Localization of Learning Objects in Mathematics

Prof. dr. Valentina Dagiene,
Head of Informatics Methodology Department, Institute of Mathematics and Informatics
Vilnius, Lithuania dagiene@ktl.mii.lt
Inga Zilinskiene
PhD student. Institute of Mathematics and Informatics, Vilnius, Lithuania
ingasolyte@ktl.mii.lt

Abstract Mathematics learning seems to be a demanding and time-consuming task for many learners. Information and communication technology (ICT) is an attractive tool of learning for students at any level and it can provide an effective atmosphere for understanding mathematics. The question is how to combine mathematics teaching contents, approaches, curricula, and syllabus with new media. The key issue in European educational policy (and other countries as well) is exchange and sharing digital learning resources (learning objects) among countries. In order to accumulate the practice of various countries and use the best digital resources created by different countries, it is necessary to localize learning objects (LO). The paper deals with some problems connected with localization of LO, developed for mathematics education, and presents some solution. Software localization is mainly referred to as language translation (e.g., translation of user interface texts and help documents). However, there are many other important elements depending on the country and people who will use the localized software. In this paper, the main attention is paid to localization of learning objects used for teaching and learning mathematics.

Keywords: teaching mathematics, learning objects, mathematics software, mathematical notation, localization

Introduction A constant development of ICT has unleashed new challenges in education. Its use is a strong element for knowledge construction support. In the Lithuanian community of mathematics educators, teaching mathematics takes a strong position in the education policy and, particularly, in the school community. Many students and parents consider mathematics knowledge as the key success for future life. However, the understanding of teaching mathematics is mostly based on an academic approach that is good enough mostly for motivated students. The majority of students especially teenagers are not interested in academic knowledge and are not able to develop mathematical literacy skills at all.

How could we make mathematics studies easier for both students and teachers? There are lots of suggestions that fall between deep "rethinking of mathematics" by S. Papert (1980) and gaming (Kahn, 2006). Using learning objects (LO's) for teaching mathematics can be one of the best ways. LO can be described as any digital resource that can be reused to support learning. To develop good LO's for teaching mathematics is great and time consuming work. So it is necessary to exchange LO's among countries. Therefore the European digital learning content implementation is based on the exchange of learning resources (European Schoolnet, 2006).

Therefore it is necessary to run the localization and adaptation of high quality content – digital learning resources. In addition, while developing new learning content it is necessary to take into account its future adaptation to other locales and other countries' educational systems i.e. to pay attention to internationalization of LO.

Overview of the tasks of localization

Countries and different speaking peoples use ever more and more diverse software. One of the main problems in software adaptation to local users is localization. Localization can not be interpreted as an action of translation (Grigas, 2000). Despite the fact that localization of software is estimated by translated resource lines, translation makes up only a small part of software localization (Esselink, 1998).

Discussions on software localization usually point out three main parts of localization (Fig 1): 1) software adaptation to target locale, 2) translation and adaptation of user interface, 3) translation and adaptation of software documentation.

Software adaptation to particular locale norms serves as the basis of a localization process. According to the international standard ISO/IEC 15897 (ISO/IEC 1999), locale is "the definition of the subset of a user's information technology environment that depends on the language, territory, or other cultural customs". Usually, three main components are attributed to locale: 1) language (which can be understood by the user and which must be handled by the software), 2) culture (non-verbal aspects of the product's functionality (Hall, 1990, Schäler, 2002)), 3) local practices and conventions (aspects such as legal requirements, notation, measurement units, etc.) (Hall, 1997).

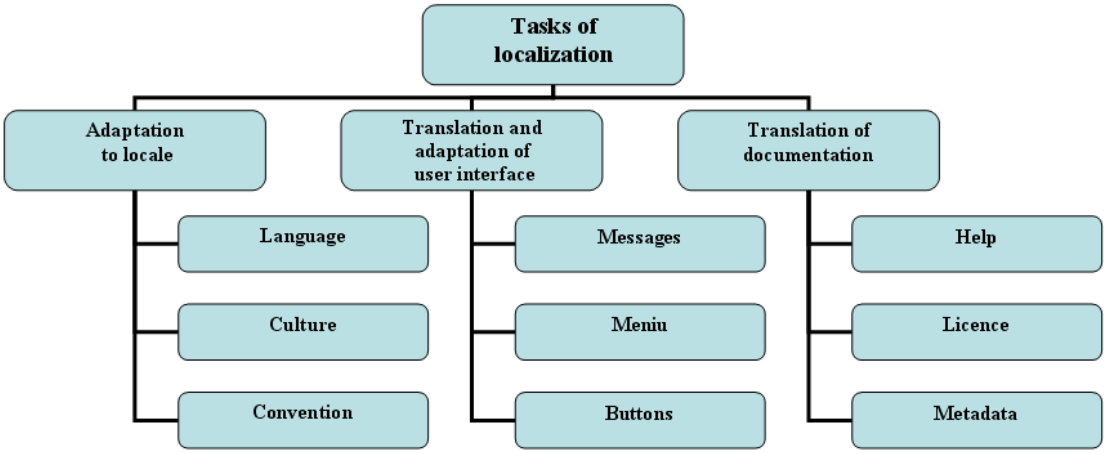


Fig. 1. Main components of software localization

Locale is usually identified by the language, using a two-letter language code (ISO 639-1), and by territory, using a two-letter territory code (ISO 3166-1). Locale depends not only on the language (for instance, locales of Great Britain and the USA are different, although these countries use the same language) or only on the country (for example, in Canada there are two official languages, English and French, each of these combinations of language and country usually have their own ways of expressing dates, times, numbers and other elements).

