WORDFAST®

It's Free, But Does It Work?

An exclusive interview with Yves Champollion, developer of a free translation memory tool.

by **Bob Clark**

The immortal words of John McEnroe, "You can't be serious!" pretty much sum up the general reaction to the initial appearance of WordFast on the email circuit several months ago. The skepticism was understandable. Here we have a full-blown translation memory system that appears out of nowhere, seems to be fully contained in Microsoft Word and it's free. Bert and I discussed including an item in *Language International* at the time and he, quite rightly, pointed out that over the years he had seen loads of macro-based "translation memory" solutions and none of them worked very well. I had a quick peek at the software and it seemed good enough to contact the developer, Yves Champollion, to find out more about it.

Champollion is not the sort of person you would normally associate with the development of translation software. Neither software engineer nor computational linguist, Champollion spent his early years drifting around the world, doing odd jobs and living by his wits. His Japanese-born wife was the one who "saved him from himself" eighteen years ago and dragged him back to France where he ran an import/export business dealing in Japanese fine art. And how did he get involved in translation? "I suppose it was inevitable," says Champollion. "An uncle four generations back, Jean François Champollion, was the one who cracked the code of the Rosetta Stone. I guess people figured that if he could do that, then I could probably translate automotive manuals. Translation is a tough job. You do the same thing over and over for years. It's not my nature at all.

Take any programmer who is in his mid-forties and you will find that he is an ex-New Age guy, an ex-Flower Power kid who sold his VW van and bought an Apple II in 1981.

That's why I went into developing a tool that helps you to translate. Translation memory is only one stage. My interest is actually machine translation."

So, how *did* this ex-drifter, with no conventional higher education manage to produce WordFast? "I am self-taught. In the 1980s I was a fan of micro-computing and I studied all sorts of programming languages on my own. Take any programmer who is in his mid-forties and you will find that he is an ex-New Age guy, an ex-Flower Power kid who sold his VW van and bought an Apple II in 1981. All these guys are reconstructed hippies. That's sort of what happened to me. In the early eighties I sold my sleeping bag and I bought my first PC. WordFast was the result of a dare, if you like. Someone told me he was writing macros using Microsoft's macro language, VBA. This guy said that a translation memory engine could never be written using MS Word's macro language because it was just too slow. I just said, 'no' and began to write a sort of rough sketch of a translation engine using the resources of Microsoft Word. That was back in August 1999 and, after a week of writing routines, I actually found that it worked. Of course, VBA is a slow language in itself if you compare it to C or Pascal or Delphi, but speed has nothing to do with the language of the platform itself, it has to do with the way that you write your software. I discovered ways of executing a database search extremely fast. By September 1999 I had a translation memory engine that ran as fast as any other but only used Microsoft Word as a

platform. Typically, a translation memory engine has to scan or search a huge database that can be hundreds or thousands of megabytes, looking for an exact match, which is no big deal, but also looking for a fuzzy match. You have a rough idea of what you are looking for but the software has to pinpoint the location of something that approaches your search. That is extremely difficult. I actually wrote the necessary algorithm that can perform this very fast just using Microsoft Word's macro language. That was a breakthrough. I could use the first sketch for version 1 of WordFast. The format of the translation memory is an open format. You could take a WordFast translation memory and open it with Excel, Word or Access, etc. It's not a proprietary format. That applies to the glossaries as well as the translation memory. It's all pure text."

Can this be true? Just how open is this open system? Can anyone take the lid off and customize its behavior? "No, it's not an open source project," says Champollion.

The format of the translation memory is an open format. You could take a WordFast translation memory and open it with Excel, Word or Access, etc.

"The source is mine but I've made what I call entry points, which means that any programmer could ask WordFast to segment a document, to find a match in a translation memory. In fact, all the basic tasks of the translation memory engine can be accessed externally. Besides the manual, there is what I call the White Book, which is more technical and shows programmers how they can write applications to enable them to use the WordFast translation memory engine."

