

The Workstation Substrate of the Pangloss Project

Sergei Nirenburg

School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213 USA

ABSTRACT. Pangloss is a new knowledge-based machine translation project carried out jointly by the Center for Machine Translation at Carnegie Mellon University, Computing Research Laboratory of New Mexico State University and Information Sciences Institute of the University of Southern California. One of the distinguishing features of systems built in this project is that they uniformly aim at high-quality output and broad coverage while accepting that full automation under such conditions is not immediately attainable. As a result, Pangloss systems rely on human-computer interaction during processing. This interaction must be supported by appropriate computer environments. There are two major kinds of people for whom environments can and must be constructed — developers and end users. Developers need help with a variety of knowledge acquisition tasks, such as dictionary and grammar writing. End users (e.g., technical writers, translators or editors of machine translations) look for efficiency enhancements beyond the level of word processing support. There are two approaches to building interactive tools. A dedicated workstation can be developed for each of the required functionalities. Alternatively, workstations can be configured as sets of application routines attached to a universal user interlace. In this paper we describe a user environment for Pangloss. It can support several machine-aided human translation configurations and a human-aided machine translation configuration. It also supports knowledge acquisition for machine translation and other NLP systems.

1. Pangloss

At present, Pangloss is dealing with translations between English and Spanish. It is planned that Japanese and German to be added at a later stage. Inputs are expected to be uncontrolled texts in the financial domain (specifically, mergers and acquisitions). Among the aims and features of Pangloss are:

- Constant high quality of output with increasing automation of processing (as opposed to systems which are fully automated but lower-quality). Configuring working systems even at lower levels of automation — MAT systems. Commitment to gradual increase in the levels

of automation of the systems.

- Emphasis on the breadth of coverage of the systems under construction.
- The use of an *interlingua* instead of transfer rules or statistical cooccurrences in the main translation engine; a version of example-based translation system for one of the MAT modes.
- The central role of *world knowledge* in addition to knowledge about language.
- The use of a text meaning representation based on commonsense semantics and pragmatics.
- Ensuring portability across domains through building reusable tools and information repositories such as lexicons.
- Developing a translator's workstation environment a) to support the creation of useful machine-aided translation systems at the earlier stages of the project, while the various automatic processing modules are being developed and b) to support the knowledge acquisition tasks for MT.

The first prototype MT system in the Pangloss project has been put together after only six months. Naturally, it uses a number of pre-existing components, for instance, the Spanish analyzer from the ULTRA MT system developed at NMSU or the English generator from the ISI generator, Penman.

The architecture of this first Pangloss MT system is illustrated in Figure 1.

In this paper we describe the translator's workstation environment for Pangloss.

2. The Pangloss TWS

There are two approaches to building interactive NLP tools. A dedicated workstation can be developed for each of the required functionalities. Alternatively, workstations can be configured as sets of application routines attached to a universal user interface. The Pangloss Translation Workstation (TWS) has opted for the latter choice. In what follows we describe the tool functionalities that

