The world on your shoulders: The development of ATLAS

Eric M. Visser, Masaru Fuji, Makoto Shiotsu†
{eric || fuji}@flab.fujitsu.co.jp,† HHA01542@niftyserve.or.jp

Fujitsu Laboratories Ltd. † Fujitsu Limited
Software Laboratory Middleware Group

1015 Kamikodanaka
Nakahara-ku, Kawasaki 211
Japan
tel. +81 (44) 777 1111

Abstract
In this paper we present some of the history and more of the present of the development work on Fujitsu's ATLAS Machine Translation systems family. The first two versions of ATLAS ran on mainframe computers, and for this reason Fujitsu has long been synonymous with "big systems" in the Machine Translation world. However, the focus of our development efforts has been shifting towards smaller platforms for several years already, culminating in the release last year of an MT system for personal computers running under Microsoft Windows. We will nevertheless argue that ATLAS is still a "big system", in terms of both the development environment and the resulting software and data.

Having argued that ATLAS is a big MT project and thus a big system, even though it is available on a "small" platform, we next show that this has important consequences for the character of our development effort. Factors like functionality, range of object texts, user profile and user requirements influence each other and the development process as a whole. We find that feedback from users of ATLAS for Windows, who have different requirements than mainframe users, influences the entire ATLAS project in ways which pose special challenges for our development team. In addition, we feel that the processes we use for internal evaluation of ATLAS output need to be rethought in order to better complement user feedback and if possible anticipate user requirements.

1 All about ATLAS

Machine translation research started at Fujitsu as two independent projects, both of them in the late 1970s. These projects led to two separate systems, ATLAS-I and ATLAS II, which were released in 1984 and 1985, respectively. Both run on Fujitsu's M series of mainframe computers. ATLAS-I, which translates from English to Japanese (E/J) only, was the first commercial E/J system in the world. It follows the so-called direct translation method, meaning that it is a syntax-based system using no explicit intermediate representation of the processed sentence. ATLAS II, which originated in Fujitsu Laboratories, translates

System	Speed	Platform
ATLAS II	60,000 wrds/hr	M200
S-ATLAS (J/E)	94,000 wrds/hr	S-4/20 model 50
S-ATLAS (E/J)	51,000 wrds/hr	S-4/20 model 50
ATLAS for Windows (J/E)	10,000 wrds/hr	486DX2 at 66MHz
ATLAS for Windows (E/J)	11,000 wrds/hr	486DX2 at 66MHz
Dictionary size	135,000 words (basic, all systems) specialised dictionaries in 24 fields	

Table 1: ATLAS vital data

only J/E and, as all consequent ATLAS systems, uses the interlingua transfer method. In a nutshell, this semantics-based method involves transforming the input sentence into a language-independent intermediate representation, called a "conceptual structure" in ATLAS; applying a *conceptual transfer* if necessary, to bring the concepts more in line with the semantic structure of the target language; and finally generating an output sentence from the conceptual structure. More detailed information about the interlingua transfer method can be found in the proceedings of the first MT Summit (Hakone, Japan, 1987, pages 128-134).

A Japanese-Korean MT system based on ATLAS-I was marketed in 1988, but mostly our research and development efforts have focussed on the interlingua transfer method. ATLAS moved beyond the world of mainframe computing with the introduction of the J/E version of ATLAS-G (which runs on Fujitsu's G series of workstations) in 1989. The E/J version was released in early 1992, as well as J/E and E/J versions of S-ATLAS, for Sun workstations. These two systems can basically be thought of as ATLAS II clones which have been ported to their respective platforms. The most recent addition to the ATLAS family is ATLAS for Windows, which was put on the market in June 1994. Some data on the various systems are given in table 1.

2 Is ATLAS a "big" system?

As we explained above, ATLAS has evolved from a bulky mainframe-based translation system 6 years ago to a PC software package now. The fact that we are nevertheless speaking in the "Big systems development" session at this convention begs the question asked in the title of this section: can ATLAS be called a big system, and if so, why?

