



MultiLingual

Language | Technology | Business

MultiLingual Computing, Inc. • 319 North First Avenue, Suite 2 • Sandpoint, Idaho 83864-1495 USA • 208-263-8178 • Fax 208-263-6310

Operations infrastructure for real-time translations

Bob Myers

More than two decades ago, I coauthored a paper on real-time translation. The basic idea was to embed a machine translation (MT) engine into the operating system and have it translate user interface (UI) elements in real time. This debatable concept was hatched in the context of the TRON Project, an ambitious attempt led by indefatigable Tokyo University professor Dr. Ken Sakamura to develop a home-grown Japanese computing architecture, spanning microprocessor to network operating system. The project has left its legacy in the form of a generation of mobile phones powered by its embedded OS.

Sure, that MT-engine-in-a-box would have yielded real-time translation – with the minor caveat that at the time it was completely impossible to actually implement. Of course, Sakamura specialized in dreaming about the impossible. His own favorite example of his vision of “ubiquitous computing” – he single-handedly imported the word *ubiquitous* into Japanese, which renders it in katakana as *ubikitasu* – was the microprocessor in the toilet that automatically performed chemical analysis on you-know-what and alerted the doctor if anything was wrong, presumably automatically translated into the doctor’s native language.

The intervening years have seen an onslaught of new technologies and computing capabilities, including much faster processors; the internet giving birth to an explosion of content; revolutionary language technologies such as translation memory (TM) and statistical machine translation; and, of course, stunning UI advances. And certainly the translation/localization process has both incorporated and been affected by these trends. Yet translation is still a much more awkward, slow and expensive process than it should be. Put a different way, the translation industry, broadly defined, has not really come together to solve clients’ true needs. It has not fulfilled what is both a promise and an obligation: to support and promote the global sharing of information and make the world a better place to live, especially for the huge swath of potential users in the so-called emerging economies whom the information revolution has just begun to touch.

A colloquial definition of *real time* would be *instantaneous* or *at least so fast as to be indistinguishable from instantaneous*. But here the technical definition of *real time* works better: *fast enough to be useful*. For instance, the computer controlling the antilock brake system in your car has to do its number crunching fast enough to avoid you sliding off

Bob Myers is COO of Moravia Worldwide. For this article he would like to thank Rustin Gibbs, solutions architect at Moravia Americas; Libor Safar, marketing manager at Moravia IT; and Sakiko Kimura, internet globalization specialist.



the road, but doesn't need to be any faster. For translation, we could usefully define a number of levels of real-time-ness, ranging from 250 milliseconds, which could be about right for certain applications, all the way up to weeks in some cases. In this sense, perhaps it would be better to refer to just-in-time translation or JIT, although that could have the unfortunate nuance of "at the last minute." Remember, even Japanese car companies are now backing away from the JIT model after experiencing some nasty production stoppages due to delayed parts deliveries.

Getting to real time actually accomplishes more than merely getting translations out the door and in front of the reader faster. Paradoxically, at the same time it also promotes both increased quality and lower cost. That is because the real-time goal functions as a sort of guidepost, an organizing principle, towards reengineered processes that inevitably also have a positive impact on quality and cost. Put simply, with real-time translation sometimes there is not enough time to spend as much money as we do now. And with real-time translation, we are forced to optimize quality processes, both human and automated, in a way that can actually result in higher quality than today's cumbersome processes yield.

Uncertainty and reliability

Whether we call it *real time* or *just-in-time*, clearly the very shortest turnaround times can be achieved only if no humans are involved at all. Thus, the translation must be done by machine, retrieval from TM or some combination of the two. Certain translations retrieved from TM can be considered very high reliability if the content of the TM is essentially an exact pre-translated match for the specific segment in the precisely desired context. Otherwise, the translation will by definition have some degree of uncertainty associated with it – at least until such time as MT systems achieve reliability levels of magnitude greater than at present.

From the standpoint of the consumer of the translation or some intermediary who is organizing the translation process for whatever economic or idealistic reasons of his or her own, there is a trade-off between reliability and speed (and quality), which is driven by the application. For instance, it's a reasonable trade-off to decide to translate a Facebook status update at low reliability in order to achieve very high, nearly instantaneous speed, presumably at very low or zero cost. In other cases, there may be a minimum threshold for reliability, even if that increases the time

If I'm sending out Facebook updates or tweets to be translated in real time, each task may be a dozen words or less.

necessary to get to the translated string. However, even in the case where one is willing to translate faster at the cost of reliability, it may still be important to advertise the level of reliability. We all know that verbal communications are accompanied by a sort of reliability index in the form of our knowledge of the reliability of the person conveying the information and the accompanying body language. Original written communications, as well, are similar. But what clues are available to the end-consumer of translated materials in the digital realm to provide this index of reliability? In other words, how is the relevant metadata – that something was translated, how it was translated and what the presumed reliability is – carried around and made accessible to the information consumer?