POSIX (Portable Operating System Interface for Computer Environments) standard was one of the first to define basic locale data. POSIX locale model (ISO/IEC 9945-2) has six main categories, that define (Jevsikova, 2006): 1) Character classification and case conversion. 2) Collation order. 3) Monetary formatting. 4) Numeric, non-monetary formatting. 5) Date and time formats. 6) Formats of informative and diagnostic messages and interactive responses.

This is a minimum set of locale elements for any software including LO as well. However that is not enough for high quality localization. Some new elements were added in later locale models.

Adaptation of user interface texts is the second component of localization. User interface texts are records and messages (text strings) within software's dialogs and their elements (buttons, captions, boxes, menu bars, etc.).

Translation of help and documentation files (printed or online) is the last component. It is important to point out that translation of user interface texts and help files usually is the most time and effort consuming task because of a large amount of such texts and a requirement to preserve the consistency with the adapted user interface.

The first step of localization is to adjust the software to the norms of locale. Only after that the adaptation of resources and the translation of help files may be performed.

Localization of mathematical notation

The software used in comprehensive schools of many countries is: operational systems, programs of file managing, text processors, spreadsheets, database programs, internet browsers, e-mail programs, antivirus programs, and presentation programs. They are almost exempt from

mathematical elements (except the use of the usual mathematical symbols in the keyboard, other minimal tools for writing equations, fractions, etc.)

Localization problem of mathematical programs have been noticed since the interest in learning objects (LO‘s).

When localizing the parts of mathematical software, as usual, the main attention was paid to mathematical characters. They are very different in different countries. The notation of even the main arithmetic operations (multiplication and division) is distinct in various countries. (Table 1).

Different countries use distinct measuring units and their notation. For example, the notation of length, weight, temperature, currency, etc. is quite different in Great Britain. The decimal separator is also a point, while in Lithuania it is a comma. It is easy to change that in one program, however, if the program uses another one that can be not localized, then we have to envisage the consequences of a possible conflict. An analogous situation is with other mathematical notation.

It is not so complicated to replace (localize) separate characters, however, it is much more complicated to localize if the order of writing or notation is changed (e.g., in Japan postfix notation is used to designate functions).

Localization of mathematical teaching strategies

When localizing computer programs of teaching (learning) mathematics especially, we have found some methodological differences between Lithuanian mathematics didactics ant that of foreign countries, e.g., while performing some arithmetic operations: not only the notation, but also the ways of calculation are different (LO “Rechenheft“, http://www.rechenheft.com, “Rainforest“, http://www.rainforestmaths.com, etc.). Thus,

Table 1. Mathematical notation in different countries

Mathematical operation		**Mathematical notation**					
	Multiplica-tion	**Austria** 1014·163 6084 3042 165282					
	Division	**Austria** 9163:14=654 076 063 7R	**Denmark** 5/375\75 35 25	**Israel** 75 375| 5 35 25 25 00	**Japan** 75 5) 376 35 26 25 1 Answer: 75…1	**Croatia** 376:5=75,2 35 _ 26 25 10	**Australia** 174 r1 5) $8^{3}7^{2}1$

localization of software, especially that of LO‘s embraces not only technical, but also methodological problems, after exposing of which there is a possibility to present a qualitative product to the society.

Some LO‘s have been essentially changed, reprogrammed in the process of localization. An example of this kind is “Rechenheft“(created by Christian Nosko, a teacher from Austria). At present the program is operating in the Lithuanian language, though it was rather difficult to localize it (despite that there is not so much of the text). It has turned out that in Austrian schools, arithmetic operations are performed quite in other methods than in Lithuania. For instance, multiplication is performed not from left to right, as is the habit with us, but from right to left.

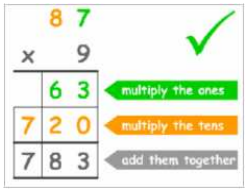
Division is also performed in an extremely complicated way. Intermediate operations of multiplication and subtraction are omitted by writing below the difference obtained only: all the intermediate operations are performed mentally.

a. 

b. 

Fig. 2. Methods of multiplication and division in Austria

One more aspect for teaching multiplication is displayed in the program developed by Jenny Eather in Australia „Rainforest Math“ (Fig. 3) Multiplication is performed in the way usual to us, however, while multiplying units by tens (hundreds, thousands, etc.), the exact number is written that is obtained multiplying units by tens, i.e., in this example number 720 but not 72, but transferred one position to the left. The multiplication operation is performed in Australia just in the same way as in our country; only it is written in another form.

a. 