There are several aspects that influence the perception of a system as "big" or "small" (we shall call this *system scale).*

Platform What size computer does the system run on?

User base Is the system used by individuals, translation companies, or in big translation projects?

System usage Is the system intended for stand-alone use, or is it to be a tool in or part of an integrated environment?

Development environment Is development done on mainframes, workstations, desk-top computers?

Project size How many man-years are involved?

System size How large is the grammar rule base, how many dictionaries are there, how large are they?

Concerning the view that the **platform** decides the scale of a system ("a system that runs on a mainframe is a big system"), we would simply like to say that we feel this view is outdated. In terms of the raw translation module, ATLAS II and ATLAS for Windows are one and the same development project most of the way. Thus, from the point of view of a developer, it makes no sense to call ATLAS II a big system and ATLAS for Windows a small one, even though they may look quite dissimilar on the outside. We conclude that platform size is not a relevant factor in determining system scale.

The question of **user base** seems to relate more to the system's price/performance ratio than to anything else. Of course a cheap system is a lot more likely to be bought by casual home users, even if it performs less well than a more expensive one, and small systems are more likely to be cheap than big systems. It's also safe to say that cheap systems with poor performance are not likely to be used at translation companies, for instance, so there is obviously a correlation; but the relation does not seem to be intrinsic. In other words, calling a system "small" because it is used mostly by home users does not seem to be productive. It is true though that the character of the user base has an influence on development; we will come back to this later, in § 3.

System usage is an important characteristic to distinguish between types of MT systems, but ATLAS is a hybrid system in this sense: ATLAS for Windows, as most PC applications, is used mainly as a stand-alone tool, while S-ATLAS especially also finds many uses in other application systems such as data base retrieval tools and electronic mail translation systems (see § 3). At this point we wish to note that in our opinion, although the distinction is important and has bearings on the question at hand, we believe that it is not enough to alone distinguish small systems from big ones.

In our view, the **development environment**, **project size** and **system size** are the most important factors determining system scale. On a heavy and networked environment more and more powerful tools will be available, while a larger number of people working on a project will result in fundamentally different development and management techniques. On the downside, there are likely to be more dependencies in large grammar rule bases, and thus making changes in these rules carries a higher risk of unexpected problems. Generally speaking, we believe a difference in *size* in these three factors will often account for a difference in *kind* in the resulting MT systems. We will come back to these points again later.

Concerning the last factor, **system size**, we may add that ATLAS for Windows is at the high end of the Japanese PC-MT software market in terms of dictionary size (twice as big as most of the competition, 5 times as big as the cheapest system), and range of applications. Unfortunately, figures comparing the respective sizes of grammar rule bases are not available, but we have reason to believe ATLAS is among the biggest in this aspect as well. That said, we believe that system size alone is but an indicator of system scale.

To summarise the above, we maintain that ATLAS for Windows is a "big system", and justify that by pointing out that machine translation at Fujitsu is a big project. The