PANGLOSS YEAR 1 SYSTEM

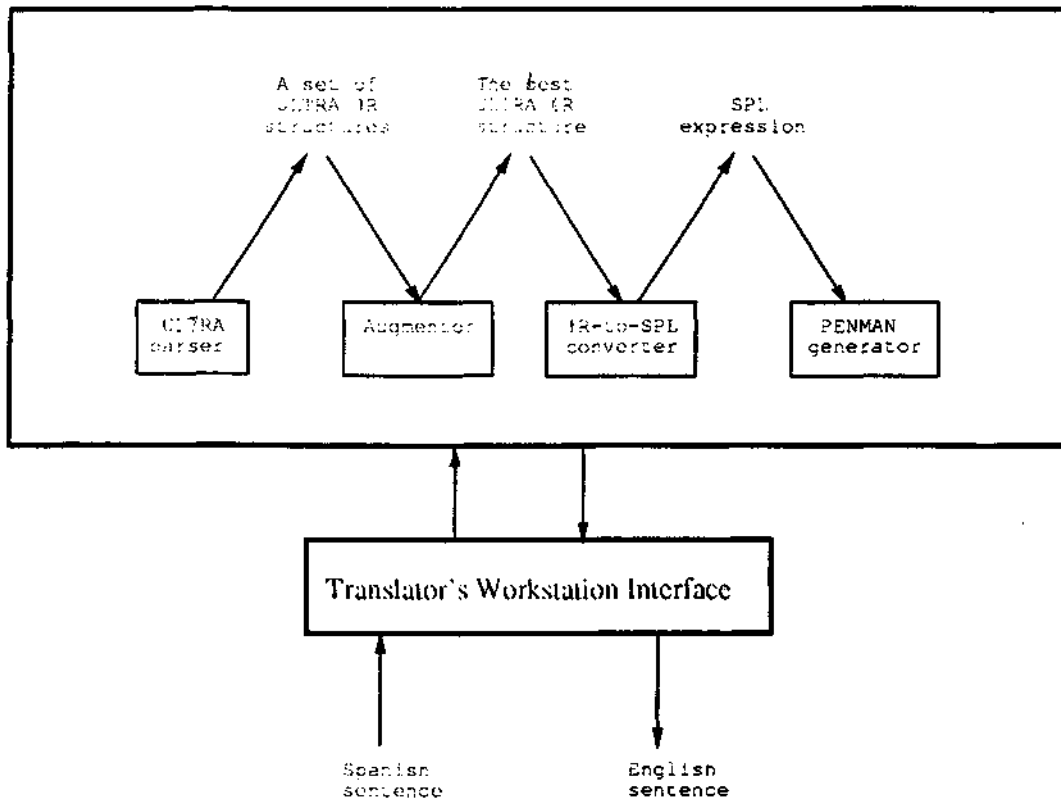


Figure 1: The Configuration of the First Pangloss MT System

have been developed and some workstation configurations in which they will be used. We believe that the TWS has potential both as a component of the Pangloss project and as a standalone tool.

TWS consists of a(n ever growing) number of application (functionality) modules which are integrated through the central user interface module. For reasons of uniformity, each of the applications uses the facilities of the user interface for display and input. Each module uses a standard window to interact with the user, and each window has standard menus which, among other functionalities, allow the user to invoke any other module. Each module also has *special* menus. Text editing can take place in any window (unless overtly prohibited). Text may be moved between windows with the help of a global kill-ring facility.

The concept of importing utilities is very important to our work. The user interface facilities of the kind we developed makes it easy to import applications with a minimum of effort and with a capability of making them "look and feel" similarly to the rest of the applications in the system. In our future work we intend to mix the development of utilities with importing and integrating existing applications, whenever that is desirable and feasible.

The major desiderata of user interfaces from the user's point of view are ease of use, predictable behavior and consistency. For the user interface builder it is important to follow strict principles of internal design, such as the following:

Uniformity The style of interaction with application modules must be standard, to make the system easy to use.

Portability The dependence of the system on a particular hardware or software configuration must be minimized.

Expandability The system must facilitate addition of new application modules, both locally developed and imported, without the need for a major reorganization or software modification. Special attention must, therefore, be paid to defining the communication strategy between the modules.

Speed. A system centered around a user interface must have a real-time response.

Guided by the above goals, we selected the following software substrate for our project. We use a Unix-based workstation with the X11 window system, the C-based X11 toolkit called MOTIF (OSF/Motif, 1991) and its CommonLisp interface called CLM (Babatz et al., 1991). MOTIF provides a high-level interface to X-11 by defining various types of *widgets*, e.g., windows, scroll bars, menus and buttons. MOTIF also allows multi-tasking. Finally, CLM uses a control-flow discipline known as *callbacks* which help to enforce good modularity.¹

A different approach to translator's workstation development has been taken by researchers at the Canadian Workplace Automation Research Center (e.g., Macklovitch, 1989), where it was decided to configure a system from commercially available components. While development costs can be somewhat lowered with this approach, it is difficult to attain the level of interpenetration of the various modules and thus further limitations are imposed on the possibilities of efficiency enhancements.

In what follows we describe the TWS functionalities (both existing and under development). The description is organized around the set of intended TWS configurations — the "base model," an enhanced machine-aided human translation environment, a human-aided machine translation environment and a set of tools for MT system developers.

The development of the TWS project is, naturally, a team effort. The development team has included, at various times, Ariel Cohen, Peter Cousseau, Bob Frederking, Dean Grannes, Chris McNeilly and Pete Shell. Many thanks to all of the above for help in preparing this report. All the remaining errors and inconsistencies are, naturally, mine.

3. TWS Configuration I: Basic Functionalities

The set of basic functionalities of TWS includes customizable text editing facilities, access to online reference sources (dictionaries, text archives, encyclopedias, parts catalogs, etc.), a facility

¹ Before settling on the above, we contemplated other substrates. For example, GARNET (Myers, 1990) is a high-level X-11 toolkit written entirely in CommonLisp, but at the time we started the project we found it to be too slow. SERPENT (Bass et al., 1990) is a faster system which allows the interactive specification of user interfaces. However, it has not been used widely yet and does not facilitate interaction with CommonLisp. Commercial systems were not used because we could find no system for which we could afford a version for a variety of hardware platforms.

for graphically acquiring, extending and maintaining proprietary glossaries and a package for manipulating large text archives (essentially, frequency counts a "key word in context" utility, etc.) Figure 2 shows a sample layout of three TWS windows: the TEXT EDITOR, the DICTIONARY INTERFACE, and the WORD FREQUENCY windows. Each of the corresponding modules is described in additional detail below.