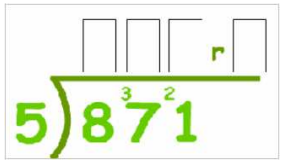
b. 

Fig. 3 Methods of multiplication (a) and division (b) in Australia

Software Internationalization

The software designed for international market should be internationalized, i. e. the provision for its adaptability in any locale should be ensured. It’s just several years ago the wider attention to the issues of program internationalization was paid. (Reinecke, 2007) That’s why these issues are still poorly analyzed from theoretical point of view and still there are no solid standards or specifications for the internationalization of software. The contemporary tools for the software development are also still not adopted in the production of the proper internationalized programs. Namely because of this the adaptation of the software localization very often turns to be quite complicated.

Localization was initially approached as an "add-on"; i.e., after the original program was fully functional in English, localizers had to work on a Spanish, French, Japanese, etc. versions. However, programmers soon had to realize that such ex post facto solutions were inadequate; in many cases they required the re-writing of source code, which was a costly step that could have been avoided, if only the future internationalization would have been a part of the initial programming plan. (Uren, 1993) Many other authors, for example, Tuoc, David, and Driscoll (Tuoc, 1995), argue that internationalization must be part of the earliest design stages of any program, which must be written so that localization would be possible without rewriting the program's source code.

Internationalization involves isolation of the culturally and linguistically-dependent parts of software. Software internationalization is a framework for software localization; it is the process of designing and developing products with sets of features, functions and options to facilitate the adaptation of the product to various international markets (Hall, 1997).

Conclusions

Software localization is the process of adapting a software product to the linguistic, cultural and technical requirements of a target country or language. This process often requires a significant amount of time from the development teams.

Localization process can be divided into three main components: 1) software adaptation to target locale, 2) translation and adaptation of user interface, 3) translation and adaptation of help and other documentation. All components are related to each other.

Learning objects are the core concept in an approach to learning content in which content is broken down into "bite size" pieces. These pieces can be reused, independently created and maintained, also pulled apart and stuck together like lego bricks. That means that a learning object could be a piece of software as well as a text document, a movie, a presentation, an mp3, a picture or even a website.

When localizing mathematical learning objects all three above mentioned components should be considered: adaptation to the locale, translation and adaptation of user interface as well as translation of documentation. Additionally, attention should be paid to the mathematical notation which might be distinct in various countries. But it is not enough.

Analysis of many localized mathematical learning objects has shown that different countries have been using different teaching approaches and strategies. So these teaching approaches and strategies should be recognized and localized as well. Sometimes it is very hard work and requires substantial reconstruction of the learning object. Consequently, before starting localization of a mathematical learning object, the LO should be thoroughly analyzed not only in technical and notational aspects but also in point of the educational approaches and strategies. Localizers of mathematical learning objects have to be acquainted with various teaching strategies and be able to adopt them.

References

1. Esselink, B. (2000) A practical guide to localization. PA: John Benjamins Pub Co.
2. European Schoolnet. (2006) Insight observatory for new technologies and education. Learning Resourche Exchange. http://insight.eun.org/ww/en/pub/insight/interoperability/learning_resource_exchange.htm
3. Grigas, G., Zalatorius, J. (2000) Cultural and economic aspects of software localization. Baltic IT Review, nr. 6, 61–64.
4. Hall, E., Hall, M. (1990) Understanding Culture Differences. Maine, Intercultural Press.
5. Hall, V., Hudson, R. (1997) Software without frontiers. John Willey & Sons, Inc.
6. ISO/IEC 15897. (1999) Information technology – Procedures for registration of cultural elements.
7. Jevsikova, T. (2006) Internationalization and Localization of Web-based Learning Environment. In: R. Mittermeir (Ed.) Informatics Education – the Bridge Between Using and Understanding Computers. LNCS, 4226, 310-319.
8. Kahn, K., Noss, R., Hoyles, C., Jones, D. (2006) Designing for Diversity through We-based Layered Learning: a Prototype Space Travel Games Construction Kit. In Proceedings of the 17th ICMI (International Commission on Mathematical Instruction) Study Conference “Technology Revised”, C. Hoyles, J.-B. Lagrange, L. Hung Son and N. Sinclair (Eds.), December 3–8, Hanoi, Vietnam, 260–267.
9. Papert, S. (1980) Mindstorms: Children, Computers, and Powerful Ideas, BasicBooks, New York.
10. Reinecke, K., Bernstein, A. (2007) Culturally Adaptive Software: Moving Beyond Internationalization. In: Aykin, N. (Ed.) Usability and Internationalization, pt II, proc. Global and Local User Interfaces. LNCS, 4560, 201-210.
11. Schäler, R. (2002) The Cultural Dimensions in Software localization. Localization Focus, vol. 1, issue 2.
12. Tuoc, V., David, J.T. Driscoll, K. (1995) Internationalization: Developing Software for Global Markets. John Wiley & Sons, Inc..
13. Uren, E., Howard, R., Perinotti, T. (1993) Software internationalization and localization. An Introduction. Van Nostrand Reinhold.