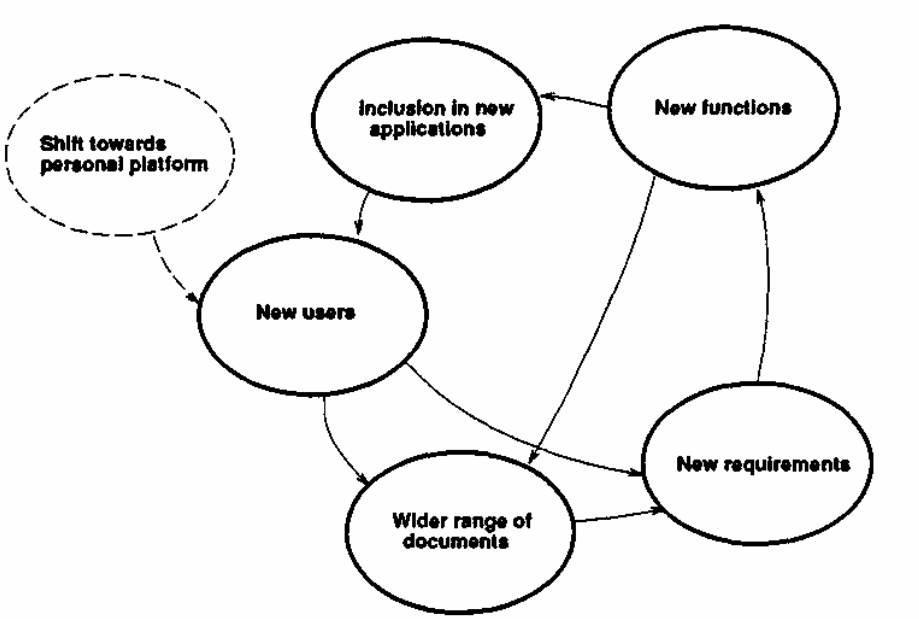

Figure 1: Factors influencing development

project is big in terms of time, people and money, the resulting translation module is big in terms of its grammar rule base and dictionaries, and ATLAS itself can be integrated into larger systems. We also believe that these are important factors differentiating ATLAS from most of its competitors in Japan.

In the following section, we will discuss some specific aspects of MT systems development at Fujitsu with a focus on the influence that the shift to ATLAS for Windows is exerting on the development process as a whole.

3 Factors underlying development shifts

As described in the previous sections, ATLAS has been developed and commercialised on a wide range of platforms, gradually shifting from large computers to personal computers. This section describes various factors which have accompanied this shift, and how they influenced the development process.

These factors, and some of the ways in which they influence each other, are shown in figure 1. When ATLAS is included as a translation option in a new application, this opens up the system to a new range of potential users. This in turn leads to an increase in the types of document the system is asked to handle. These two factors then conspire to create new requirements for the system, both linguistically and in terms of functionality. The new requirements are then taken into account in consequent development work, and new functions are added to the system to meet the requirements. These new functions in turn may open the way for the system to be used in new applications. Thus the above factors are linked in a chain of dependency, which finally feeds back into the development work.

Each of these factors will be described in more detail below.

3.1 Applications and functionalities

In the early days of development, when mainframe computers were the only platform. capable of handling heavy processes such as MT, ATLAS was designed to be used on a stand-alone basis. That is to say, ATLAS software was installed on individual computers. and used solely by the users who had direct access to it. These users generally used ATLAS only for one fixed purpose.

However, the emergence of the workstation version of ATLAS has dramatically changed the situation. The various functionalities offered by workstations have created a plethora of new applications for ATLAS. This has resulted in ATLAS being available as a part of various application systems, and in various forms. The PC version itself is mostly used as a stand-alone system again, but its users — supported also by the PC's user-friendly interface — are more likely to use the system for varying purposes, and thus the emergence of the PC version has even accelerated the trend towards multi-functionality. Conversely, PC-based MT systems also increase the PC's functionality.

The following are some of the applications in which S-ATLAS is being used today or could be used tomorrow.

Mail service The UNIX operating system, on which S-ATLAS was developed, offers powerful networking capabilities, which has allowed us to build an experimental translation service that can be used via e-mail. The user requiring translation service just sends in his texts to a special e-mail address, the text is translated automatically, and the translated results are sent back to the user also by e-mail. This service was introduced to a commercial VAN in Japan in 1990.

Database service This is an experimental service that allows users to retrieve information, in the users' own language, from databases containing texts in a foreign language. The user enters a query in his own language, which is then translated into the language used in the database. This translated query is used to retrieve texts from the database. The retrieved texts are translated to the user's own language and displayed to the user.

Machine Translation for Internet services Internet services, such as WWW, are displaying rapid growth in Japan as well as elsewhere in the world. Currently, much of the information available via Internet is expressed in English; this presents difficulties to Japanese end-users, many of whom feel most comfortable with Japanese texts (even though they probably can read English, if they have to). Fujitsu is studying the possibility of adding automatic translation capacity to its Internet services.

