[From: *International Journal of Translation*, vol.13, no.1-2, Jan-Dec 2001]

# Creating Enterprise Machine Translation Systems

WINFIELD SCOTT BENNETT

## INTRODUCTION

You have just come up with the greatest theoretical approach to machine translation ever. Naturally you want to use the concept to create the killer app enterprise MT system and perhaps make some money in the process. So what do you do?

The first thing is to stop and think. Enterprise MT systems, i.e., those used for high volume publication quality MT, are orders of magnitude different from the systems found in research groups. Such systems are intended for use in corporate and government settings for the production of multilingual documents. They must compete in a market. As such they must address certain issues successfully or they will fail.

What must be considered in developing such an enterprise system? If it is to become available to the target users in a reasonable period of time, it is necessary to plan the development very carefully. In this brief space I explore a number of issues that must be addressed for any work in creating an enterprise MT system.

## PICK A TARGET

It is essential that the target users be clearly defined. Enterprise systems are generally intended for use by professional translators who take the MT output and post-edit it into the final published version. This means that the system must meet more stringent criteria than those for systems intended for gisting or casual use. This sounds as though it is an easy step, but, in fact, I believe it to be much more difficult than most projects have imagined. The term "professional translators" covers a large group of people who do not always have the same criteria.

The best approach is to determine the user group and find out from representatives of the group just what they need and want for an MT system. An enterprise system must meet the needs of the user or it will not be accepted. Users' wants are also important since they more they are covered, the more acceptable the system. This takes time and effort, but will result in considerable timesavings in the development work. From my experience and observations years of wasted development have resulted from the failure to do this first step.

MT system developers must always remember that MT is not an essential technology. Either the system meets the demands of the target users or it will not be used. To guide the development process it is highly recommended that representatives of the target group be part of the process with significant voice in decisions on how the project proceeds. The cost of such staff members is little compared to the loss of time and money developing something which will not be used.

*Which languages?*

This is again not as trivial as it sounds. And it is clearly essential to success of an enterprise system. While many of us in the MT field are not geared to think in such terms, it is critical to look at the market. What language directions are in highest demand? What are not? Is there something on the horizon?

At this time the biggest demand for commercial language directions is: English to and from the following languages:

Chinese
French
German
Italian
Japanese
Korean
Portuguese (Brazilian)
Spanish

These language directions dominate the general commercial translation market now. Of course, this does not mean that other language directions are not important and this does not mean that certain market segments may focus on other languages. Further, the most global corporations translate into many more languages than these.

These data are an indication of what language directions an enterprise MT system may want to address. And it is also an indication that any system's design must be prepared to deal with non-Roman characters, if it is to meet market demands.

### HOW BIG A BITE?

A third issue that must be addressed up front is just how wide an application the system is to be. In the past most enterprise MT systems were designed to handle all of the language as the developers imagined it to be. The results have varied depending on how well the development team was able to cover the language.

Some systems have taken the opposite approach; these have focused on the sublanguage for a particular field or markets. The implications of this decision are clear. If an enterprise system to be used only for a particular field (e.g., aeronautical, medical or geology), the amount of time to create a production system is reduced because the system dictionary does not have to cover all the terminology for all the fields and the grammar does not have to parse those structures which do not occur in the particular field. This is particularly attractive for fields which have adopted controlled language in their documentation. The downside of focusing on a sublanguage is that the system is not as widely marketable as a more general one. Such considerations have to be made early in the development cycle since a change in strategy can add considerably to the cost of the work. Even changing from a general-purpose system to a sublanguage system requires time and effort.

### *What about the fuel?*

All machine translation systems are fueled by the dictionary data. Even in a system which is delivered with a large dictionary, there will be an immediate need to add to the dictionary. Thereafter it will be necessary to update the dictionary on a regular cycle. The truth is that any given corporation or government agency will have its own terminology which will vary to some degree from others in its field. A customized and customizable dictionary is essential to the success of the application. Unless the developer is prepared to upgrade the dictionary for each client constantly on demand, the system must include a way to build and maintain the dictionary - easily.

This view is often dismissed with an “everyone knows that” nod, but the fact remains that many systems are launched as enterprise systems without true regard of what this means. Providing what the users need to augment and maintain the dictionary is never trivial. Some questions that should be addressed as early as possible in the development are:

- Does the user have to have extensive linguistic background for dictionary work? (Some systems seem to expect an experienced computational linguist to do the dictionary inputting.)
- How long does it take to enter a new term? (Time is more than money here. Too much time per entry makes for very unhappy users.)
- What can be done to relieve the user of repetitive inputting? (For instance, does the user have to re-enter the part-of-speech every time? Or will the system assume that it is the same as the last one?)

