Tracing Translations in the Making

Elliot MACKLOVITCH **Ngoc Tran NGUYEN** **Guy LAPALME**
RALI - DIRO
Université de Montréal
Montréal, Québec, Canada, H3C 3J7
{macklovi,nguyennt,lapalme}@iro.umontreal.ca

Abstract

This paper presents TTPlayer, a trace file analysis tool used to develop TransType, an innovative computer-aided translation system. We first discuss the context of the project and the design of the tracing tool. We show how it was used for discovering interesting patterns of use as well to guide further developments in the TT2 project.

1 Introduction

In the world of sports, it is common practice to film games or competitions in order to analyse the tactics of an opposing team, or simply to improve one's own performance. The ability to stop, slow down and replay the action often allows one to detect things that would otherwise escape attention: a telltale gesture a pitcher makes just before throwing to first base; a tendency to lift the head early and follow the ball, thereby making one's drive erratic.

Now wouldn't it be nice if we could do the same thing in other areas of human activity? In the sciences, for example, where the aim is often to achieve a better understanding of the behaviour of a human expert so that we can provide her with an intelligent form of automated assistance. Alas, more often than not, it turns out that this is just not possible either for practical reasons or for ethical ones. With the exception of professional athletes, artistic performers and exhibitionists, people do not generally like to be closely observed at their work for extended periods of time; it riles them and tends to make them uneasy. Nor do personal interviews in which human experts are asked to explicitly describe their work practices or thought processes offer a satisfactory alternative. For one thing, the expert's answers are often unreliable, because she may not be aware of, or be able to make explicit what they do instinctively; for another, the expert may provide the responses she thinks the interviewer wants to hear.

Whence the interest of an automatic tracing program, particularly in the context of an interactive computer application. The tracing program doesn't overtly observe the expert, thereby ensuring that her tasks will be performed naturally. Instead, it unobtrusively records the expert's every action as she works with the system, as well as the system's response to each such action, and discreetly stores them in an electronic file that can later be analyzed in detail. Just as with the filming of a sports event, careful analysis of this trace file can often prove very revealing; or at least this has been our experience. We have designed such a tracing facility as part of our contribution to a research project called TransType, the goal of which is to explore the feasibility of a new type interactive machine translation (Foster et al., 1997; Langlais et al., 2002a; Langlais et al., 2001). Our automatic tracing program, christened TTPlayer, analyses the trace file of a translator's working session using TransType and generates detailed statistics on a host of interesting questions. Furthermore, TTPlayer can also read the trace file and visually *replay* the translation session, much like a video cassette recorder. In effect, it's almost as though we were actually present and able to peer over the translator's shoulder, except that we can stop, slow down, accelerate and rewind the action at will. This too facilitates our efforts to better understand the behaviour and strategies of the expert user, in this case a professional translator.

The remainder of this paper is structured as follows. In the next section, we will provide a general overview of the TransType system. In section 3, we will then describe TTPlayer in some detail: not just how the program functions, but also its role in the usability trials that were central to the TT2 project. We conclude by trying to understand how this approach can be extended to other contexts.

2 TransType

While TransType is certainly not the first attempt at interactive machine translation (IMT), there are a number of important characteristics that do serve to distinguish it from its predecessors. In all previous IMT systems, the focus of the interaction between the user and the system has been on the source text. In particular, whenever such an IMT system is unable to disambiguate a portion of source text, it requests assistance from the user; and in principle, once the system has obtained from the user the information required to disambiguate the source text, it can then complete its analysis and continue to properly (or hopefully) generate the corresponding text into the target language. Now this is not the place to enumerate all the difficulties that have dogged this classic approach to interactive MT; but let us suppose, as is often the case, that the user in question is a professional translator. Notice that the kind of information being solicited from her by these classic IMT systems does not focus on translation knowledge per se, but usually involves formal linguistic analysis, of a kind that many translators have not been trained to perform; e.g. What is the morpho-syntactic category of such-and-such form? Or does this prepositional phrase modify the verb phrase or the preceding noun phrase? Although professional translators are highly skilled language specialists, these are not the kinds of questions that they (as opposed to formal linguists) have been trained to resolve.

