Meta
Journal des traducteurs
Translators' Journal

META

User Driven Development: METAL as an Integrated Multilingual System

Thomas Schneider

Études et recherches en traductique / Studies and Researches in Machine Translation
Volume 37, Number 4, décembre 1992

URI: https://id.erudit.org/iderudit/004050ar
DOI: https://doi.org/10.7202/004050ar

See table of contents

Publisher(s)

Les Presses de l'Université de Montréal

ISSN

0026-0452 (print)
1492-1421 (digital)

Explore this journal

Cite this article

Schneider, T. (1992). User Driven Development: METAL as an Integrated Multilingual System. *Meta*, *37* (4), 583–594. https://doi.org/10.7202/004050ar

Article abstract

The METAL machine translation system originally evolved from a cooperative effort with the University of Texas. After ten years of development, it became a product, intended to serve as a tool for the professional translator. The system is integrated into an office environment, incorporating processes like the automatic deformatting and reformatting of documents and interfaces to various word processing and desk top publishing systems. Recursive grammars and linguistic parallel processing plus a flexible lexicon structure lead to adequate translation quality. Experiences from some twenty installations, while very positive, have pointed out the need for further development beyond the purely linguistic: simplification of the user interface, higher error tolerance and integration with additional tools. Therefore, recent changes in the system have added components like pattern matching, version control, terminology and style checking, automated accounting and interfaces to a terminology data base. Both internal structure and user environment will be described.

USER DRIVEN DEVELOPMENT: METAL AS AN INTEGRATED MULTILINGUAL SYSTEM

THOMAS SCHNEIDER
Siemens Nixdorf, Munich, Germany

Résumé

Le système de traduction automatique METAL est né d'une coopération avec l'Université du Texas. Après 10 années de développement, c'est devenu un produit devant servir comme outil pour le traducteur professionnel.

Le système est intégré dans un environnement de bureautique et incorpore des processus tels que le déformatage et le reformatage automatiques des documents ainsi que des interfaces vers divers systèmes de traitement de texte et d'éditique.

Des grammaires récursives et un traitement linguistique en parallèle, plus une structure de dictionnaire flexible conduisent à une qualité de traduction satisfaisante. Des expériences sur une vingtaine d'installations, bien que très positives, ont révélé la nécessité de développements allant au-delà du domaine purement linguistique : simplification de l'interface utilisateur, plus grande tolérance à l'erreur, et intégration avec des outils supplémentaires.

C'est ainsi que des changements récents au système ont ajouté des composants tels que l'appariement, le contrôle des versions, des vérificateurs de style et de terminologie, une tenue des comptes automatisée, et des interfaces à une base de données terminologiques.

Nous décrirons aussi bien la structure interne que l'environnement pour l'utilisateur.

Abstract

The METAL machine translation system originally evolved from a cooperative effort with the University of Texas. After ten years of development, it became a product, intended to serve as a tool for the professional translator.

The system is integrated into an office environment, incorporating processes like the automatic deformatting and reformatting of documents and interfaces to various word processing and desk top publishing systems.

Recursive grammars and linguistic parallel processing plus a flexible lexicon structure lead to adequate translation quality. Experiences from some twenty installations, while very positive, have pointed out the need for further development beyond the purely linguistic: simplification of the user interface, higher error tolerance and integration with additional tools.

Therefore, recent changes in the system have added components like pattern matching, version control, terminology and style checking, automated accounting and interfaces to a terminology data base.

Both internal structure and user environment will be described.

1. PROTOTYPE VS. PRODUCT

The publication of the ALPAC report in 1966 (which everybody refers to but which hardly anybody seems to have read) ended a period of euphoria and lavish funding for research and development in machine translation (ALPAC 1966). The subsequent hangover days eventually gave rise to the realization that the analysis and generation of natural language, aggravated by the booby traps of structural and semantic transfer across

linguistic and cultural boundaries, is one of the most complex problems for man and machine.

Some critics of the field maintain that actually no progress has been made since the 1960's and that the resurrection of the subject area is mainly due to its redefinition in more modest terms. While it is true that the approach to MT has become more realistic, *e.g.* in gearing systems towards specific types of text and specific subject areas, undeniably we have seen remarkable progress — in linguistic treatment, in implementation as well as in operability.