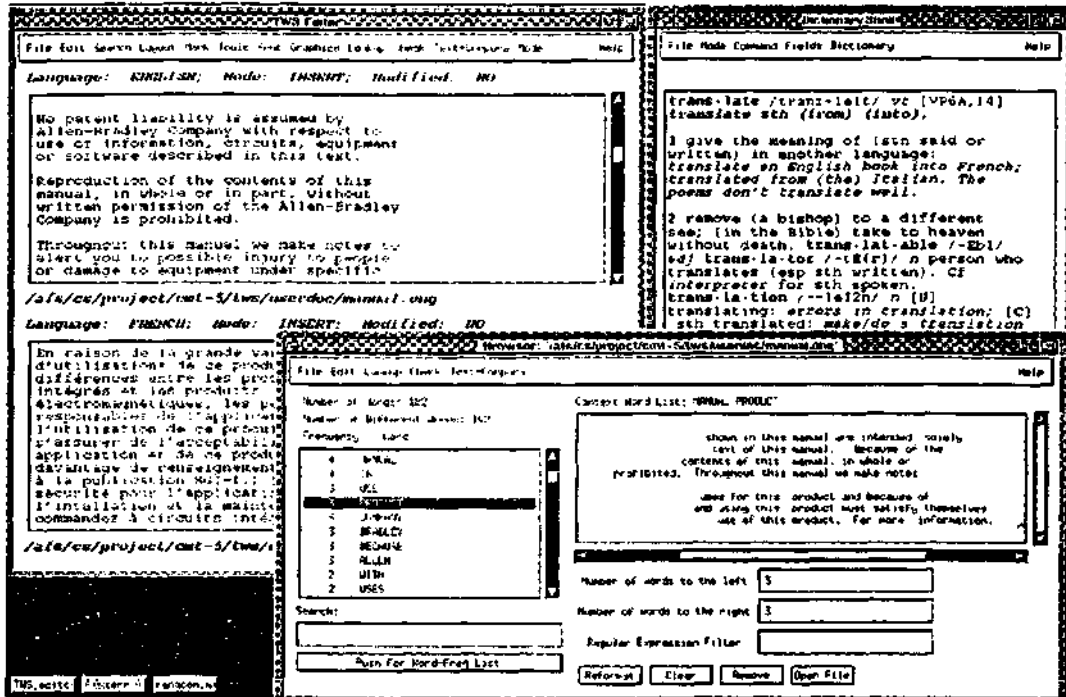


Figure 2: Sample TWS Windows. The text-editing, dictionary and open-text modules each use their own window.

In what follows we briefly describe these functionalities.

3.1. The Text Editor

The editor facility is one of the most basic functionalities for both end-user and developer workstation configurations. The TWS EDITOR must be flexible enough to provide the functionality required by the various applications, yet rigid enough to provide an identical appearance, regardless of the application using it.

The TWS editor is itself an application. Its central feature is its capability to emulate various existing word processors. This feature is important from the standpoint of acceptance of the

workstation environments by end users. Currently supported word processor emulation modes include WORD PERFECT™, VI and EMACS. The emulation modes were reverse-engineered. That is, no part of the design or actual code of the word processors that TWS emulates was used in TWS itself.

The TWS editor imitates the features (e.g., the menus and key bindings) of its client word processors as much as possible, in order to reconstruct their "look and feel." The TWS editor is able to read WYSIWYG (What You See Is What You Get) files, strip off any non-textual elements, such as file headers and formatting directives, allow the user to operate on the text, and then write the file preserving the original WYSIWYG format. Formatting directives, such as special typefaces, are still present in the text, and can be made visible at the request of the user.

A TWS editor interface is illustrated in Figure 3. In principle, the user can use any number of editor windows simultaneously.