In TransType, in contrast, the focus of the interaction between the user and the system is squarely on the drafting of the target text. The way it works is roughly as follows: After reading the current source text segment, the translator will begin to type her desired translation. Based on its analysis of the same source segment and using its statistical translation and language models, TransType will immediately propose an extension to the characters the user has keyed in. The user may accept all or part of the proposed completion, or she may reject it as inappropriate by simply continuing to type. However, each new character that the translator enters provides the system with additional information on the target translation which she has in mind; and that is precisely what the system is trying to divine. TransType responds by recalculating its predictions in order to make them compatible with this new input and then proposes a new completion – all in real time. When the system performs well, the user will normally accept these machine-generated proposals, thereby reducing the number of characters she has to type and hopefully increasing her overall productivity. But the important point is that in this approach both the user and the system contribute in turn to the drafting of the target text, and the translator is not solicited for information in an area in which she is not an expert.

A second important difference between TransType and previous IMT systems has already been alluded to, namely that TransType uses statistical, or probabilistic translation and language models, of a type that were not generally available when most classic IMT systems were developed. However, the significance of this shift in paradigm goes well beyond simply keeping up-to-date with the latest NLP trends. The probabilistic models employed in TransType are critical to the system's ability to adapt its completions to the user's input. A basic postulate of statistical MT is that there is no one correct translation for a given source text segment, but rather a multitude of translations, all more or less probable. Needless to say, the system will first propose to the user the completion(s) which it calculates to be most likely, based on the large volumes of previous translations it has been trained on. But should the user ignore the proposed extension, TransType will then respond by proposing the next most likely completion that is compatible with the prefix the user has keyed; and so on, in order of decreasing probability. In contrast, almost all classical MT systems (interactive or not) were of the rule-based variety, and were generally programmed to generate a single target language equivalent for some source text segment. This too was somewhat unnatural from the user's point of view, contradicting a very basic fact about the nature of translation.

So much for some of the properties that distinguish TransType from previous attempts at interactive MT. As we have said, the system is intended to be used by professional translators whose job is to produce high-quality translations. This too is an important fact about the TransType project, distinguishing it from other MT applications where less than high-

