An Interview with Dirk Goepfert, Localization Engineering Manager

Readers of recent columns will have noticed that *OWOW* has been visiting different planets in our linguistic universe, boldly going where not many translators have gone before. The space travel metaphor is, in my opinion, apt here, since we are going to be talking about localization, which involves adapting ideas, products, and services to languages and cultures all over the known world. If we ever find digital life anywhere else, there will be great demand for the services provided by my guest today. Dirk Goepfert is the localization engineering manager at Siemens Product Lifecycle Management Software Inc. in Milford, Ohio (*www.siemens.com/plm*).

Thank you for talking to us, Dirk. Please tell us a little about yourself.

I was born and raised in Germany and obtained my MA in technical translation with a specialization in software localization from the University of Applied Sciences in Flensburg, Germany, in 2000. I started my professional career working as a freelance English>German translator for various international language services providers as well as some direct clients. In 2001, Interpro Translation Solutions, a small Chicago-based translation services provider offered me an in-house position as an English>German translator. After about six months I was promoted to project manager for small-scale localization projects (e.g., user interface, online help, training documents). In my role as a German localization specialist, I performed localization activities and managed teams of localization specialists working on Windows and IBM iSeries user interface localization projects.

In 2002, UGS Corp. (now Siemens PLM Software) needed a native German localization specialist and offered me a position at their location in Maryland Heights, Missouri. Later, I moved to their location in Milford, Ohio, where I was made the localization engineering manager responsible for all localization activities for German, Italian, Spanish, and French markets. My team (five localization specialists) is responsible for localizing user interface strings, online help, training documentation, and multimedia deliverables, and for testing the appropriate local language applications/deliverables.

Dirk Goepfert

When you graduated with an MA, how extensive was your knowledge of computer programming? If technical translation and computer programming were two languages, which would be your dominant one?

When it comes to software localization, having a good understanding of both those languages can be extremely beneficial. I do see an emphasis on technical translation, supported by the relevant programming or software framework information. My university's curriculum provided students with an introduction to programming, which was meant to create a foundation for future learning activities.

As a software localization student, what were the basics of your training?

My training entailed getting to know the basics of translation theory, translation methodology, terminology research, and terminology management. It also included an introduction to computer-aided translation tools. Students were taught the basics of technical infrastructure (operating systems) and certain programming languages.

How much programming must a translator know to work on a software localization project?

That really depends on a variety of factors, the scope and complexity of a solution requiring localization being one of them. When I started my professional career as a freelance translator, I was involved in a range of projects, some requiring hardly any programming skills (localizable content was pre-processed, so all that was required was to complete the text translation), and others requiring front-end pre-processing, code editing, scripting, compiling, and testing the localized deliverables. Scripting can be very beneficial for localization engineers, since certain tasks can be automated. When properly internationalized, software localization activities do not include any code changes.

There are many online descriptions and definitions of localization. What's yours?

Indeed, there are quite a few different definitions available. My focus is the global local-language end user, so, to me, localization stands for adapting a product or deliverable based on the requirements of a defined target market and locale.

Would you say that the localization process stands on the shoulders of terminology management?

Terminology management is a very important *support* process that enables successful localization activities. Terminology is what conveys corporate identity and helps differentiate one product from a competitor's. In my opinion, a well-defined and effective approach to terminology management must consider both the source *and* target locales, using high levels of automation and minimizing manual tasks. I would say that only a highly automated terminology management process can be scaled up to support very comprehensive solutions suites.

Your team is responsible for localizing user interface strings. Could you give us some examples and explain what they are doing?

The linguists on my team are called *localization engineers* because their duties go well beyond translation. They localize user interface strings; perform dialog resizing; translate the online help, courseware, presentations, marketing collateral, and videos; perform post-editing of machine translation output; evaluate third-party translations using our quality assessment model; and participate in research and process definition. Our localization engineers also test all our localized software and perform language engineering activities, such as local language synthetic text-to-speech customizations.