Of course, a lot of linguistic problems remain to be solved, and an unappealingly large amount of research and development work lies ahead of us. Unfortunately, a fair number of the plethora of isolated studies are not very helpful. Often such publications present a cleverly thought up analysis problem, attested to by five to ten contrived examples, and show how to solve it prototypically with a radically new formalism. The only task left for the world to deal with is the treatment of the remaining millions of analysis problems.

In the area of natural language processing, the existence of a "prototype" is of no relevance. It can only be likened to declaring the existence of a black powder fireworks rocket the prototypical achievement of a lunar expedition. It is fairly simple to implement a miniature system for a minimal segment of controlled language. However, there is absolutely no certainty that an approach which may be viable for such a small linguistic sample could even be extended to cover a larger segment. Problems of ambiguity, of over-generation or of combinatorial explosion arise when the heterogeneity of linguistic expressions in large samples of a language requires large grammars and large lexicons.

Let us illustrate this point with a simple example. A standard phrase structure grammar will construct a noun phrase (NP) from a sequence of constituents like: determiner (DET) — adjective (ADJ) — noun (N), as in "That red wine is a good substitute for paint remover". It may also cover a structure like NP — ADJ N, as in "Red wine is the opium of the academic". However, we also need to treat other forms of NPs, as in "Wine adds flavor to sauces", "That is an inexpensive red wine", "I prefer that red to the other wines" or "Red is my favorite color". Just to cover these common structures, the grammar has to have the following rules building NPs:

NP — DET ADJ N
NP — ADJ N
NP — N
NP — DET
NP — DET ADJ
NP — ADJ

Now that we have extended our grammar and again tried an analysis of the simple phrase "that red wine", all of a sudden, miraculously, there are six legal NPs in that structure, "that", "red", "that red", "wine", "red wine" and "that red wine". Which interpretation is the correct one?

Our little problem may be easily solved by strategies of the longest match — provided we are dealing with simple phrases only. Once we enter into the real world of complex sentences with myriads of ambiguities across word classes and phrases, such naive strategies are hopelessly useless.

Similarly, a small restricted lexicon simply ignores problems of homography. If the single entry for "back" identifies it as a noun, there is no major ambiguity to be resolved in a sentence like "My back hurts". But what about sentences like "We will not back illiterate politicians", "I take it all back", "Back to the future", "A back issue of META"

or "Meanwhile, back at the ranch..."? If we add the other possible entries for "back" to the lexicon, homograph resolution becomes an unpleasant issue.

Strategies to analyze large segments of natural language may need to differ drastically from those sufficient for simplistic prototypical models. Moreover, real-life texts do not always conform to the rules of school grammars — sometimes due to the sloppiness of the authors, sometimes because the sublanguage is characterized not only by a separate lexicon but by a different syntax. In German, prescriptive technical texts such as maintenance documents generally use infinitives to express imperatives, as in "*Schraube anziehen* ('tighten screw')". This does not constitute an analysis problem per se, but combined with the ubiquitous use of ellipsis in technical texts, such expressions can become quite ambiguous. In the example of "*Programme speichern*" the reading of 'programs are storing' vs. 'store programs' may not be very likely, but it cannot be excluded a priori since the sublanguage-specific deletion of implied direct objects occurs very frequently.

These trivial examples (which barely scratch the surface of the immense problems encountered in natural language processing) were meant to show that prototypical solutions may, by their inherent limitation, ignore the real analysis problems and may not be suitable for a larger-scale application at all.

And that was just the linguistic aspect. A useful, productive MT system is not simply a black box that translates. First of all, is it designed in a modular way that makes it extendable and maintainable? Can errors be corrected, and are there efficient tools for lexicon update etc.? What about every developer's headache: documentation? Machine translation systems are not self-explanatory, and certainly users need extensive training and support.

If translators work for several clients, they are usually faced with the unpleasant situation that the source texts can come from a sundry collection of incompatible word processing systems. Page layout and format information are stored in a variety of ways, but need to be preserved for the target text. A productive MT system must be able to import documents from various sources, or else it is nothing but an expensive toy. And of course, prototypes are usually designed by specialists with specialists' knowledge; a product has to have a user interface for a non-specialist.