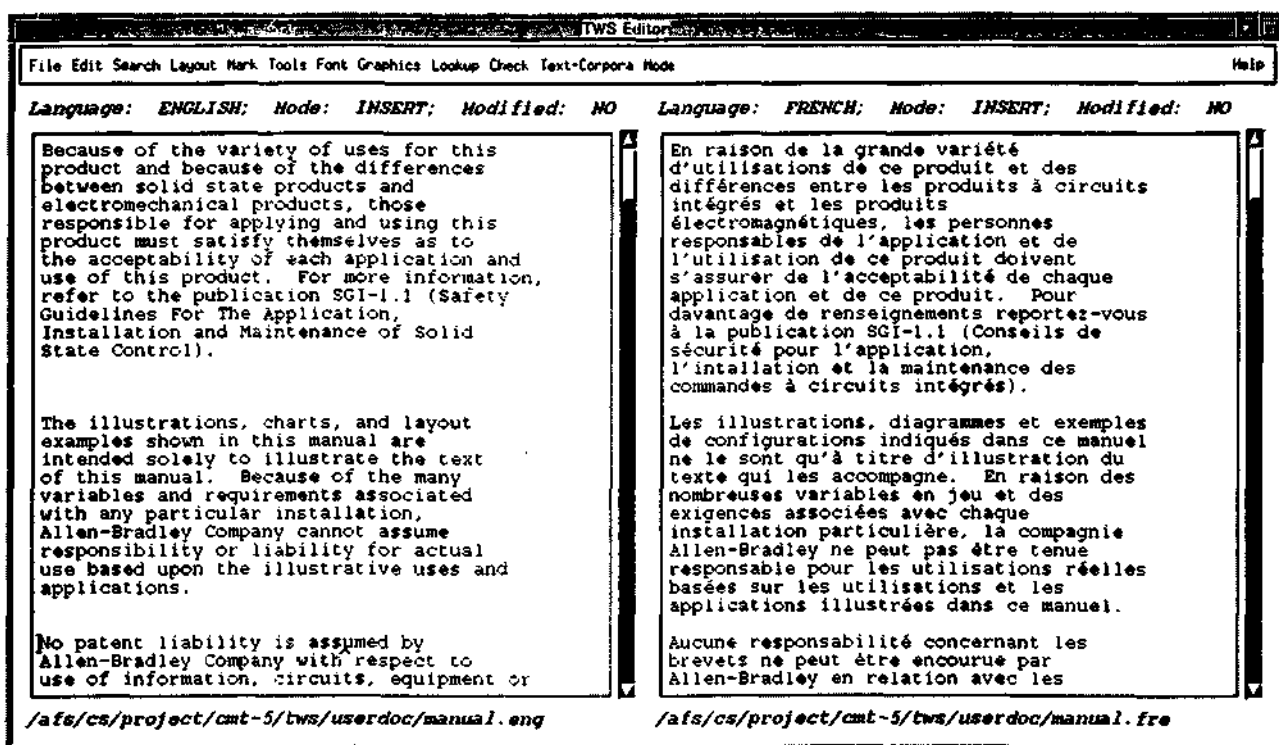


Figure 3: The TWS EDITOR Window. The source and target texts are aligned by paragraph.

If the translator so wishes, the TWS editor can be instructed to synchronize scrolling in several editor windows. When this feature is invoked, the paragraphs are automatically aligned, so that

extra blank lines are inserted after a paragraph in the text which is shorter than its corresponding paragraph in the other language.

In addition to the standard cursor movement and character insertion facilities, the TWS editor provides facilities standard in advanced word processors — text block commands, search-and-replace, insertion mode, special characters, multiple buffers, modification of key bindings, keyboard macros, "undo," etc.

3.2. The Dictionary Interface

TWS supports access to read-only materials (e.g., DICTIONARIES) and reference files which the user can modify (e.g., USER GLOSSARIES).

The DICTIONARY functionality provides the ability to access and display entries from machine readable dictionaries (both mono- and multilingual). After a word is selected, either by highlighting it in the editor window or typing it in a special dialog box, a dictionary window will appear with the definition of the word displayed (Figure 4).

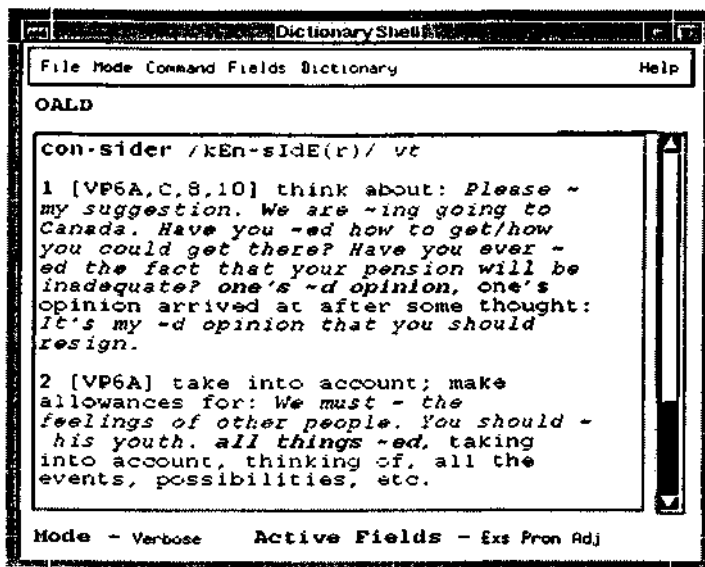


Figure 4: The TWS dictionary interface showing the entry for *consider* in the Oxford Advanced Learner's Dictionary of English.