I have no definitive answer, but here are a couple of suggestions. Perhaps the most obvious is a pop-up or tooltip of some sort when the translated text is hovered over or otherwise pointed to. The tooltip could contain information such as "Translated from the English by Google Translate: estimated reliability 68%." Other ideas include distinguishing the text with colors or other decorations such as distinctive wavy underlines or perhaps even a change in font. What about using increasingly pixelated fonts to indicate increasingly lower levels of translation reliability?

Of course, indicating reliability requires that someone knows what the reliability is. The engine should be able to produce this information. For instance, at an AMTA conference I attended ten years ago in the lovely city of Cuernavaca, one IBM system reported implementing what it called the Translation Confidence Index or TCI, which, since theirs was a rule-based system, operated by applying penalties when various translation issues were encountered during the translation process. Moravia MT's partner Asia Online has statistical engines that can provide "confidence indicators" on segments, although it is reported to sometimes give low scores to good segments that have low frequency statistically. However, in addition to such confidence indicators being produced, they must be made available to the downstream component that is making use of its output.

Language professionals responsible for polishing or post-editing the MT output are presumably using some sort of environment that can easily display metadata such as the confidence indexes from the engine. If they are not, they should switch to one with deeper MT integration. But merely knowing the reliability of a translation and being able to make this information available to the language professional or end consumer are not enough. Even if we choose to go with an initial unreliable translation and claim that we've done our job by making sure the translator or consumer knows that the translation is iffy, we would prefer not to stop there, but rather to continue – assuming the economic drivers of the translation, also known as "who's paying for it," support it – translating it into ever-increasing levels of quality.

In the early days of the web when people were using dial-up, there was a concept of progressive JPEGs based on multiple compression passes at progressively higher levels of detail, for large images to be displayed while downloading over a slow connection, allowing a reasonable preview after receiving only a portion of the data. In a similar vein, one could imagine the Facebook status update to initially display at low resolution – in other words, a pure MT translation – and then to update itself, hopefully in real time as the viewer views his or her friend's page, to a

higher resolution version with the benefit of some quick crowdsourcing, let's say. This implies that when considering real-time translation, we may need to consider not just a single quality/cost/turnaround outcome, but potentially two or even more to be accomplished in succession. This notion has actually already been implemented in systems such as online support bases, where the first pass might be pure MT, the second either additional post-editing or human retranslation, driven by consumer feedback such as the frequency with which the page is viewed or page quality ratings.

Why real-time translation?

But reliability and confidence are to some extent peripheral issues. Let's return to a more basic question: What are the factors that are converging today to drive the urgency of real-time translation? To start, we may forget that there are still huge numbers of large software companies building those old-fashioned desktop applications, which survive for extremely good reasons, and they are eager for faster turnaround times in order to speed the cycle of building and testing localized versions. Whether the application is desktop-based or web-based, faster turnaround has the effect of speeding time-to-market, which is a surprisingly strong driver of return on investment. For example, earlier time-to-market can accelerate the stream of upgrade fees, prevent end users from defecting to the competition and help attract new users. But in the context of applications and their interfaces, we are talking about turnaround times that are unlikely to need to go below three hours or even 24 hours. For such applications, this can be considered real time.

Where we dip below the three-hour mark is with translation of what might be called content, although the boundary between content and applications is increasingly blurry. A company announcing a new alliance might want its press release translated within one hour. A news site posting breaking news about an earthquake might want the information available in reliably translated form in ten minutes. An obsessive Twitterer might feel his or her tweets deserve to be translated in one minute. And the Facebook addict we mentioned above might truly believe

the world would come to an end if his or her latest status update was not blasted out to other Facebook compatriots in their native languages in less than ten seconds. This is the spectrum of "real-time-ness" that we discussed at the beginning of the article.

Getting to real time

Whichever level of the hierarchy we – or rather a client or other initiator of the translation process – are shooting for, there are some obvious inevitable characteristics we can identify regarding the relevant systems and processes. First, at least once the three-hour barrier or maybe even the 24-hour barrier is broken, no one has time to send files here and there or run scripts or send e-mails or even stop and think, really. Every second must be value-added. All the assets must be in a single place in the sky available to whatever robot or human needs to do something to them or with them. Second, to the extent multiple processes are needed to get to the required level of translation quality – whether it be T-E-P or MT-PE or whatever model – those processes must be overlapped and pipelined, and this requires, on top of the assets being shared in the sky, a segment-level granularity of workflow allowing radical concurrency, a new workflow model not supported by the majority of extant workflow engines. Finally, we must attempt to optimize the efficiency of individual worksteps by assigning them to smarter, more productive workers and providing those workers with richer environments in which to work. To summarize, real-time systems must be cloud-based, concurrent and optimized.