1.1. ECONOMIC CONSTRAINTS

It is a long and thorny road from prototype to product, and an expensive one at that. Whenever in industry a decision is made about the development of a new product, a market analysis is required. Machine translation, however, is a new technology, and there are few objective figures available as to its potential market. So on the one hand there is no certainty about economic success, on the other hand it is a well-known fact that innovation involves errors, *cul-de-sacs* and no certainty of feasibility. Small wonder that business controllers in industry are generally very hesitant to approve machine translation projects which are likely to require many years of R&D efforts by highly qualified specialists (who may not even be found). Almost as costly as the development itself may be the necessary marketing, user support, quality control and maintenance of system versions.

Many attempts have been made by small companies to make a commercial success out of the development of machine translation systems, and almost all of them have failed — mainly because of naive approaches to a very complex problem, but to some degree also because they underestimated the necessary investment. It seems that only very large companies with a solid financial base (or small firms backed by such companies) can muster the stamina to support long-term development, and can provide the infrastructure for adequate maintenance and user support.

2. HISTORY

Contrary to public belief, there is a noticeable shortage of technical translators which causes great concern in the industrial sector. To give an example: the complete documentation for a large-scale telecommunication system may amount to more than 100,000 pages. As on the average technical translators produce about 1,000 pages per year, the task of translating just this single set of documentation requires about 100 person-years. Any company would be hard pressed to find sufficient qualified personnel, and even if twenty specialists could be found there would be a delay of five years between delivery of the finished product and its operation. Such delays can easily lead to the loss of markets and shareholder unrest.

In an attempt to solve this in-house problem, Siemens became involved in the area of machine translation. After negative experiences with commercially available MT systems, a decision was made to start a research and development project. Its goal was to build an operative machine translation system which would be able to increase the productivity of the in-house translators and reduce turn-around time, *i.e.* the time span from receipt of a source document to delivery of the target document.

In 1978 Siemens entered into a cooperative agreement with the University of Texas at Austin. The Linguistics Research Center at UT was in the fortunate position of having been able to devote many years to research to this field. The work was conducted under the title of "METAL", and even though the present system bears little resemblance to the early versions, the name has been retained. A first prototype was presented in 1979. However, it has taken ten long years of development before METAL finally reached the status of a product.

3. SYSTEM STRUCTURE

The linguistic component of METAL is written in CommonLisp, the other functions such as the text processing component are written in C.

The system is implemented on a hardware package consisting of several translator workstations and a dedicated LISP machine running as a server in the background. Throughput on the LISP machine is about 200 pages per day. By the time this publication goes to print, the system will have been ported to a UNIX platform with comparable performance.

The LISP server in today's configuration is linked via Ethernet to a multi-user translator workstation. From these terminals, translation jobs are started and all the tasks of deformatting and reformatting and post-editing are handled. The translation process running in batch in the background is detached from other processing steps and does not interfere with any of the tasks at the translator's terminal. The translator workstation also provides the standard interfaces to other office systems.

From the outset, METAL was built in a highly modular way so as to permit the inclusion of new elements or the modification of existing elements without major ill effect on the other components. There is a language-independent linguistic processor to which language-specific modules for analysis, transfer and synthesis are added. A standardized interface representation ensures that the analysis module of a given language can be used as the basis for transfer to various target languages without any modification. This decreases development time and expense for new language pairs. Furthermore, the "open" system structure also makes METAL an adequate basis for future applications in semantic content analysis, information retrieval or as a natural-language front-end for expert systems or data bases. Its first application, however, is in machine translation.

3.1. GRAMMAR

In spite of all research, there is at present no linguistic theory available which could describe even a single language unambiguously and completely. Thus a somewhat eclectic approach has to be chosen in the grammar, drawing among other things on aspects of x-bar formalisms, deep case representation and phrase structure rules which are augmented by tests on the constituents, their interaction and various other constraints. In contrast to other systems, the rules are recursively applied so that their number can be kept low.

Conventional machine translation systems of the past usually tried to account for every possible surface structure with a separate rule. This approach assumes (falsely) that a natural language is a finite system and that a sufficiently large set of individual syntax rules would eventually cover all cases. Aside from the fact that for free word order languages this is intrinsically impossible, managing tens of thousands of individual rules is very difficult. METAL by contrast uses no more than 600 grammar rules but is nevertheless able to handle sentence structures it has never encountered before. In other words, the METAL grammar is an "open" system whose coverage extends far beyond that of a set of explicitly stated shallow rules.

Besides recursive rule application, the second principle in the METAL grammar is parallel processing.