The user can modify the display mode (from *terse* to *standard* to *verbose*) and call the next or the previous word through the menu. It is also easy to go to a different reference source.

The TWS includes, as a back-end functionality, a parser for an SGML-marked text which produces the necessary indices for flexible storage and retrieval of entries and entry parts. As SGML conventions allow for variability, introduction of new SGML-compatible reference materials will involve modifications to the parser. It is clear, though, that the availability of the parser will simplify the task of adding new reference materials to the system. For non-SGML materials special conversion procedures will have to be written.

The USER GLOSSARY functionality allows the user to create, modify, and maintain any number of personal reference materials. In the translation environment we expect that most of these will be glossaries containing definitions and/or translations for words, phrases, and/or idioms in the source text. One can retrieve a word or phrase from a glossary by highlighting it in the text editor and choosing an appropriate glossary in the menu.

The glossary is organized as an alphabetized list of entries. Each entry contains the following information:

- an English term;
- the corresponding terms in each of the required languages;
- a set of usage examples of all of the above terms, from Allen Bradley manuals;
- an explanatory description of the concept to which the term refers, in English; and
- a graphical representation (a line drawing) of the object represented by the term (if applicable and desirable).

Once accessed, the glossary will be displayed in a single-pane editor window, with the cursor positioned at the word or phrase with which it was called. To add a new term to the glossary, the user chooses the *Add a Term* menu option and the entered term will appear in the alphabetically correct position in the file and the cursor will be placed immediately after it, ready for the user to type in the translation or explanation.

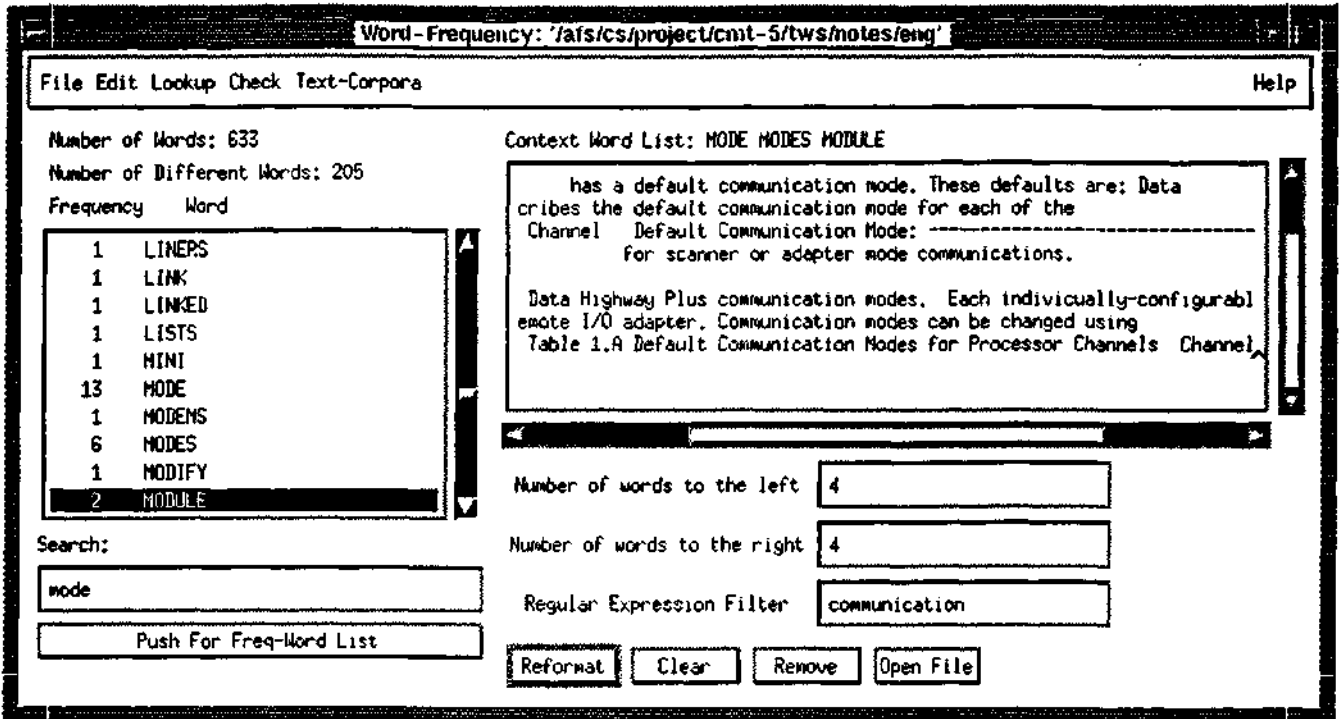


Figure 5: A Word-Frequency Window. The sub-window on the left shows frequency counts for each word in the chosen text, while the right sub-window shows the context of selected words.

