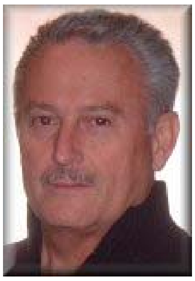
Volume 9, No. 2
April 2005



Kenneth (Sandy) McKethan, Jr. is a globalization project manager for IBM's Tivoli division. He is a foreign language professional with over 30 years' experience in various facets of the language industry: as a translator and interpreter; in recruiting, sales, marketing, and international business development; vendor management; as well as project management. He is a certified Project Management Professional (PMP) and a German-government certified technical interpreter and translator (Staatlich geprüfter Dolmetscher und Übersetzer für die englische Sprache [Technik]). He holds the rank of Lt Col in the U.S. Air Force auxiliary (Civil Air Patrol).:

He can be reached at smcketha@us.ibm.com.

Born and raised in Buenos Aires, Argentina, Graciela earned her BA in Legal Translation from Universidad Nacional de Buenos Aires - Law School (1980). She worked as a senior technical translator/vendor coordinator for IBM Argentina and traveled to several IBM development sites in Europe and the US to perform software verification testing. In 1988, she moved to the United States where she worked as a freelance translator in the fields of: law, software localization, telecommunications, and electrical engineering. She is currently a Globalization Project Manager with IBM where she oversees the fulfillment of IBM's goals of enablement and localization in all its software offerings. In the last few years, Graciela has also been a workshop leader/presenter of software localization related topics before local and international audiences.

Graciela can be reached at whiteg@us.ibm.com

Front Page

Select one of the previous 31 issues.
Select an issue:

Index 1997-2005

Translation Journal

Localization

Demystifying Software Globalization

by Kenneth A. (Sandy) McKethan, Jr. and Graciela White

Representing a growing segment of the translation industry, software globalization (G11N) remains shrouded in mystery to many. What is Globalization? G11N ensures availability of a software product in languages besides the language of origin, traditionally US English. It is driven by huge revenue opportunities outside the Anglophone world for software companies and translators alike. This presentation will introduce both the basic concept of globalization and how it involves the translator, in particular. In this paper we will describe the process from early design and coding to release in the global marketplace.

1. WHAT IS GLOBALIZATION?

In just the last few years, the term *Globalization* has become a household word around the world. Just what **is** Globalization? Most often, the term is associated with the globalization that periodically appears in the news stories. This is *economic* globalization. Economically and in other ways, the world is shrinking, becoming more interdependent. We are witnessing the emergence of the Internet, global communications, global travel, global village, global—almost stateless— corporations, intertwining of the world's economies, blurring of national distinctions, challenging existing legal structures, global e-business, etc. Whatever one's views on the subject of economic globalization, several facts are clear:

- It is now a fact of life.
- It is here to stay.
- It is changing forever the world, as we know it.
- It is ever more evident.
- It is changing the way we do business at an increasingly faster pace.

In fact, 94% of CEO's of major US corporations have agreed, "Globalization is THE most important trend..."

Bottom line: Economic globalization is basically about the bottom line—get used to it!

But globalization also assumes worldwide communication. Worldwide communication assumes language.

Despite the proliferation of English through commerce and US pop culture, not all users of software speak English. Not all speakers of English as a second language throughout the world are able to use the language efficiently in their work. Some know just enough English to be dangerous! Nor would all even prefer having to use English to accomplish their daily tasks. This is particularly true at the end user level. In fact, somewhat of a backlash to English has been noted, even in some traditionally English-friendly countries such as Germany and Israel. In other words, despite the advance of economic globalization in all its manifestations, national language identity remains alive and well, often for very practical reasons. As one German customer recently told us: the user should be able to use the product seamlessly in German, and not have to go through a translation exercise in order to utilize the output of a particular application. In a real-world business environment, all users need to understand application output, accurately and in real time, and not just those who happen to be initiates into the English language.

We are witnessing trends within the software user community that underscore these conclusions. Given the increasing level of automation within the industry, emerging hands-on users are increasingly unlikely to have sufficiently strong English skills to justify an English-only deployment in some environments.