In a major localization project, what is the range of required skills? Who else is involved?

Strictly from a localization perspective, most projects will involve localization management (local language release planning), project managers, localization engineers, subject-matter experts, and additional global or in-country project stakeholders. All activities are workflow-based and each project stakeholder can be assigned to one or multiple workflow roles.

As manager, you are responsible for localization activities in German, Italian, Spanish, and French. Are you equally at home in all those languages?

As a localization engineering manager I do not have to be fluent in any of those languages, since we have in-house localization experts. In my role, I resolve issues, facilitate local language activities, keep an eye on current and upcoming projects and activities, schedule localization work, connect team members with project stakeholders, provide application and process support, and make sure my team members' training needs are addressed in a timely manner.

What sort of cultural issues does localization address?

If a product is internationalized properly, many potential cultural issues will already have been resolved. Localization engineers make locale-specific decisions on a daily basis—from deciding whether or not a formal way of communicating with an end user is applicable for a particular locale to considering different versions of a language (e.g., Spanish for European versus American markets).

> *When it comes to software localization, having a good understanding of both those languages can be extremely beneficial.*

Are there any truly global concepts that can be understood everywhere? Which come close?

I think ROI (return on investment) and TTM (time to market) may be concepts that are used and understood globally.

Translators are involved with terminology research, editing, proofreading, and, in some cases, formatting. Where do translation and localization overlap?

Localization engineers are constantly involved in terminology research, editing, and proofreading. All those activities form a substantial part of a successful localization strategy.

When I think of certain texts in my computer software, and imagine them expressed in Asian characters, for example, I begin to get a sense of the complexity of this process. Tell us something about the systems required to manage that complexity.

For an application to be localized, it must first be internationalized. As part of the internationalization activities, the English code base will support localization activities for double-byte languages (Simplified Chinese, Traditional Chinese, Japanese, Korean, etc.), different scripts (Latin, Cyrillic), and right-to-left languages. The internationalization activities are performed on the code base. Thanks to XML using Unicode encoding, all local language versions can be created and maintained using a workflow-driven unified and standardized localization infrastructure.

What are your thoughts on volunteer-driven open source projects? Are they useful? How?

Developing free and open-source software (FOSS) is a great way to address a very specific need in a very short timeframe using a global community of developers and contributors. During my freelance translator years, I contributed to quite a few FOSS projects. I think it's a great way to contribute one's specific expertise to a common goal, to be part of a global team and expand one's skill sets.

If someone wanted to volunteer as a translator for one of those projects, how would they do that?

There are plenty of FOSS projects out there, and localization and/or translation activities seem to be in high demand, for certain locales at any rate. Some projects may require a contributor to sign up directly to be able to work on a particular deliverable, but crowdsourcing projects are also available. I would suggest looking for potential projects of interest and checking with their owner(s) or coordinator(s).

There are so many separate pieces in a localization project. How do you keep everything current?

It all starts with the right planning and the proper production tools. We use our translation management system to keep track of all local language production activities using highly customized workflows. Given our Agile software development lifecycle model, certain localization activities have to be launched or re-launched at certain stages during the overall cycle. Our planning tools enable us to keep track of current activities and plan future ones.

How have online translation sites (Google Translate, etc.) affected the formal, professional world of localization? Is there any reciprocity?

I don't know whether the use of online translation sites has affected the professional world of localization, since our organization does not use such sites/services. We do use enterprise-level machine translation systems to support our localization activities for certain deliverable types. Both our internal localization engineers and our language services providers post-edit machine translation output before handing off any translations. This ensures that machine translations are always reviewed/revised by a human localization professional. The use of machine translation systems has enabled us to localize more content in shorter intervals, thus reducing overall time to market.

How do you keep a language's localization current?

Once a new release is in progress our translation management system compares the source assets against the previous version, and all new and changed assets are processed, analyzed, and launched for translation. If an existing local language deliverable version needs to be updated, we use our translation management system to launch a "maintenance project" that our localization engineers can use to implement any required changes or updates.