All possible interpretations of all elements in a sentence are stored in a chart. The parser builds structures, utilizing the grammar rules and information contained in the lexicon. These structures are weighted based on probabilities and compared. Only when an interpretation spanning the whole sentence and accounting plausibly for all elements is reached, the transfer to the target language is attempted. In other words, no decision about the function of a sentence element is made until all other elements have been considered as well. If no interpretation spanning the whole sentence can be found, *e.g.* because of ungrammatical input, the system invokes a fail-soft mechanism and delivers a translation of the individual phrases it had been able to interpret. In some language combinations, *e.g.* from German to English, the output may still turn out to be grammatically correct. In the majority of cases, however, the posteditor needs to correct the output. For a more detailed description of the linguistic and computational aspects see Thurmair (1990a and 1990b).

3.2. LEXICON

Even the most sophisticated MT system cannot operate without an adequate lexicon. But the overall number of entries in a system dictionary is not a relevant criterion for a qualitative assessment or for a legitimate comparison of different systems. For one thing, the internal structure of an entry may differ. Perhaps all stems or even all tokens of a word are listed separately in the dictionary, or by contrast all forms may be subsumed under a single entry, with internal pointers to tables and rules so that full forms (including compounds) can be generated. METAL employs the latter structure.

Secondly, it makes a difference whether a system relies on one monolithic, bilingual unidirectional dictionary, with a hard-wired link between one source language word and one target language word, or whether multiple dictionaries are used. METAL operates on both monolingual lexicons and a transfer lexicon. The monolingual lexicons contain morphological, syntactic and semantic information needed for the analysis and/or generation a language. The transfer lexicon provides a link from the source language to the target language, indicating under which conditions, in which contextual environment and in which subject field a source language entry should point to a specific target language entry. As an example, the German verb "zerlegen" would be translated into English as "analyze" if the direct object had the canonical form "Satz" (sentence).

It would be transferred as "dissect" if the direct object had the semantic type 'human' or 'animate' but it would be translated as "disassemble" if the direct object was concrete.

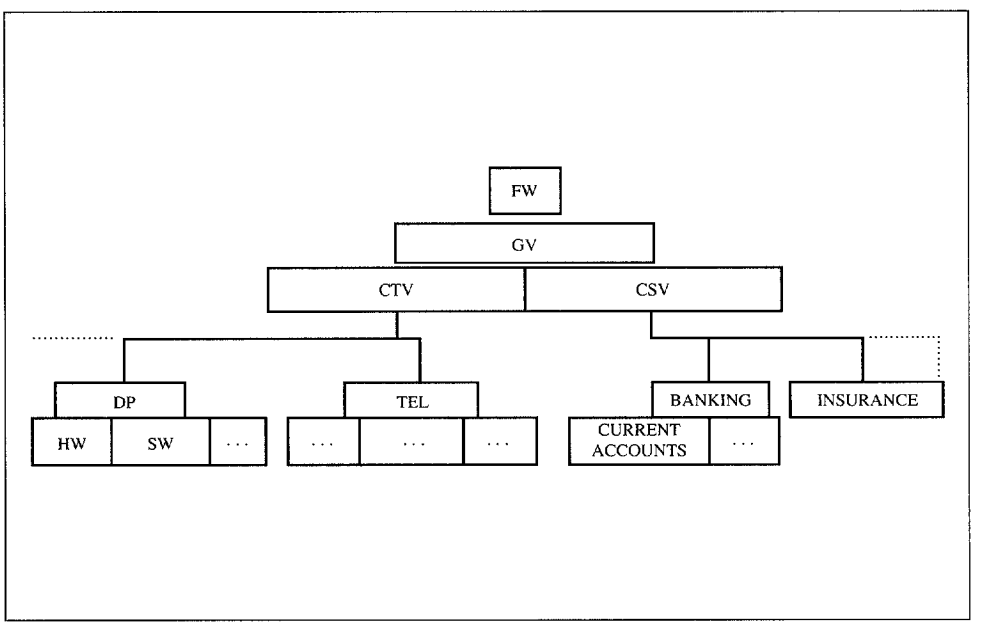
The advantages of such a lexicon structure are obvious. The extensive grammatical information contained in the monolingual dictionaries needs to be carried only once, even if many different entries in one of the languages correspond to the same entry in the other language. Moreover, if monolingual and transfer lexicons are kept separate, the monolingual entries can be re-used in other language combinations without modifications.