The following statistics provide context. Global Reach, an international marketing and Web site promotion service, broke down the world's native speaker online population by language zone as follows:

Language	2002 Statistics	2004 Statistics/Population
English	37.3%	35.2% (508 M)
Chinese	11.1%	13.7% (874 M)
Japanese	9.9%	9.4% (125 M)
Spanish	7.6%	9.0% (350 M)
German	6.8%	6.9% (100 M)

For further details, cf. the Global Reach site at http://global-reach.biz/globstats/index.php3.

Clearly, "English only" is no longer the right answer!

Among other things, all this means that globalization is more than just a buzzword. The focus of Internet marketing, e-business, and e-commerce is shifting from the US market to a truly global economy. It follows then, that such a marketplace also demands that software be tailored or "globalized" to support key non-English markets.

So just what is *software* globalization? The short answer is making software products run anywhere. In our context at IBM, it means making applications work seamlessly, regardless of the user's language and culture. Globalization is all about choices: it gives the user the choice to use the original US English version of the product, or in fact, any of the other supported languages.

Software globalization must start well before the translation phase. It begins with the very design (as opposed to retrofit) of software to allow its use in non-US locales. Known in this connection as Internationalization (I18N), product developers must deliver designs that allow for such features as selectable date and currency formats, as well as dynamic resizing of buttons and boxes. Users must be able to input, view, and print data using their own character sets. It must also address different character sets such as ideographic (also known as Double-Byte Character Set or DBCS) and simple text (Single-Byte Character Set or SBCS). Complex text languages (BiDirectional or BiDi) such as Arabic and Hebrew must likewise be seamlessly supported. Observance of local government requirements must be factored in, such as the Chinese government GB1830 certification character set standard. Another example is the Québec Law 101 that makes French the language of education, work, and business in Québec. Absent such compliance, the product cannot be sold in the country. It is as simple as that. Unicode™ support is also a must at IBM. In fact, IBM applies an entire set of globalization-related product design or NLS (National Language Support) enablement requirements. Any deviations are subject to approval. These internationalized capabilities are built into the initial design for a simple reason: it is far less expensive and disruptive to do so from the outset than to go back later and retrofit. Some companies have had to learn this the hard way. Internationalization then, is the process of producing a product (design and code) that is totally free of any dependency on the language, script, culture, and coded character set.

Strictly speaking however, an internationalized product is not usable in any region of the world unless it is localized to that specific region. It must also speak the local language in every sense of the word. With a solid foundation of internationalization in place, it is relatively straightforward to then "localize" into the desired language. Localization (L10N) is the process of adapting an internationalized product to a specific language, script, cultural, and coded character set environment. In localization, the same semantics are preserved while the syntax may be changed. Localization goes beyond mere translation. The user must be able to not only select the desired language, but other local conventions as well. For instance, one can select German as a language, but also Switzerland as the specific *locale* of German. Locale allows for national or locale-specific variations on the usage of format, currency, spellchecker, punctuation, etc., all within the single German language area.

This simple equation is used to put these processes into context:
G11N = I18N + L10N

Today's global economy is unthinkable without software, specifically software that supports multiple languages and locales simultaneously.

2. WHY DO WE GLOBALIZE OUR SOFTWARE?

As part of the development process, IBM early recognized the importance of its presence in foreign markets. Software that is available in the user's own language and runs on a localized operating system provides a great convenience to the user and represents a huge competitive advantage. Given the predominance of languages other than English, it just makes good business sense to address those markets.

As a pioneer in the field of both economic and software globalization, IBM has always emphasized products that can be used worldwide. The reason: the majority of IBM revenue is generated outside English-speaking countries. Multinational companies want uniform solutions that can be used worldwide. In some countries, laws leave no choice to those who want to do business there.

The IBM vision sums up our approach to software globalization: "A user can access a server from anywhere in the world, using a client in

- Web Surfing for Fun and Profit
  by Cathy Flick, Ph.D.
- Translators’ On-Line Resources
  by Gabe Bokor
- Translators’ Best Websites
  by Gabe Bokor
- **Translators’ Events**
- **Call for Papers and Editorial Policies**

the language of his or her choice, work with applications and interact with other users in the language and cultural conventions of choice." This direction is implemented in the strategic objective to deliver comprehensive, consistent solutions to meet worldwide customer requirements.