And what about that inevitable question, 'Isn't this yet another TRADOS clone?' "All products with a segment-based translation memory will inevitably look alike," replies Champollion. "I use colors to represent match types. I use green for exact matches, there is no copyright on green. A green light means, 'It's OK, move on.' Yellow or orange means 'caution'. I use a sort of grayish background to represent a no-match. I don't think this breaks any rules. Many other products use colors the same way. My shortcuts are not identical. There is some overlap with TRADOS but

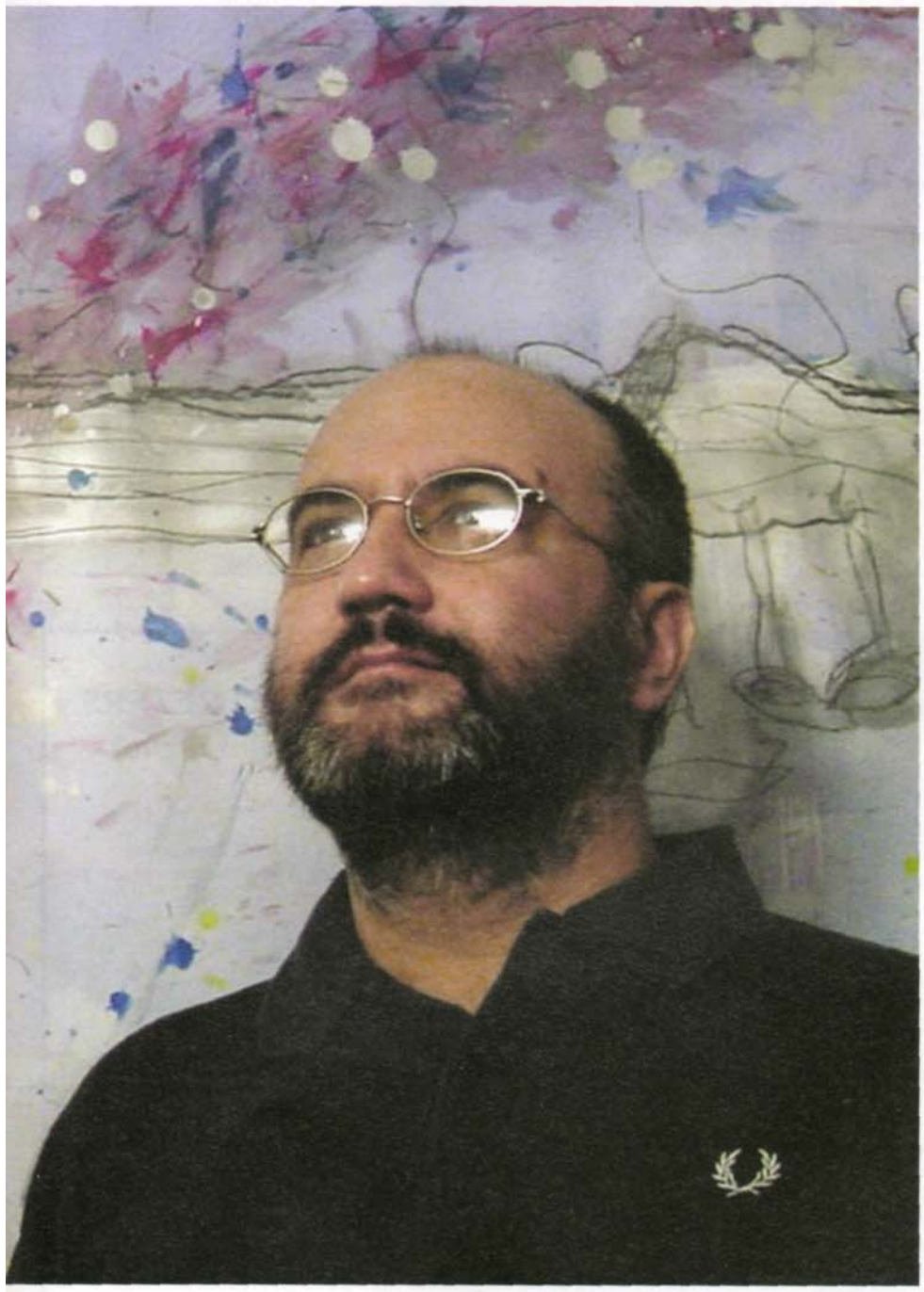
others are different. In any case, there are quite a few features in WordFast that TRADOS doesn't have, so you couldn't really call it a clone. I built on a concept that existed long before TRADOS came along. Look at word processors. They all look and feel the same. WordFast also has a totally different terminology approach. In addition to its built-in glossaries, WordFast can be interfaced with any third party dictionary. Translation memories can also be shared through a network. There is no complicated setup to do this. You open the same translation memory and that's it. From that point, WordFast knows that someone else is using the same translation memory. It's just like sharing an Access file. Up to 20 people can share a translation memory over a network and a translator's input is automatically shared by all the others on the same LAN. Web-based sharing will be done in the next version. The other concern that has been addressed is interchange with other translation memory software. WordFast can read IBM TranslationManager .exp files. It can read TMX files, which means it can read practically any memory from any tool. It also reads TRADOS native .tmw files under certain conditions, and text exports. The translation memory of WordFast is so simple that you can open it with Excel. You could paste segments into that database and resave it as a text file and it works. It is a very transparent and open format. In the latest release, the translation memory can be maintained in Unicode format or simple text format. WordFast will automatically detect a Unicode translation memory."