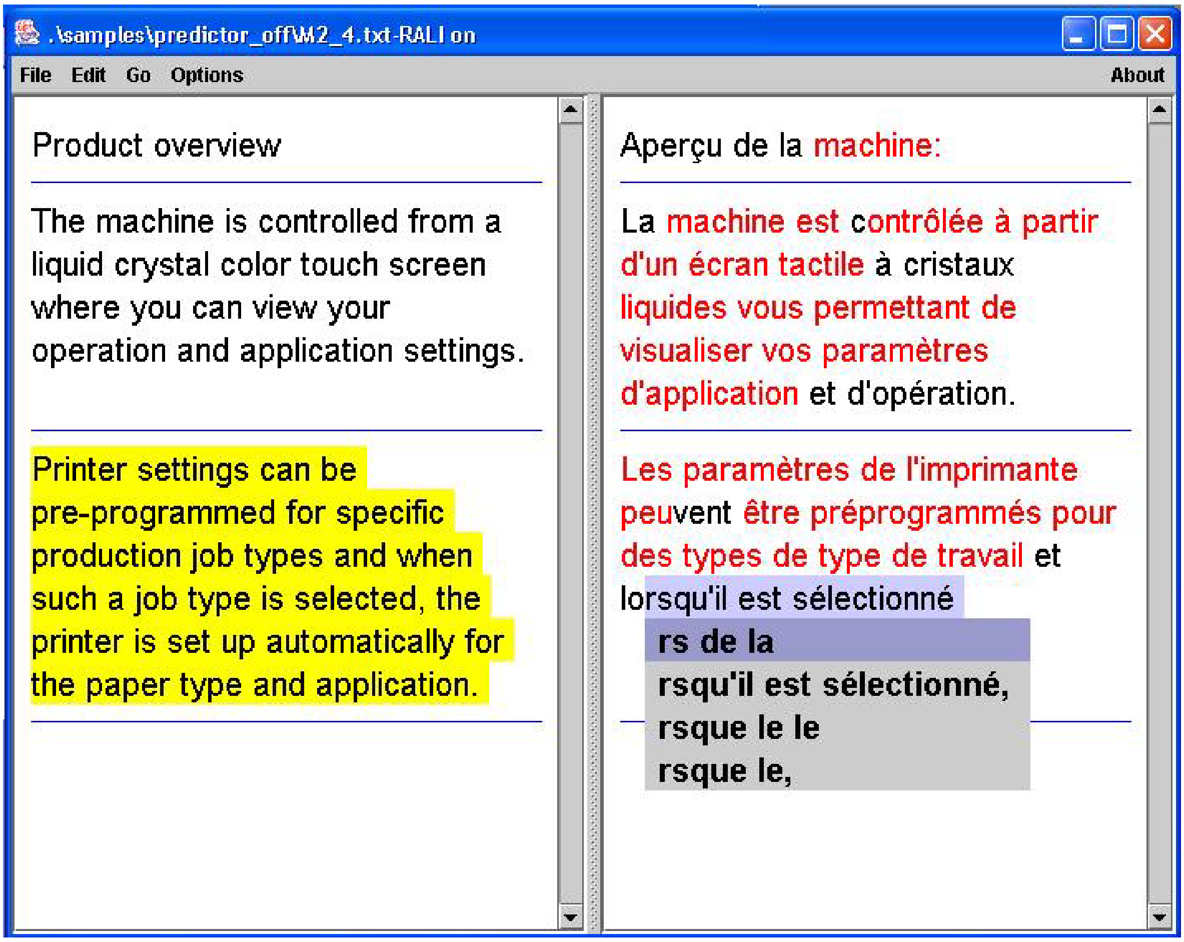

Figure 1: Screenshot of a TRANSTYPE English-to-French translation session. The source text is on the left half of the screen, while the target text is typed on the right. The system's proposals appear in a pop-up menu at the cursor position in the target text.

quality output may be altogether adequate, e.g. for information gathering purposes. But given the state of the art, high-quality translation is not consistently possible without human intervention. In our IMT project, the translators serve as the guarantors of linguistic quality, while the system's proposed completions should serve to make them more productive.

TRANSTYPE takes the form of a two-paned text editor in which the source text appears in the left-hand pane, divided into sentence-like segments (see Figure 1). As soon as the user selects a given segment, the system responds by placing a proposed translation for it in the corresponding cell in the right-hand pane. TRANSTYPE uses statistical translation and language models (Foster, 2000) to compute the completions that are presented to the user for selection. The current version of the system handles three language pairs: English-French, English-Spanish and English-German, and all three in both directions.

The graphic user interface (GUI) offers a fairly basic set of text editing operations, e.g. copy, cut-and-paste. Ours being a research project, the fundamental aim of which was to explore the feasibility of interactive machine translation, we did not go to the trouble of trying to emulate all the options offered in word processing packages like MS-Word. But one important feature that we did add was a provision for keeping track of all the actions of a TRANSTYPE user and their precise time of occurrence; these are automatically recorded in a trace file of the session. Figure 2 shows a (very short) excerpt of such a trace file, which can easily contain 10 000 to 20 000 lines.

The GUI also allows the translator to modify a number of basic parameters that directly affect the manner in which the system proposes it completions. Thus, the user may instruct the system to display one or more alternative completions, and only if they contain some minimum (or maximum) number of words. Should the system not perform well on a given text, the user can actually turn the predictor off and still call up the predictions on demand by hitting a keyboard shortcut. She may even instruct the system to remain silent unless there is no user input for a certain number of seconds (as when the translator is stumped). Many of these features have been added to the system as a result of suggestions made to the developers by our users.