In turn, this vision drives a set of baseline requirements aimed at delivering consistent end-to-end solutions in a set of languages with cultural support. These include globalization architecture requirements and worldwide availability of all languages, generally simultaneous to the US English product release.

3. Implementation

Current implementation of globalization covers several aspects. First, we will discuss specific elements such as processes that we have in place and the tools used to execute them. Then we will address the people who really make globalization happen, the project teams.

3.1 Tools and Processes

IBM translates millions of lines of product information every year at more than 30 translation centers around the world. This infrastructure has evolved over a 25-year period to meet a growing demand for good quality localized software. As a true pioneer in the field of software localization, IBM was quick to recognize the need for this kind of translation, altogether different from traditional documentation translation, and to put into place the needed processes and guidelines.

IBM's process includes a central clearinghouse for incoming source files. Before being released to translation worldwide, all material for translation must undergo file quality checking. Translation memory project folders are created and an English-to-English word count is performed to derive the delta.

Two key actions by developers have direct bearing on translation cost: communicate early and often with the translation centers, and select appropriate authoring tools. All files sent to our translation centers must be verified against a set of internal tools to ensure that they can be translated without errors or unnecessary expense.

We take pride in our planning and interaction with other teams to make sure that the translation centers are not surprised by source files arriving late, unannounced substantial volume increases, file errors, and unscheduled shipments after user interface or documentation freeze. A reporting system is also in place to make sure that we can quickly and efficiently communicate with all software localizers regarding any questions on the source material. Answers to those questions are broadcast to the originators as well as to all the other translators involved in the project. This prevents those same questions from being posed again.

Early in the planning cycle it is important to identify translation assets already in place. These include determining the availability of any English source (from earlier versions of this product or other similar products) that can be reused, and the availability of applicable translation memory. Another consideration is whether source conversion will be necessary (for multiple outputs). Finally, this is also the time to identify any new terms that will be used in the product. All too often, terminology is left until very late in the development cycle, causing increased costs or delays. We have in place a terminology process that describes the tools, process, and responsibilities for handling terminology and ensuring the availability of approved glossaries to all translators.

One of the vital elements in this localization process is the use of TranslationManager, a proprietary translation memory management tool. Like other tools (Trados® Freelance™, Atril Déjà Vu, STAR

Transit, SDLX™, only to name a few) quite popular among the members of the software localization community these days, TranslationManager has been developed by IBM for use in translating its UI (User Interface) and publications.

IBM TranslationManager is a Computer-Assisted Translation (CAT) system that automates repetitive tasks, freeing the professional translator to attend to the finer points of translation that require the judgment of an expert. TranslationManager helps accelerate turnaround time and save money in overall translation costs. It enables the translator to handle large volumes efficiently while eliminating much of the tedious routine work inherent in translation projects. TranslationManager supports single-byte languages, double-byte languages, and those languages with bidirectional scripts, such as Arabic and Hebrew. In bidirectional languages, the general flow of text proceeds horizontally from right to left, but numbers, English, and other left-to-right language text are written from left to right.

As part of the localization process, we perform a series of tests to verify the quality of the translated version. Globalization Verification Test (GVT) provides early verification of translatability and globalization enablement function during product functional testing, well ahead of the next step, translation verification testing. In a nutshell, GVT ensures that the source product works properly on a localized operating system. Later in the cycle, mock versions of the source product are used to check for truncations, corrupted characters, and space limitations.

As a final step, Translation Verification Test (TVT) provides the real test of localized version quality. This stage requires close cooperation by development team and translation testers. Ideally, this testing is performed in-country using the translated document. This is because it is difficult to set up a specific country's test environment elsewhere in the world. If the test is to take place in the country (a recommended Best-of-Breed practice), the tester must communicate the specific hardware and software needs early and schedule time to install and test equipment. On the other hand, if the test is to take place in the development lab, the lab developers must ensure that the appropriate test equipment is installed. Developers must also make sure that testers have access to the tools required to change the information source where needed, for instance in the case that SGML source is being translated and transformed into several output formats and an error is discovered in one environment. In this situation, the fix needs to be made to the SGML and not to the output format. Translation testers arrive equipped with TranslationManager installed on a laptop computer so that any changes they need to make can be directly fed back into the translation memory. They also come armed with test cases. These are prepared in advance to permit complete focus on testing activities.