For the traditional localization or translation company stuck in the past, getting from here to there is going to be a long, hard slog. No degree of optimizing existing manual sequential processes can reach true real-time translation. Sorry, you've reached a brick wall here.

The necessity of avoiding human intervention is all the more crucial when you consider that the trend toward real-time translation is inevitably accompanied by a reduction in the size of the typical job or batch or task. After all, if I'm sending out content every day instead of every week, each day's portion will be only one-seventh as large. If I'm sending out Facebook updates or tweets to be translated in

real time, each task may be a dozen words or less. Even web application updates, if they occur frequently, might be just a few hundred words. This forces us to examine the assumption in classic localization workflows that each batch is of a size that can support some level of overhead in multiple human handling steps and interventions. These new, smaller batches have been called drips – as opposed to the classic drops – and the ongoing series of drips is the stream, replacing the chunk. Not only can a drip not support ponderous human language processes, it cannot support even the simplest human administrative or finance task. Just as micropayments are batched for efficiency in settlement, microchunks must also be batched for purposes of administration and billing.

Another point well worth making is that traditional measures of productivity and expectations of turnaround in translation and localization projects have been polluted by the mixing in of one-time, start-up and ramp-up tasks and the associated costs. The industry has done a poor job indeed in setting mutual expectations between parties at each stage in the value chain for the cost and time involved in such tasks. There is often an implicit assumption that ramp-up is free and/or instantaneous. After all, asks the client, if you're a competent localization company, why can't you start doing my stuff on Monday and have the first delivery back on Friday? The requirement for real-time translation forces us to confront and distinguish one-time, preparatory tasks and explicitly allocate time and budget for them, whether they be related to terminology, style, custom verification rules, acquisition of resources, training of those resources or anything else. That is a good thing. Once the investment is agreed upon in terms of the time and money required by all parties, we can move assets rapidly down the assembly line with no unnecessary interruptions except to re-ramp – retool the line – when requirements change.

The assignment problem

To get to real time we need to identify each individual element in the workflow and characterize its time impact – time spent in processing, hand-offs and human handling – and figure out how to remove it. For instance, Hertz reengineered the process of getting

people into cars, which used to require waiting in line for an agent to give you the keys to the car, by simply putting the keys in the car for you and letting you get in your car and drive away, with a post-delivery sanity check at the exit gate. I recently discovered that this sanity check does actually work when I accidentally hopped into the wrong car on a Hertz lot. Getting the renter into his or her car can be easily equated to getting the work to the end-language worker.

In the localization process in common use today by large multilanguage vendors (MLVs), language-specific work is commonly done by single-language vendors (SLVs). This is increasingly the case as the number of languages that content is being translated into rises dramatically from the five or ten of yesteryear to today's 50 or 100 or even 150. No MLV can build offices or even easily build freelance networks in every region of Africa or the Indian subcontinent. The SLV layer constitutes the single most important challenge to achieving real-time translation. Files go there and stop because there is a national holiday in Tajikistan. The project manager is sick. There is a glitch in the mail server. A day is lost when the engineer had to leave early to go to his or her child's school event. Remember that this is not only a problem for traditional human-based translation, but equally for post-editing of MT content. A crucial challenge for the industry as it moves toward real-time translation is to reduce friction through the SLV gateway to near zero. This is not to say that SLVs will not continue to play a critical role in our industry. How could they not do so? They know their country, their language, their market, their culture and their freelance base. But this knowledge and the value that they provide based on it and the compensation they receive for that value must be divorced from the physical, file-based workflow.

Part of that workflow is assigning translators and editors. Currently, this is accomplished virtually without exception via a human expert who knows the freelancers, their availability, their skills, their dislikes, their track record, their daily schedules and their peccadilloes. This human expert can and must be replaced by a system that can automatically assign jobs based on extensive information about past work, performance, experience, quality

ratings and productivities, doing this across multiple workers for large jobs and doing dynamic reassignments partway through when necessary. Such a system could produce assignments that have certain demonstrable statistical properties, such as meeting predetermined quality requirements with, say, 95% confidence – or 98% if someone were available to pay extra money or willing to accept lower quality to get the extra three points.



Figure 1: Different crowd types.

If doing this requires breaching the inner sanctum of the SLV's castle, namely the confidentiality of its secret pool of top freelancers, so be it. A secret pool is hardly of any value if the people in it cannot be assigned quickly enough to satisfy the needs of the stakeholders in the end-to-end translation process. There are plenty of ways for SLVs to monetize their value-added besides taxing every word or erecting tollbooths on the translation roadway and making everyone stop to throw in their quarters.