Another aspect of a lexicon to be considered is the organization of its terminological content. In most European languages, the set of the most frequent 5000 words makes up approximately 90% of any given text (on the average). Beyond this limited set, the point of diminishing returns is soon reached. Increasing an undifferentiated general lexicon to more than 100 000 words, for example, would not increase text coverage significantly. Moreover, many unpleasant ambiguities would be introduced which can be avoided in a modular structure.

The METAL lexicon is organized as follows: there are modules for function words (FW) like prepositions, determiners and conjunctions, for general vocabulary (GV), for common technical vocabulary (CTV) and common social/administrative vocabulary (CSV) organized in a tiered hierarchy.

From the next level down, all users can define and structure their own modules and tailor them to their specific application. For in-house use in Siemens, there are for example modules like Data Processing (DP) with submodules Software (SW), Hardware (HW) etc. Furthermore, it is possible to define transfers on the basis of a specific customer, a specific product or a specific target country. Thus a text translated into British English will show "pavement" instead of "sidewalk" for the USA, and a text intended for Spain will automatically have "ordenador" instead of the Colombian "computadora". The METAL lexicon structure can be visualized like this (simplified):

METAL Lexicon Structure



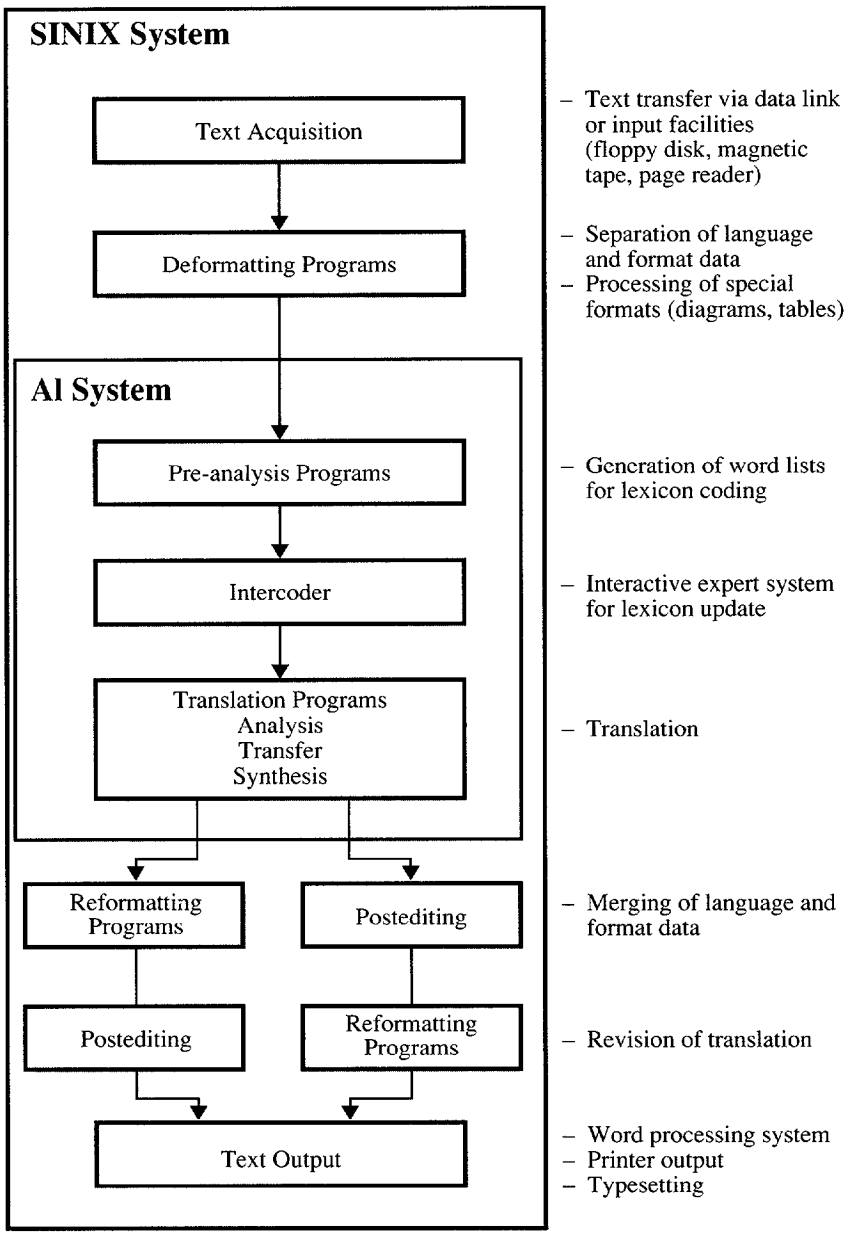
Before a translation run is started, the modules appropriate to the subject area of the text are defined. If the syntactic and semantic criteria for the selection of a lexicon entry are met and if there are several candidates for transfer, then the one tagged for the subject area of the text or tagged for the hierarchically closest module is chosen. In this way the highest priority is assigned to subject-specific transfers, *e.g.* if in an engineering text German "Mutter" occurs without grammatical context which could be used for disambiguation, it will be translated as "nut" rather than as the general language "mother".