3.2 Teams

The globalization equation also has a human component, the people who actually make globalization happen. These include the engineers who work on internationalization, software localizers, testers, and project managers. Given the widely distributed work environment of software globalization, these resources by necessity comprise what are known as virtual teams.

In the past, people had to be physically located in the same place to be considered as working together. In today's world, individuals no longer need to be co-located in order to work together[1]. By virtual teams we mean "groups of employees spread across countries and companies that work together with little face-to-face interaction"[2]. What was once considered a far-fetched concept has exceeded all expectations by its original proponents, all within the space of the last decade. The concept of team embodies much more than just sharing ideas to kick off a project. It is predicated on reciprocity and interdependence. A member is unable to complete a piece of the project without deliverables from other members, each piece of the project being a two-way street. In other words, this sense of interdependency improves and enriches the work experience.

How did virtual teams come about? Back when traveling was cheap and easy, people readily defaulted to it. Lately, new factors include increasing pressures to lower costs and increase quality, combined with rising expectations in terms of work/family balance. This changed work environment has provided motivation to participate in this new dynamic of collaborative work with people located worlds away. For now, this phenomenon is mostly seen at big corporations. It is safe to say that smaller companies, concentrated in a single region and having a local customer focus as yet have less need to play by these rules[2]. Another major factor contributing to the rise of virtual teams has been reduced communication costs. Before the dominance of the Internet, it was relatively expensive to place a phone call overseas. Lowered communication costs have gone a long way in making it possible to establish this voice-to-voice interaction so important to the success of a virtual team.

Another key characteristic of virtual teams is the flexibility of its members. The demands of this unorthodox approach to project execution can elicit some creative solutions to problems. Such solutions often include taking a well-deserved nap during the day in order to be able to work through the night to meet a deliverable for a coworker located halfway around the world, or to take a late-night phone call to brainstorm with the members of a team stationed in another time zone. Likewise, those who participate in virtual projects tend to be self-motivated individuals who possess good communication skills. As stated earlier, virtual teams make frequent use of audio conferences, a setting that requires that expression be concise and articulate. Since participants often represent a variety of native languages, this can be challenging.

One aspect of virtual teams merits particular attention: sensitivity to the cultural background of participants. Every culture has both widely known and unverbalized codes and customs that should be taken very seriously, even if they seems really "foreign" to other team members. In these circumstances, it is entirely appropriate for the project manager to suggest a cultural immersion to help the members get acquainted with this behavior. This could include such simple concepts as the proper form of address for one from another culture. For instance, a very casual style might be offensive to someone else. Other considerations include schedules that require other members to work during their national holidays or festivals, or planning meetings that impact religious celebrations. The lack of visual contact sometimes leads to misunderstandings that would not occur in face-to-face communication. All of this calls for an extra measure of tolerance on the part of all.

Virtual teams are also built on trust and common expectations. Without the usual organizational walls to serve as general parameters, virtual teams need guidelines. These enable the team to set personal as well as team expectations for what they are and are not allowed to do.[1] Managing expectations is necessary in a creative environment like a virtual team to head off frustration with the system and other team members. There should be strong commitment from each member to the fulfillment of common goals. Any deviations should be immediately communicated and discussed in order to avoid unwanted results.

Basically, virtual teams have come into their own as a very efficient way of getting things done across borders. This is particularly the case in the field of software localization.

4. CONCLUSION

Economic and software globalization are very much intertwined and interdependent. The latter is both an outgrowth and a major contributing factor to the former. Today's massive volume of e-commerce would be unthinkable without the support of globalized software applications. Users must be able to operate software and expeditiously utilize its output. Carefully engineered software that can operate seamlessly in a variety of locales is now standard. It is this reality that drives the business of software globalization. Software companies internationalize and localize their products simply because this makes good economic sense.

NOTES

[1] CSWT Papers, *Virtual Teams* by Cynthia Cantu. Copyright 1997, Center for the Study of Work Teams, University of North Texas.

[2] *Virtual Teams: How they work*. Interview of Arvind Malhotra, Assistant Professor of Information Technology at UNC-Chapel Hill's Kenan-Flagler School of Business, by Jonathan B. Cox, Staff Writer, published in The News & Observer (Raleigh, NC) on 04/23/03.