Quarterly usability trials were a central component in the 3-year (2002-2005) TT2 project[1] that was launched under the European Union's Fifth Framework Program. The TT2 consortium included two bona fide translation firms, one in Spain and one in Ottawa, and every four months their translators were provided with the latest version of the system, which they tested in their own offices for two weeks. These usability trials followed an elaborate protocol, but one of the fundamental questions they were intended to answer was whether TransType actually allowed these professionals to increase their productivity. As a baseline reference, the participants were first asked to translate a substantial portion of text within the TT2 editor but without the benefit of the system's predictions. For the last evaluation round, we had 6 professionals each translating 8 texts, all of which gave rise to around 950 000 lines of actions to analyze. TTPlayer was thus essential for automating the computation of statistics. This was much more convenient than loading the trace into Excel or developing special purpose programs, as we did in our first experiments.

Using the information from the trace, we were able to compute the translators' production time on the so-called *dry run*, i.e. when they translated without the help of the system's predictions, and convert it into a ratio of words per hour. The same was also done for all the texts that were translated with the predictor turned on in each evaluation round; a comparison of the two ratios told us whether TransType's interactive proposals enabled these users to increase their productivity and if so, by how much.

[1]For more information about the project, see : http://tt2.atosorigin.es

These trials also allowed us to measure many other parameters, e.g. the average time required to accept a prediction, the average number of words in the accepted predictions, the extent to which the users avail themselves of the keyboard shortcuts and the mouse, etc.

3 TTPlayer

Given the innovative tight coupling of TransType's prediction engine and the actions of the translators, we did not have *a priori* a clear user interaction model to build on. The only way to understand how TransType is used is to observe its usage. Thus the idea of playing back the trace in order to see how the translators actually worked with the tool in a real translation setting. This idea of playing back the translators' actions originated with Philippe Langlais, one of our colleagues in the TransType team, and grew out of the difficulty of making sense of all the actions in the trace. This can be readily appreciated by looking at the few lines of Figure 2 and comparing it with what can be seen in Figure 3. The situation is even clearer when you can see it develop dynamically; unfortunately, this discovery process is hard to convey in a paper... A first version of TTPlayer with the associated traces was offered as a human interface resource at LREC2002 (Langlais et al., 2002b). The current system is a complete redesign of that original idea.

We now describe the current state of TTPlayer illustrated in Figures 3 and 4. Opening a trace file creates an instance of a TransType GUI which is controlled with a *simili media player* that, instead of playing a movie, goes through the events recorded in the trace file and replays them either in *real-time* or at a certain rate: a rate of 3 or 4 events per second is usually the best for demonstration purposes and for understanding the behavior of a translator. It is also possible to skip forward and backward to a certain event or go directly to a certain event number. All these actions are possible via the GUI widgets in the *Controls* panel at the bottom left of Figures 3 and 4. TTPlayer also computes statistics, such as the number of completions accepted with the mouse or the keyboard, the number of characters typed and erased and the number of characters finally entered per minute, which we call *productivity*. These figures are dynamically updated as the trace is playing, and they are shown