3.3. Open Text Processing

This functionality is still quite simple. It produces frequency counts and key-word-in-context (KWIC) listings. For a practical translator, this application is helpful for retrieval, from an archive of past translations, of a section similar to the one which must be translated currently. As it is possible to get to the appropriate place in the file from a line in the KWIC window, complete paragraphs and even larger sections of text could be moved into the current translation. This facility has a powerful keystroke-saving potential. In a development environment, this facility is useful for corpus analysis and vocabulary determination prior to lexicon acquisition. Figure 5 illustrates the WORD-FREQUENCY application.

4. TWS Configuration II: Machine-Aided Human Translation

In developing tools for end users one must take into account an entire production process. In translation business, the process often starts before a document is submitted for translation. A lion's

share of material translated around the world is documentation accompanying various products — typically, operator's and repair manuals for various types of equipment. The manuals have to be written and then updated and revised for every new model or version of the equipment. This offers opportunities for efficiency enhancements with the help of TWS-MAHT.

One of the most important tasks in a multilingual document production environment is to minimize the effort required for introducing revisions into existing documents. After the text of a new manual is forwarded to a translation vendors and before it actually arrives at a translator's workstation, a revisor must detect all the differences between the new version of the manual and the old ones and make it as easy as possible for the translator to introduce the changes.

The revisor's workstation is equipped with a special package for text comparison. The differences between the old and the new texts come in many varieties. Paragraphs, sentences and words can be added, deleted or changed; layout can be changed; graphics and tables can be added, deleted or changed, paragraphs and entire pages can be moved to a different place in a document, etc. The revisor's workstation functionality facilitates such comparisons. The layout of the revisor's workstation screen is illustrated in Figure 6.

The chain of retrieval for the task illustrated in Figure 6 is presented in Figure 7

A major interactive efficiency enhancing facility for a *technical writer* is the ability to retrieve from a corporate archive paragraphs, sentences and phrases which include a given word or phrase. This module includes a user interface allowing a variety of display modes and a set of algorithms for search and retrieval of information from the archive. A technical writer will also benefit from a grammar and style filter which will automatically flag sentences and phrases which include nonstandard technical language or convoluted phrase structure (TWS does not at present include this functionality). Additional authoring workstation functionalities form an integral part of our approach to knowledge-based machine-aided translation.

When configured for a *translator*, TWS-MAHT includes all the above functionalities. In addition, the revision module is somewhat modified, so that it supports a version of the so-called "example-based" translation. In particular, there is no presumption that a source text is a new version of another known document. A database of translated materials is of central importance as

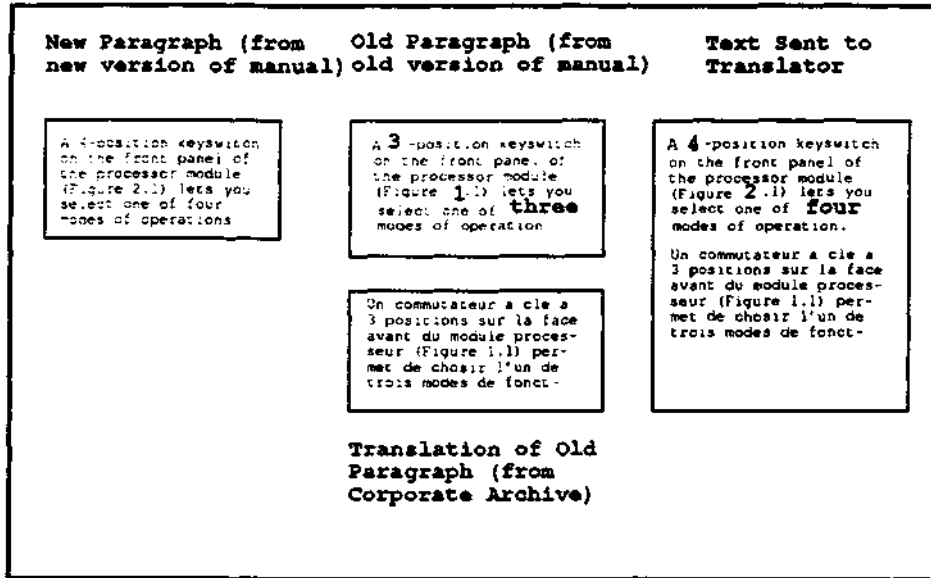
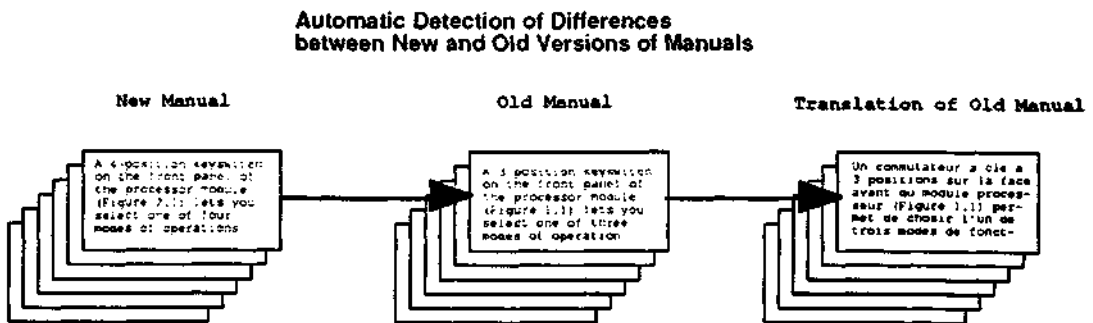