External users update their own lexicons with the aid of the so-called INTERCODER, an integrated expert system. It guesses at the morphological and syntactic behavior of new lexicon entries and proposes the necessary coding; the missing pieces of information are inferred from a set of rules and partial information already contained in the lexicon. The INTERCODER has proven its usefulness in reducing coding time by a factor of ten.

Even though the grammar rules are not accessible to a user, the transfer lexicon permits significant syntactic transformations. Subjects can be turned into direct objects, indirect objects into subjects ("*Das Buch gefällt mir*" — "I like the book"), prepositional phrases can be mapped to other structures, and nodes can be added or deleted ("*einen Befehl erteilen*" — "to order"). Some translators with a background in linguistics make use of a system feature which permits the graphic representation of sentence analysis trees and the grammatical and lexical information attached to the various tree nodes. It is a powerful aid in the diagnosis of faulty analyses or lexicon coding errors. However, in a productive system, great care has to be taken in the design of the user interface so as not to overburden a translator with linguistic detail.

4. TEXT PROCESSING ENVIRONMENT

An operative machine translation system needs to do more than simply translate individual sentences entered from the keyboard. Most of the texts which have to be translated quickly and are of great volume such as *e.g.* technical documentation are heavily formatted. In some texts more than half of the characters on a page may be non-translatable material, notably flow charts, diagrams, tables and various control characters for format and layout. It would be highly uneconomical to manually extract the text portions to be translated and afterwards manually re-input them. That would not only be expensive but it would also invite errors in the additional reformatting tasks. Therefore, METAL has been integrated into a chain of processes, from text acquisition via automatic deformatting and translation to automatic reformatting procedures. A translation run usually goes through the following steps:


SINIX System
Text Acquisition
Deformatting Programs
AI System
Pre-analysis Programs
Intercoder
Translation Programs
Analysis
Transfer
Synthesis
Reformatting Programs
Postediting
Postediting
Reformatting Programs
Text Output
– Text transfer via data link or input facilities (floppy disk, magnetic tape, page reader)
– Separation of language and format data
– Processing of special formats (diagrams, tables)
– Generation of word lists for lexicon coding
– Interactive expert system for lexicon update
– Translation
– Merging of language and format data
– Revision of translation
– Word processing system
– Printer output
– Typesetting

A source document is usually received in machine-readable form, by file transfer, floppy disk exchange or from a font reader (an optical scanner with an additional software package which converts the pixels into "understandable" characters). Several programs running on the translator work station check the pages for tables, graphs etc. and mark them. They identify the text portions to be translated and generate a template of the page. The individual translation units, usually sentences but in the case of headlines or table entries also single words or phrases, are automaticallly recognized, numbered consecutively and extracted from the page template. They are written into a text file and transferred to the LISP server for translation.

After translation, the file containing the target language text units is returned to the translator station for post-editing. Here, the translators can choose whether they want to postedit an interlinear version which groups single source language/target language units sentence by sentence, or work on two windows with source and target text, or whether they prefer a target language output that has already been reformatted. In the former cases, the posteditors would start the reformatting program after having made their corrections. At the end, the target language text is available with all the formatting information and with the same layout as the original.

Unfortunately, the heterogeneity of desk top publishing systems and editors and the lack of standards do not permit the automatic treatment of random formats, not to mention the excessively creative use of multiple graphics layers etc. However, conversion packages for most of the popular editors such as WORD, INTERLEAF or VIEWPOINT have been implemented. Other editors are channeled via the European standard ODA/ODIF interface.