time	action	arguments		
39.998	ENTER_SENTENCE	0		Close the transfer station properly (see "Opening and closing the transfer station" on page 6.47).
45.326	COMPLETION	0	0	Fermez correctement le module de transfert (voir "ouverture et fermeture de la station de transfert" à la page 6.47).
50.383	KEY_ACCEPTED_COMPLETION	0	0	Fermez correctement le module de transfert (voir "ouverture et fermeture de la station de transfert" à la page 6.47).
54.188	BACKSPACE	0	50	o
54.569	ADD_CHAR	0	50	O
54.599	ADD_CHAR	0	51	P
55.540	BACKSPACE	0	51	P
65.625	BACKSPACE	0	21	e
65.775	ADD_CHAR	0	21	a
66.045	ADD_CHAR	0	22	s
66.546	BACKSPACE	0	22	s
66.776	ADD_CHAR	0	22	
66.857	ADD_CHAR	0	23	s
67.057	ADD_CHAR	0	24	a
67.247	ADD_CHAR	0	25	t
67.638	BACKSPACE	0	25	t
67.768	BACKSPACE	0	24	a
67.848	ADD_CHAR	0	24	t
67.968	ADD_CHAR	0	25	a
68.048	ADD_CHAR	0	26	t
68.209	ADD_CHAR	0	27	i
68.249	ADD_CHAR	0	28	o
68.359	ADD_CHAR	0	29	n
69.951	DELETE	0	30	
70.181	DELETE	0	30	m
70.382	DELETE	0	30	o
70.862	DELETE	0	30	d
70.892	DELETE	0	30	u
70.932	DELETE	0	30	l
71.353	DELETE	0	30	e
...	...	...		...

Figure 2: Excerpt from a trace file. The first column indicates the time in seconds since the start of the program; the second column is the name of an action and the other columns are supplementary information specific to each action. The first line gives the English sentence to translate for which the system suggests a completion (second line) that is accepted (third line). The rest of the trace shows the corrections made by the translator by removing and adding characters (the third column gives the sentence number and the fourth column the character number in the sentence). This result is the first French sentence of Figure 3.

in the bottom right part of the Figures 3 and 4.

3.1 Implementation

The TransType GUI is driven by events that can originate either from actions by real users at the keyboard or mouse, or from TTPlayer which generates the same events as it reads a trace file in which the users' actions have previously been recorded. This is why TTPlayer is implemented as an event generator for one or more instances of the TransType GUI running in the same setting. For example, it is the same code that is running in the different TransType GUIs in Figures 1, 3 and 4, the latter running two instances. Our TTPlayer application is implemented as a Java Swing Multiple Document Interface (MDI). In order to guarantee a faithful rendition of the actions of the trace, TTPlayer disables all user events sent to the GUI except for scroll-bar manipulations. On the other hand, the user can control the manner in which the trace is played via the forward, rewind, fast forward and pause buttons. Using the same framework of sending events to a common interface, we have also implemented a non-graphical mode to compute global statistics over the whole trace file without displaying the user actions.

3.2 Role in usability studies

An earlier version of TTPlayer was used by Foster (Foster, 2002) for developing a user

