffs, New Jersey 07632.

Rolland, K. H., and Monteiro, E. 2002. "Balancing the local and the global in infrastructural information systems," The Information Society. 18(2), 87-100.

Russo, P., Boor, S., 1993. How fluent is your interface? Designing for international users. INTERCHI¨93 Conference on human factors in computing systems: INTERACI¨93 and CHI'93. (Amsterdam, 24-29,April). ACM Press. 342-347.

Susman, G., I., and Evered, R., D., 1978. An assessment of the scientific merits of action research, Administrative Science Quarterly 23, 582-603.

Taylor, D., 1992. Global software. Springer-Verlag, New York.

Uren, E., Howard, R., Perinotti, T., 1993. Software internationalization and localization. Van Nostrand Reinhold, New York.