Before a text is translated, it is advisable to run a comparison of text and system lexicon. As linguistic processing is based not only on grammar rules but also on information contained in the lexical entries, sentences in which several words are unknown to the system are difficult to analyze, and the translation is likely to be inferior. Therefore missing words should be added to the lexicon. In METAL, the comparison of lexicon and text produces several files. One is a list of unknown words, each listed with its location and context so that transfers are more easily found. This list will actually also show faulty orthography so that the program can be used as a spelling checker as well. The second output is a list of compound words which were not found in the system lexicon but for which a translation is proposed on the basis of the individual components. Here the translator is called upon to make sure that the proposed translations are appropriate to the subject area. The third output is a text-based glossary, listing source term and proposed translation. This may be used to review subject area adequacy of the lexical entries. It is also useful if, in a large document, one portion is to be translated by the machine translation system, but the initial pages are written in a style which makes them unsuitable for machine translation. In such a case, the human translators can be given a glossary of only the terms contained in the pages to be translated so that they don't have to search through piles of subject-area listings. This ensures that consistent terminology is used throughout the whole document.

5. EVOLUTION BY USER REQUIREMENTS

The state of the art in computational linguistics does not permit the perfect translation of random texts. Therefore, if a text is translated not simply for the purpose of getting a rough idea of the content but with the aim of publication, postediting by a human translator will remain a necessity. Even if a system is tuned for specific subject areas there are still sufficient problems in linguistic analysis, especially if the meaning to be conveyed is hidden "between the lines". One should not attempt to measure the "correctness" of

machine translation in percentage points. Just as with human translation, there is not necessarily a single solution. The quality of a translation does not hinge on the quality of the machine translation system alone but is equally dependent on the quality of the source text. Inputting ungrammatical text or sequences of Kantian complexity will hardly ever produce satisfactory target versions. One also needs to consider the intended purpose of the text, expectations of the readers and even the stylistic preferences of the posteditor.

By now, several conferences have addressed the issue of evaluation of MT systems, *e.g.* a session at the 1991 MT Summit (Summit 1991). From a practical point of view, it seems that the quality of an MT system can only be judged in regard to the question whether translators working with the system have been able to increase their productivity and decrease turn-around time. One prerequisite of course is the willingness of translators to use the system in their daily work, and that presupposes not only a fairly high level of translation quality but ease of operation as well.

A new technology can easily be proven inadequate or even useless if the intended recipient refuses to accept such a system or insists on applying it in unsuitable ways.

Therefore the introduction of a machine translation system into an existing organization, be it a large industrial company or a translation bureau, requires several steps. First of all, users must have a clear picture of what can be expected from an MT system and what is beyond the scope of today's technology. Inappropriate use will only lead to frustration.

Once the conditions for the installation of a system have been assessed, *i.e.* translation volume, suitable types of text, hardware environment, and a positive decision has been reached, the organizational setup needs to be discussed. From which sources does the translator receive the original texts? Is there a possibility to influence the style of the original, to impose certain guidelines in regard to complexity of verbal expressions? And can the customers be persuaded to use standardized formatting and layout routines so that the tasks of deformatting and reformatting can be simplified?

Postediting machine output is different from revising a "human" translation. While the machine will make "severe" errors in syntax, a human translator will make fewer but random and less predictable errors. Usually, it takes a translator several weeks of practical work with an MT system to be able to anticipate the common errors perpetrated by the system and look for them.

In 1982 the METAL developers were sure they had a viable system: it could translate, it had a user interface, and it could deformat and reformat certain document types.

In retrospect it seems fortunate that METAL was tested by in-house translators who immediately came to the conclusion that the version was more of a hindrance than an aid to translation. Harsh criticism centered around both linguistic shortcomings and difficulties in day-to-day operation. The output of the deformatting programs was faulty if documents contained complex tables, and manual correction proved time-consuming. On the linguistic side, there were problems *e.g.* with the analysis of gapping constructions in passive clauses, ellipses, the improper generation of definite articles, with inappropriate word order of prepositional phrases etc. The list of bugs and untreated real-life phenomena seemed endless.

Though it was a disappointment to the group of developers who would have much preferred to move on to other, more "interesting" topics, step by step the problems were addressed. The scope of the grammar was enlarged, lexicons were extended and the user interface was reworked. Again, the system was tested in-house. The results were less than encouraging. A verdict of "Better than the last version, but still useless" does not generally increase a developer's motivation, and business controllers were taking a very critical look at the expensive project.