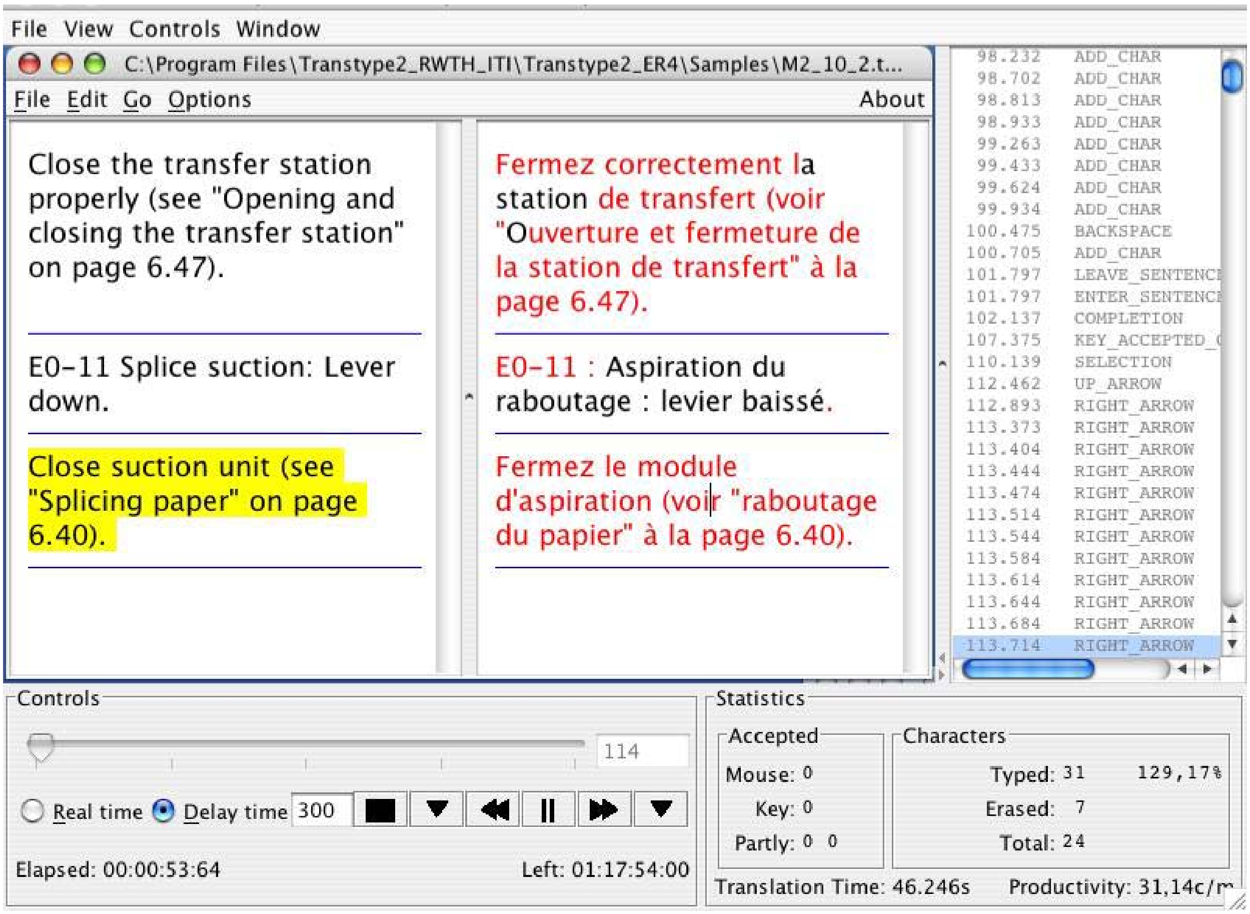

Figure 3: Screenshot of a TTPlayer session. The content of the trace appears on the right; the bottom panels show the controls of the player on the left and the statistics of the translator that are updated on the right. The center is taken by the trace file that is being played back. Characters that were suggested by TransType and accepted by the translator appear in red (or in shaded grey on a black and white printed copy)

model that takes into account such parameters as the time needed to read a suggestion, the probability of rejecting or accepting it, the probability of continuing to type even if this means typing the same characters as those proposed by the system, etc. Although these parameters for the different components of the translation engine are estimated directly from the trace itself and do not require a trace player, we do not think that we could have developed such a user model without seeing the translations unfold before our eyes.

As for the current version of the system, the only real users of TTPlayer have been members of its development team or analysts who wish to understand how the underlying system can be improved, e.g. with a better translation engine or a more appropriate GUI.

The TT2 consortium included three university research labs, each of which developed its own statistical translation engines: ITI at the Polytechnical University of Valencia in Spain, RWTH at the University of Aachen in Germany, and RALI at the University of Montreal in Canada. The Madrid office of Atos Origin was responsible for overall project management, as well as assisting in the integration of the engines within a single GUI. The consortium also included an industrial research partner - Xerox Research Centre Europe - that furnished multilingual corpus materials as well as various forms of corpus processing. TTPlayer has played a crucial role in helping to communicate needs and problems between members of this team; the tracing facility helped ensure that developers and users understood the same thing.