Which team?

Our entire discussion thus far may be overly focused on the model of a small number of relatively highly-qualified professional language workers forming the pool from which a team is selected. Of course, that model is and will remain the most appropriate for certain categories of translation and localization.

But as anyone who has thought about crowdsourcing is perfectly aware, there are alternative models where the pool is formed, along various dimensions, by the crowd. Here it is important to distinguish between different flavors

of crowds (Figure 1). The Preselected Expert Team is perhaps closest to the types of teams that a big language service provider would currently deploy to translate, say, a large specialized software application. The Specialist Community differs in that it is broader and may be less eager to be compensated monetarily at a level which would support them. Pointy Heads are the equivalent in their domain of Linux geeks, and the Unwashed Masses are those on whom we would call to translate those crucial tweets.

As the pool of available resources increases, there are certain statistically predictable effects on both the expected turnaround (both time-to-assignment and time-to-completion) and cost. I am not an economist and am not acquainted with the models that could predict the magnitude of those effects. Doubling the size of the available pool might have only a marginal effect on time-to-assignment, perhaps reducing it by 10%. Still, every little bit counts. The impact of larger pool sizes on cost could be more dramatic, especially when combined with methodologies such as staggered auctions or incremental pool widening, also known as "waiting a bit before you give it to the really expensive guy."

Quality and real-time-ness

Quality and real-time-ness go hand-in-hand. The additional review, error checking and retranslation steps that currently contribute to quality run counter to the real-time objective. All else being equal, running assets faster through the factory could actually increase the need for such time-consuming quality steps.

Our language industry has worked hard to define error measurement frameworks and build error-checking tools, but the fruits of all these efforts remain elusive. Dozens of different translation environments all have their own *ad hoc* hard-wired rules. People fill in Excel sheets to count error categories. Error-checking is separated in space and time from the language worker, and the materials found to be in error must be rerouted to the language workers. How can an industry that at least pays lip service to the notion of interoperability and standards have failed so miserably to establish a standard for error

detection rules and interfaces for error detection engines? Such work will be key to raising quality levels, especially as the pool of workers expands to those who don't have error rules burned into their fingertips. To put it another way, better standardized, extensible error checking allows for the use of wider resource pools, which in turn reduces costs and improves turnaround closer to real-time levels.

The move toward real-time translation also has certain implications for the end-translator or post-editor. He or she will be required to be in an always-on mode, to receive and start work on requests in five minutes rather than 30 or a 120. In one real-time translation scenario I'm personally familiar with, the maximum expected end-to-end turnaround for translation on alerts about breaking financial news and its possible impact on stock prices was 90 minutes. The time-to-assignment and time-to-start-of-translation could not exceed five minutes, requiring the translator to either be sitting at his or her computer to get the urgent pop-up

in the corner of the screen or to have a phone with him or her when walking the dog so as to be able to receive the text message request – having planned the walk with the dog so as to be able at any point to get back home in front of the computer within five minutes. Yes, it may be necessary to compensate the translator/post-editor for offering this level of responsiveness, either in the form of a higher unit rate or, preferably in the spirit of more precisely mapping compensation to value, a retainer or availability premium charged by the hour. It will also likely be necessary to widen the geographical spread of freelance translators to ensure access to an English > French translator even after the City of Light has gone to bed.

It may also be necessary for the individual translator to change the way he or she works. Starting from the top of a document and working slowly down through it at high-quality levels increase the risk that the deadline might come and go while still in the middle of

the document with nothing to deliver. It may be better for the translator to do a series of passes through the document, an initial lower-quality pass, followed by additional polishing and rewriting passes, so when the deadline comes there is at least something to deliver. Of course, the ability offered by the inspiring latest generation of completely online translation environments to monitor segment-by-segment progress, and if necessary reassign work from one translator to another at the segment level, could make this less of an issue.

As end-to-end processes morph and evolve, hopefully faster than they have to date, to better meet the needs of the range of constituencies in the translation task, nexuses of value creation will appear, disappear and shift. The wise will navigate these shifting sands with care. What will indisputably continue to grow is the value of and rewards for designing, creating, deploying and managing the technologies, processes and resources needed to truly deliver on the promise of real-time translation. **M**

Efficiency

Out with the old, in with the new. Sound processes and smart technology are at the heart of any truly efficient localization effort. Suboptimal processes mean, well, you never get to optimum performance. Use our Process Optimization Practice to address your language technology challenges. Being smart about efficiency is always the smart option.

moraviaworldwide.com/efficiency

Moravia
worldwide

moraviaworldwide.com | AMERICAS | EUROPE | ASIA