Translation aids for professional translators A translation firm within the Fujitsu family has developed an integrated translation environment called TranText which incorporates ATLAS. This environment was originally intended for their own use, but commercialisation is under consideration now. This is yet another example of an ATLAS application which has been made possible by the shift towards workstations and PCs. TranText provides user-friendly functions for the use of professional translators. ATLAS is merely one of the many facilities incorporated to the total system, which also include automatic dictionary lookup, a text management system, desk-top publishing, bilingual text editors, grammar checkers, etc. Networking is one of the most important features of the system, in that this allows translators, who usually work in small groups, to share their texts and translation resources.

3.2 User Profile

ATLAS users used to be mainly translation professionals whose firm or department had acquired the mainframe-based ATLAS II. The release of the workstations versions and the subsequent inclusion of these systems in applications like ATLAS through e-mail have opened up the system to a whole new class of users. The most marked difference is that many of these users are not translation professionals, but rather computer network users. Often, these people will be using ATLAS to make a quick, rough translation of a foreign-language document they want to have a look at. We call this "browsing", and it clearly puts different demands on the system.

It goes without saying that the introduction of ATLAS for Windows enforces this trend. Moreover, it also internationalises it: whereas the vast majority of users on the VAN are Japanese and therefore not likely to request a J/E translation for browsing, overseas users of the PC package will probably be looking to the J/E feature mostly for browsing.

3.3 Documents to be translated

The developments outlined above have also affected the range of document types that ATLAS is expected to translate. The overall tendency is that, more and more, ATLAS is expected to cope with a larger number of document types.

The range of document types was quite limited in the early days of development: ATLAS was mainly used for producing translated documents, mostly in translation departments of manufacturing firms. Car maker Mazda was one of our first serious users. The texts to be translated by users like Mazda were mostly technical manuals. These exhibit characteristics favourable to machine translation, with well-structured, fairly uniform sentences and little ambiguity. ATLAS only needed to be fine-tuned to handle such texts; since the volume of translation was large, this was a feasible option.

With the widening of the user base, ATLAS has come to be used for more diverse purposes; especially, people who use an MT system for browsing are likely to feed a wide variety of texts to the system. This is even more true for people who “surf” the Internet with the MT system ready and waiting in the next window. Thus database queries and entries, personal communications, advertising texts and articles from newspapers and magazines have all become potential ATLAS input.

As a result, we have been forced to increase the generality of the ATLAS grammars, while at the same time we have investigated the peculiar structures of various types of text and written add-on grammar modules for the most important of them. These grammatical options, which can be turned on and off at will, presently number 25.

3.4 Requirements

The requirements imposed on ATLAS have also been affected by the shift towards personal platforms.

In document production departments with “big” translation environments, requirements arise not only from the translators involved, but also from systems engineers building the translation environment. Such translation departments have to build their own translation resources, such as dictionaries and grammar rules, and hence their requirements are centred around useful tools for building translation resources. The latter tools include dictionary editors and grammar editors, as well as bilingual text depositories and translation evaluation environments.

On the other hand, browsing raises quite different requirements. As the users are not professional translators, they tend not to spend much time and effort to modify the translation environment provided by ATLAS. In general, they are more concerned about how to minimise the steps involved in translation processes. We see ATLAS for Windows as a tool that can help the users increase the functionality and user-friendliness of their PC, and regard this as an important development goal.

The sheer number of users also influences the requirements that are made on ATLAS and the way developers deal with them. During the initial stages of development, the number of users was quite limited, which made it possible for the development side to meet user requirements promptly, and on a one-by-one basis. As the number of users gradually increases, it becomes increasingly difficult to grasp individual user's requirements. It is interesting to note, that this is a typical problem common to most package software.