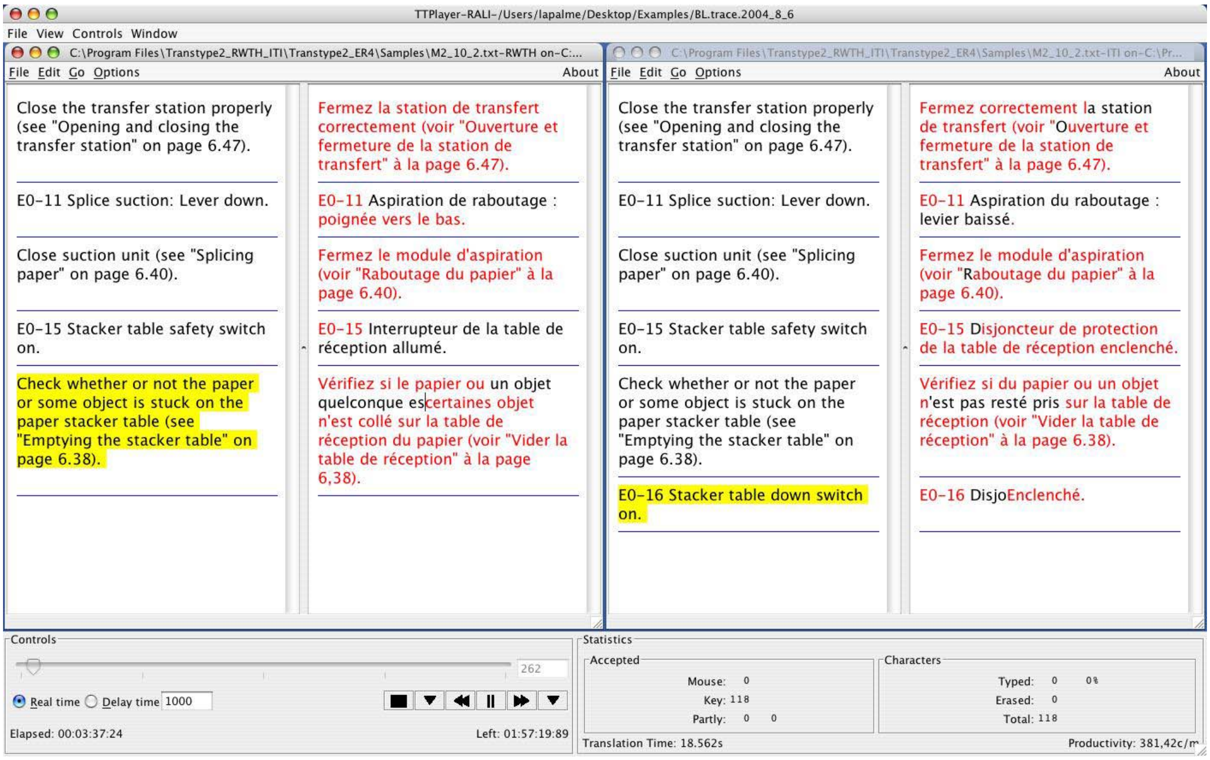

Figure 4: Screenshot of a TTPlayer session for comparing two translators working on the same input file. As both displays are shown in *real time*, it is interesting to watch the two translations being built in parallel. The statistics and controls are displayed for only one of the translations, the active one, here the one on the left. A translation can be made active by clicking on the title bar of the window.

TTPlayer was also used to demonstrate that the more translators make use of TransType, the better their productivity. It is interesting to compare how two or more translators deal with the same text. We can even compare a translator with herself by playing two traces and studying their behavior in accepting the suggestions.

More generally, TTPlayer has also been very useful for public demonstrations. It provides a very good way of showing a *real professional* translator in action, as opposed to a technical developer struggling with either the interface or some simple translation problem.