If you want to enter the entire US Government terminology list, if you want to have one gigabyte of glossary on line, WordFast will open that in no time and quickly find an entry.

So far, so good. What are these other features? "We offer a suite of tools, called PlusTools," says Champollion. "Initially, PlusTools was mainly geared for doing search and replace among vast numbers of files. You know, if you have a project with 200 files and the customer comes and says, 'This has to be replaced by that'. You have to open up every file and do it by hand. So this can batch-process up to 1000 files in one go, with all the refinements that Word offers. There is a tagging utility that allows you to tag and then un-tag HTML, SGML and XML files. This prepares files for translation and then reconstructs the files after translation. There is a conversion utility. You can convert files from one Word format to another in batch mode. You could also set up a conversion table to convert, for example, non-Unicode Russian text to Russian Unicode text. There is also an alignment utility to align existing file pairs to create translation memory. This will produce a WordFast translation memory but WordFast can export this to TMX format. So, you could use this tool to align files and export the output to any translation memory software. There is also a glossary that has just been ported to Unicode, containing CJK characters, that complements the existing WordFast glossaries. The WordFast glossary format is tab-delimited pure text, whether it's Unicode or not. If you have an Excel file with a glossary set up in columns and you save that as a text file, you have a WordFast glossary file. You could also have it in an MS Word table and save it, which would produce the same sort of file. At the moment, the established fields are Creation Date, Creator Identity, then you can

Photo: Courtesy B. Vaccari

add fields like Client, Subject and so on. You can then merge glossaries or extract sub-glossaries from the main one. The technology that I developed for the translation memory has been reused in the glossary. There is no limitation of file size. If you want to enter the entire US Government terminology list, if you want to have one gigabyte of glossary on line,



WordFast will open that in no time and quickly find an entry. That sounds amazing but that's the way it is. There is also a built-in terminology compliance feature. You enter the customer's terminology in WordFast using wildcards so the system recognizes all forms of the same term. As the translator moves from one segment to the next, WordFast will scan the source segment and, if it finds a customer's term in the source text, it will expect to find the preferred target term in the target segment. If that is not the case, the translator is warned that there is a discrepancy. The translator then has the opportunity to change the target term, or not, and then move on. This process can also be done at a later stage in batch mode and a report is produced. This approach not only applies to terminology but also typographical rules and non-translatable elements, for example, numeric parameters and tags, if it is a tagged file."

So, we have an apparently fully-operational translation memory system, but why is it *free*? "The main reason is that Logos noticed WordFast, and expressed interest in it. They offered not only to share their own technological developments in the same area, but to sponsor WordFast so that it is made available for free to the translation community. Although a later version could possibly go commercial, the basic version will most likely remain free. When I first made it available, there were about 20 units per week being downloaded. Now it's about 40 per day and we have well over 1000 users, so Logos was right in deciding to keep WordFast free. The user base is also contributing to technical support. We have a very active email-based user group constantly exchanging information and supplying feedback. A nobler, less commercial motive was to make the technology freely available to those parts of the world where the cost of currently available software makes adoption impossible. They have the need but not the necessary financial resources. Of course, it would also benefit the universities training future translators."

When I first made it available, there were about 20 units per week being downloaded. Now it's about 40 per day and we have well over 1000 users.

So, does it work? Only you can say. Fortunately, it won't cost you anything to find out. Healthy cynicism is always wise when investing in new tools. However, I would have thought that people should be encouraged let their imaginations run wild and be adventurous. That's how innovation comes about and, hopefully, stagnation is eroded. Maybe, just maybe, Yves Champollion and people like him are showing the way. Still, not everybody has an ancestor that cracked the Rosetta Stone code. The good news is that the Logos group has been impressed enough to underwrite the future of the software and their developer, Bruno Vaccari, is collaborating with Champollion on the next version of WordFast.

WordFast can be downloaded from http://champollion.net.