Part of the difficulty in this is that often the development staff are not the kind of people who will do dictionary work. Computational linguists are often not, in my opinion, well suited to be computational lexicographers. A development effort should include one or more computational lexicographers whose voice in matters of the dictionary is fully heeded.

### *What's it run on?*

In the early days of machine translation the choices for enterprise systems were different types of mainframes. The choice broadened with the advent of minicomputers and workstations. Of course this was in an era when personal computers of whatever type were not available. With the rise of personal computers the possibilities for

MT systems increased, although many enterprise systems did not move immediately to them. Partly because the PC did not offer the kind of power that a true enterprise system requires. Now the Internet offers another possibility.

Thanks to the Internet, many enterprise systems now run in what a client-server configuration. The server carries the weight of system which allows a minimal client to benefit from the powerful system on the server. However, before embracing this configuration too quickly, it is important to recognize that for a client-server system to be valid it must allow the user all that he or she needs for the job. For example, any client-server configuration to be a true enterprise MT system, the user should have full access to the dictionary including the ability to add or modify entries as needed. Limitations in client-server configurations may make the system less usable than users will accept. Before choosing a platform it is essential to be sure that the result will be usable.

### *How easy is it to use?*

Machine translation developers too often think entirely in terms of the technology of the system. The result may then be great technology which simply does not sell. In the world of enterprise machine translation systems no one buys technology as such. Enterprise systems, by definition, are for use in particular environments by real people (not researchers) on a constant basis. They must be products.

In products, as Microsoft as demonstrated continually for years, the technology is secondary to how it is delivered. The hard fact is that people will use easy-to-use less-than-perfect systems over hard-to-use outstanding systems - every time. Enterprise machine translation systems, then, must provide the users all the functionality they need in a way that accommodates them. The statement that "they'll get used to it" is often the kiss of death for a system. In reality potential users simply choose not to use the system at all, rather than dealing with its foibles.

The lesson here is that the plan for developing the enterprise MT system must include plans for the best possible interfaces and tools to meet the needs of the user. Since users will interact with the various components in the system continually it is important to include ergonomics in the design. Enterprise systems have not succeeded just because they were too hard to use over time.

## STARTUP, CARE AND FEEDING

Getting started with an enterprise system is not an easy matter. Systems at this level are not plug-and-play. This means that someone has to train the user and provide assistance when all does not go well. All this must come from some sort of customer support. Too often this is not given much thought until the first systems are sold.

Customer support, as I see it, takes two forms:

- Documentation
- Human interaction.

The form of the documentation is not entirely relevant; the content is all important. Poorly written documentation can contribute to the failure of a system, especially since most users will have little or no prior experience with any machine translation system. Writing the documentation should start very early in the development process and be continually updated throughout. Critiques of the documentation by target users at various stages are essential.

Even with the best documentation human intervention with users is inevitable. Too often the plan is to use developers are the customer support people. Such an idea seems to have its merits, but consider two points in this. First, interrupting development efforts on a regular basis to handle customer issues may cause unacceptable delays in development. The new improved version of the system may take twice as long as it should because development staff is constantly fielding customer questions. Two, developers are many times not the kind of people who interact well with customers. Highly skilled computational linguists, lexicographers and computer programmers may not be mentally equipped to deal with naive users who do not have the same backgrounds.

This is a matter, then, for serious consideration at the beginning of the project. The fabulous enterprise version of the machine translation system will not succeed in the market if the customers cannot use it to their satisfaction. And their satisfaction will rest with customer support.

### *What's this baby cost?*

Is this yet another non-development issue? I add this to the list simply because it should be addressed up front. It is a business issue, but it is fully relevant to the whole idea of planning the development of the best ever enterprise machine translation system. What the market will bear has impact on what can be devoted to the development work. Producing a wonderful system which fully addresses all the issues above can still result in failure, if the system cannot sell in sufficient quantities to repay the cost of development.

Systems developed in the late 70’s and in the 80’s often sold for US $ 100000 or more. The price tag was an effort to recoup the cost of development (sometimes around US $ 25000000). Few systems sold at that price.

The prices now are much lower. Enterprise MT systems are often priced around US $5000 per license and below. Prices seem to be dropping still. Before entering the enterprise market it is well to consider whether the development effort will be such that the system can never pay for itself. Someone has to decide on whether to proceed or not.

## CONCLUSION

This paper has offered a number of issues which must be addressed in the development of an enterprise MT system. I do not claim that it is exhaustive; these are ones that strike me as important. It is not my intent to cast a pail on the idea of developing enterprise machine translation systems. My intention is to lay out the issues my experience has shown me to be important. If they are fully addressed in the planning stage of the development effort, the resulting system should have a reasonable chance of success.

So take your killer idea for MT and run with it! But do not forget to think about these issues first.