Given the many research ideas that were prompted by looking at these trace files, we are now convinced that any interactive application should be provided with such a facility. Observing *real people* using our systems is always instructive, even when they do not use them as we wish they could (or should!!!). Some of the features we worked hard to develop based on our own experience (e.g. using the mouse for entering partial completions or using cut and paste) were almost never used by the professionals. Given the time and energy involved in user trials, it is important to be able to get the most out of them by keeping all information in a way that can be analyzed off-line. Reiter (Reiter et al., 2003) rightly argues for clinical trials of AI systems, even when the results do not always go *in the right direction*.

In our case, we were able to come up with interesting arguments through careful analysis of the trace files made possible with TTPlayer. At the outset of the project, for example, we were convinced that the best way of justifying this new kind of interactive MT was to show that it actually speeded up the keying in of translations. In fact, at first we used to describe TransType as a typing accelerator for translators. But while increased productivity is certainly an important argument, it is not the only way to validate this type of translation aid. Statistics from the trace files showed us that TransType could also reduce the effort required to produce a given target text, independent of the speed at which it is typed. On the dry-run, where the participants receive no suggestions from the system, TTPlayer informed us that for one translator, an average of 6.65 actions per word (ie. keystrokes or mouse clicks) were required to produce her translation. With the system's predictor turned on, the same participant required an average of only 3.48 actions per word to produce her target texts. In other words, TransType was allowing this participant to produce her translations with about half the (physical) effort she required to translate on her own. And, of course, translation quality can also benefit from the system's predictions, by reducing the time that users have to devote to terminological research. A more detailed analysis of the results obtained with TransType in our usability trials can be found in (Macklovitch, 2004).

4 Conclusion

This paper has shown the interest of tracing as a tool for obtaining knowledge about potential users in the context of developing applications that embed advanced AI components. Given the fact that the medium of interaction in our application is a text editor, instrumenting it through a trace file was a natural and unobtrusive way of keeping track of users' actions. This general approach can certainly be extended to other interactive applications, particularly those that involve text.

5 Acknowledgment

We want to thank all members of the TT2 consortium, especially the translators who participated in the user trials at both Gamma and Celer translation bureaus. We also want to express our gratitude to current and former members of the RALI who participated in the development of TT2: Philippe Langlais, Simona Gandrabur, George Foster, Marie Loranger, Laurent Nepveu and Sébastien Sauvé.

References

George Foster, Pierre Isabelle, and Pierre Plamondon. 1997. Target-text Mediated Interactive Machine Translation. *Machine Translation*, 12:175–194.

George Foster. 2000. A Maximum Entropy / Minimum Divergence translation model. In *Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 37–42, Hong Kong.

George Foster. 2002. *Text Prediction for Translators*. Ph.D. thesis, Université de Montréal, May.

Philippe Langlais, Guy Lapalme, and Sébastien Sauvé. 2001. User interface aspects of a translation typing system. In E. Stroulia and S. Matwin, editors, *AI2001 - Advances in Artificial Intelligence*, number 2056 in Lecture Notes in Computer Science, pages 246–256, Ottawa, Ontario, 7-9 juin.

Philippe Langlais, Guy Lapalme, and Marie Loranger. 2002a. Transtype: Development-evaluation cycles to boost translator's productivity. *Machine Translation (Special Issue on Embedded Machine Translation Systems)*, 17(2):77–98.

Philippe Langlais, Marie Loranger, and Guy Lapalme. 2002b. Translators at work with TransType: Resource and evaluation. In *Third International Conference on Language Resources and Evaluation (LREC)*, pages 2128–2134, Las Palmas, Canary Islands, Spain, May.

Elliott Macklovitch. 2004. The contribution of end-users to the transtype2 project. In Robert E. Frederking and Kathryn B. Taylor, editors, *Machine Translation: from Real Users to Research: 6th Conference of AMTA*, Lecture Notes in AI 3265, Springer, pages 197–207, Washington, DC, September. Association for Machine Translation in the Americas, Springer.

Ehud Reiter, Roma Robertson, and Liesl M. Osman. 2003. Lessons from a failure: generating tailored smoking cessation letters. *Artif. Intell.*, 144(1-2):41–58.