Still, the feedback from technical translators proved very valuable. In system design, developers may set priorities that do not coincide with the user's needs. To come up with a productively useful system, a cooperation between developers and users is of utmost importance. Thus, work on METAL did benefit from the seemingly harsh requirements laid down by the practicians. When the first systems were installed outside of the company, new problems surfaced. External customers generally operated within a different hardware environment, and numerous new modules had to be written to permit the integration into existing office structures. Questions concerning the integrity of lexicon modules and their maintenance had to be addressed, and the documentation needed to be upgraded.

Since then, experiences with more than two dozen METAL installations have been generally quite positive and can be summarized as follows:

Translators as well as their upper management need to understand that a machine translation system is not a substitute for a highly qualified translator but no more (and no less!) than a powerful tool.

For the productive use of METAL, an initial training period of one week has proved to be sufficient. A second week of training after a few months has turned out to be beneficial as it can answer questions which have arisen during the actual application. After that, consulting on a case by case basis seems adequate. It may sound strange, but during the first few months of operation, the translators' productivity will actually decrease. There is the initial overhead of bringing the lexicon up to a level where it covers most of the specific texts to be handled. Also, translators have to get used to the different work technique of postediting machine-translated texts and acquire skills in system administration and lexicon building. For a user report see (Little 1990). After this initial learning phase, which may vary from a few weeks to more than a year, users have reported considerable gains in productivity and a marked reduction in turn-around time. It appears that under favorable conditions a productivity gain by a factor of 2 to 3 is a realistic goal. In addition to the benefits derived from increased productivity, the consistency of terminology throughout all documents has been viewed as a qualitative improvement of the target text which could not have been achieved with conventional "human" translation.

By now, a METAL user group has come into existence. One of their many functions is to set priorities among the many individual requirements. The input by the user group has led to many improvements within the overall system. Coverage of the grammar was extended, the deformatting and reformatting programs were redesigned, and errors in the general system lexicon were corrected. Requirements for the exchange of lexicon modules among METAL users led to the definition of a lexicon interface structure. The need by some of the translators for a more flexible treatment of specific phenomena like constants or acronym + noun constructions resulted in the introduction of switches which allow a user to choose between the system's default solution and a personal override. For example, some may opt for "system UNIX", others for "the UNIX system". Such decisions can be set in a parameter file.

Similarly, certain sentences or phrases may need to be translated in non-standard ways. The direct German translation of the English form of address "Dear Sirs" would be "Sehr geehrte Herren". However, in business correspondence the phrase "Sehr geehrte Damen und Herren" is more appropriate. Therefore, the METAL Pattern Matcher was developed. This tool marks certain phrases of the source text as 'not to be translated' and replaces them with a predefined translation. Again it is due to the user group's input that it was designed. The same holds true for the implementation of additional conversion packages to and from different desk top publishing systems.

Constructive criticism of the training courses and requests for additional information caused an overhaul of the whole training program. A hotline was installed to help translators with a sudden problem, and an error message data base now permits everyone to trace the status of development.

Natural language processing, and especially machine translation, addresses highly complex problems. Any preconceived idea about potential requirements implemented in a vacuum is doomed to failure. So, developers and users must work together in the definition of system evolvement. To be sure, not all requests can be implemented, perhaps because the effort would be disproportionately high vs the benefit, or perhaps an extension of grammar coverage would require the recoding of masses of lexical entries. But as a rule, success in natural language processing can only be achieved on the basis of teamwork: user-driven development.

BIBLIOGRAPHY

ALPAC (1966): Automatic Language Processing Committee, National Academy of Sciences, *Language and Machines: Computers in Translation and Linguistics*, US National Research Council.

LITTLE, P. (1990): "METAL — Machine Translation in Practice", C. Picken (Ed.), *Translating and the Computer II*, London, Aslib.

SUMMIT (1991): *Proceedings of MT Summit III*, Washington, D.C., pp. 141-146.

THURMAIR, G. (1990a): "METAL, Computer Integrated Translation", J. McNaught (Ed.), *Proceedings of the SALT Workshop 1990*, Manchester.

THURMAIR, G. (1990b): "Complex Lexical Transfer in METAL", *Proceedings of the 3rd International Conference on Theoretical and Methodological Issues in Machine Translation of Natural Language*, University of Texas at Austin.