Figure 6: A sample display window of the revisor's workstation. The four panes on the screen show a paragraph in the new manual; a paragraph from the old manual, with differences from the text in the new manual highlighted; the corresponding part of a translation of the old manual; and a composite paragraph, with new source text on top and old target text below. The contents of the latter window will form a part of the text sent out to translators. The manipulation of various pieces of text can be done automatically, where feasible and manually by the revisor, where not.

Figure 7: A Document Retrieval Chain



A 4-position keyswitch on the front panel of the processor module (Figure 2.1) lets you select one of four modes of operations

Figure 8: A paragraph for translation.

CORPORATE DOCUMENT ARCHIVE

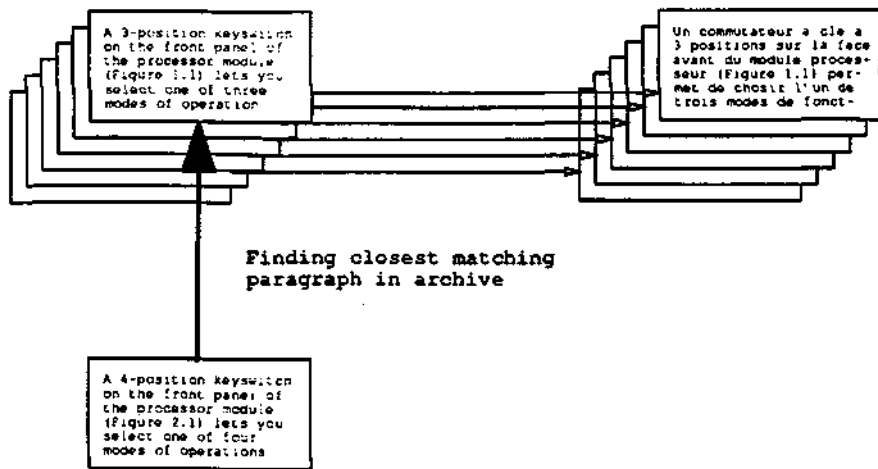


Figure 9: Finding Closest Match.

the knowledge source for this functionality. Figures 8 through 12 illustrate the process of operation of the example-based translation module. This module is presently under development. The details of our approach to EBT, including search and matching algorithms, will be reported elsewhere. A number of experiments with example-based translation have been reported (e.g., Brown et al., 1991, Sato and Nagao, 1990, Kugler et al., 1991).

5. TWS Configuration III: Human-Aided Machine Translation

The MAHT configuration of TWS described above can support postediting. The results of MT can be displayed in the target text window, and the posteditor can work in it using the same tools as a translator would. Our approach to human-aided machine translation is different and requires