How does localization manage the rapid evolution of slang in most languages?

The English version of our software is designed to address English speakers worldwide. This means the English versions must be understood by end users in the U.S. and Canada, the U.K., India, the Philippines, Australia, and New Zealand, to name a few. To provide this "international" English version, software developers and content authors have to standardize simplified technical English as much as possible, minimize the use of variants and synonyms, and refrain from using country-specific slang. Authoring activities are governed by authoring style guides and verified/enforced by content editors. Using a standardized English source also supports higher translation memory matches and higher quality machine translation, which result in reduced localization cost.

Has the widespread use of smart phones affected localization in any significant way?

From an end user perspective it's great to have a choice, since (local language) content can be accessed in so many different ways, whether on a workstation, laptop, tablet, mobile device, or a combination of any or all of the above. This gives the end user a lot of flexibility. To support such a wide range of display devices, different content types and delivery systems had to be established. Cloud content delivery, in particular, provides a great deal of added flexibility. Using cloud-based content delivery systems, we are no longer constrained by scheduled releases or defined release vehicles, but can "push" content to the cloud whenever needed.

Have spell checkers and grammar checkers become standard everywhere?

I'm not sure they have become standard everywhere else, but both our internal localization engineers and our external translation services providers have been required to perform spelling and grammar checks for a long time.

Localization is referred to as "L10N." Would you explain, please?

L10N is a numeronym, a number-based word created as an abbreviation of a much longer word. In a numeronym, the letters between the first and last letter of the longer word are replaced with a number representing the number of letters omitted, such as I18N (internationalization) or globalization (G11N).

Is there a founding father of L10N or a country of origin?

I don't know whether a "founding father" of software localization can be determined, but I think the need for software internationalization and software localization surfaced once English software was adopted worldwide and U.S. software companies were faced with the challenge of providing non-English versions of their product. According to "The Evolution of Localization," an article by Bert Esselink (a solution architect at Lionbridge), Microsoft opened sales offices in Tokyo in 1978 and started its expansion into Europe in 1979.[1]

In a global project, translation and terminology management provide suitable equivalents in other languages, but some languages are so much wordier than others. What does localization do about that?

During the software internationalization phase, text expansion/contraction rates are designed and tested to make sure the software supports local language text expansion ("wordier" languages) as well as text contraction (selected double-byte languages). Our organization is responsible for verifying the display of local language strings within the localized user interface as part of our localization testing activities.

Where does localization go from here? What do you see as the future of your field?

Personally, I can see software being used in areas where it once played a minimal role, or none at all. The Internet of Things (IoT) will spawn generations of devices and systems that communicate with each other.[2] I imagine my future smart home will be able to monitor the inventory in my refrigerator, notify me regarding expiration dates, and order a gallon of milk when needed, which will be delivered to my doorstep by a drone and put in my refrigerator by my in-house personal assistant (droid). Whether speech recognition or a touchscreen human-machine-interface is used, information will have to be localized to be conveyed to the (human) end user effectively. I am convinced that localization will be required in an increasingly expanding range of projects as time goes by.

Thank you, Dirk, for those fascinating insights into the localization process, and for your very futuristic vision of what lies ahead. I'm sure our readers were as rapt as I was as they read this column. ○

NOTES

[1] Esselink, Bert. "The Evolution of Localization," http://bit.ly/Esselink-localization.

[2] IoT: the network of devices, vehicles, buildings, and other items in which embedded electronics, software, sensors, and network connectivity enable the collection and exchange of data.

Tony Beckwith was born in Buenos Aires, Argentina, spent his formative years in Montevideo, Uruguay, then set off to see the world. He moved to Texas in 1980 and currently lives in Austin, Texas, where he works as a writer, translator, poet, and cartoonist. Contact: tony@tonybeckwith.com.