4 Evaluation of MT output

Somewhat separately from the processes outlined above, we believe that a revision of the evaluation process should also be one of our priorities. In an MT development environment, there are basically two ways to discover what parts of your system need more attention: comments from users, and evaluation of the output. As described above, widening of the user base will almost automatically result in more and more varied user demands; but it goes without saying that the internal evaluation process will not change of its own accord. However, we believe that it is important to realise that evaluation is more than just a way to measure progress, or to compare one's own system to the competitors'. Thus, we strongly feel that the way we evaluate the system's output has to evolve along with the kinds of uses the system may be put to.

More specifically, we have noted that more and more users are looking to MT systems for the purpose of browsing. The expectations and demands of a person who uses an MT system for browsing are quite different from those of someone who intends to post-edit the output and produce a perfect translation. In a way, they will be less strict: the output does not necessarily have to be grammatical, as long as key terms and passages are understandable. In another way, they will be stricter: the output text as a whole needs to make sense, and be consistent at least to a certain extent.

It should be clear from the above that a strictly technical/linguistic per-sentence evaluation might fail to catch aspects of the text that are important to a browsing user. Thus we are working to develop new evaluation methods that are more suited to expressing the understandability and readability of the text as a whole. We will use these methods in addition to the more technical methods we have used so far.

It should also be noted that the widening of the user base will make it more difficult to react directly to user demands, simply because of the increase in their number and variety. Another aspect is user loyalty: when you've sold a more expensive system, the user is in a way hooked and it is in their own interest to report problems. This phenomenon can be observed with PC software packages as well, but to a lesser extent: users are more likely to give up on you, give you bad publicity and buy another package next time. Thus, it is more important than before to anticipate the user requests before they even occur, which is another reason why we want to put increased emphasis on in-house evaluation.

One recent addition to ATLAS, *designed at Fujitsu Laboratories, is called the* Tuning Kit. *As the name indicates, this is a system that allows the user to tune* ATLAS *translation to enable it to handle e.g. idioms which are not in the dictionary. The interesting thing about this system is that in fact it doesn't tune translation at all. It merely registers the idiom in the dictionary, and automatically pre-edits the input text into a format which is strictly speaking not grammatical, but which* ATLAS *can process with existing grammar rules.*

Figure 2: Example of an ATLAS option: the Tuning Kit outline

5 Conclusion

Most PC software packages started out as PC software packages; ATLAS, in contrast, started out as a big system which has migrated down to workstations and finally to PCs. We have discussed the influence this has had on the nature of our development work above; in conclusion, we would like to add a few words comparing our history to that of "normal" software packages.

In general, large systems will have covered a high percentage of the basic sentence patterns, and be working on rare and exceptional cases. This easily leads to interdependencies in the grammar rules, which in turn makes it difficult to implement changes: they may lead to problems elsewhere, and in the worst case even to an overall drop in the quality of the output. Unfortunately, the state of the art in software engineering as well as linguistics is not yet at the point where this problem can be avoided altogether. This was one of the considerations behind the design of our Tuning Kit (see figure 2).

Large rule bases have other disadvantages as well. Enlarging the rule base will also slow down speed, which can become a serious problem at a certain point. Likewise, big is beautiful in dictionaries, but memory size, although not as precious as it used to be, is still limited. And finally, when you have a large number of people working on a project, it becomes more difficult to coordinate their activities.

These may sound like disadvantages, but it is our belief that they are not necessarily so, and may even be turned into strengths. We regard the above as challenges to our design and management techniques, and believe that over the years we have acquired the knowhow to largely neutralise them. We have already touched on the add-on grammar options, which help to increase output quality without compromising the structure nor increasing the size of the grammar itself. We also count our large and varied user base as an asset which we thank to the scale of our system. Apart from that, we naturally number the linguistic knowledge, dictionaries and corpora we have accumulated over the years among the advantages of a large project as well.

In short, a big system's history is a heavy load, which ATLAS must carry like its namesake from Greek mythology. Nevertheless, without wishing to pass judgment on small systems, and noting that of course we couldn't change history even if we wanted to, we hope we have shown that the fact that ATLAS is a big system even in its smaller manifestations is one of its principal merits.