CORPORATE DOCUMENT ARCHIVE

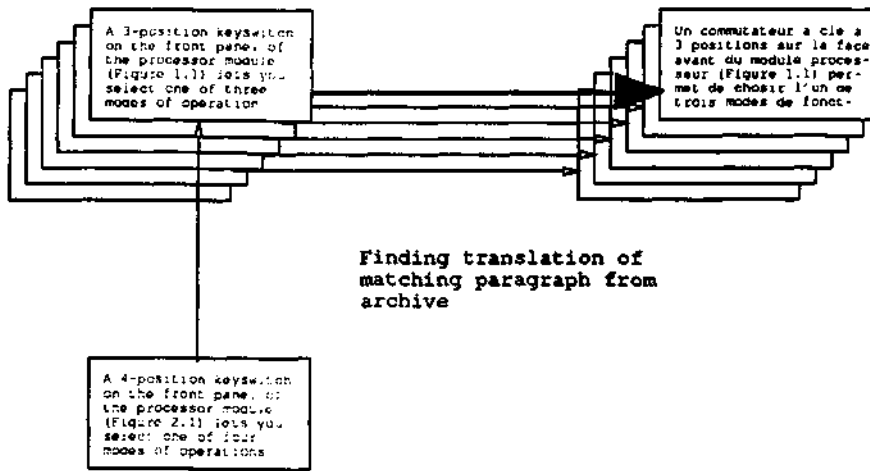


Figure 10: Retrieving Translation of Closest Match.

CORPORATE DOCUMENT ARCHIVE

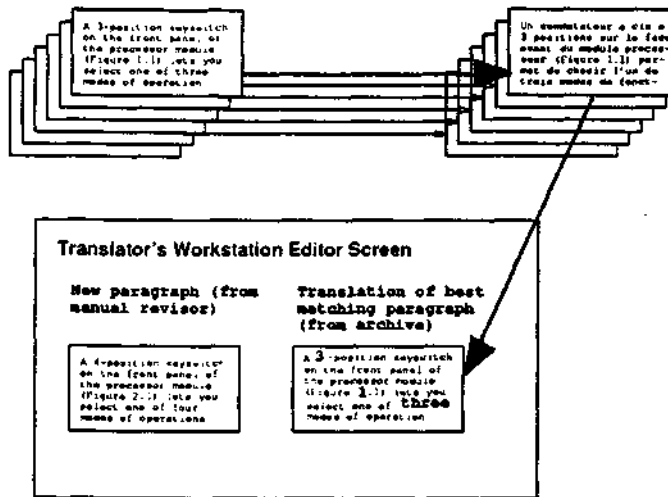


Figure 11: Retrieving Translation of Closest Match into Editor.

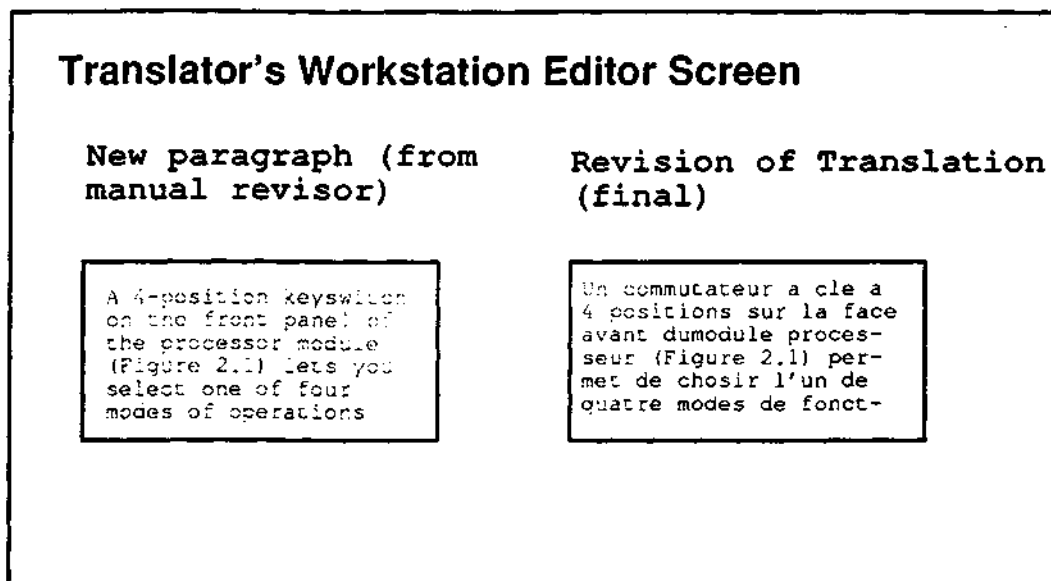


Figure 12: Final Result of Editing Old Translation.

a different TWS configuration. In this approach, while a computer tries to translate a document, a human translator monitors its progress and gives the system guidance when called upon to do so.

TWS-HAMT serves as a front-end interface to knowledge-based machine translation systems. The primary role of the human is a) to help the system make processing choices for which it lacks sufficient knowledge and b) to update the lexicons to cover unexpected vocabulary.

The human-computer interface that supports a subset of such functionalities has been implemented on a small scale in the MIND machine translation project at the RAND Corporation (Kay, 1973; the module was called "disambiguator"). A device of this sort was also discussed by Tomita (1986). A slightly larger interactive module was implemented in the KBMT-89 MT system, where it was known as the "augmentor" (Brown, 1991).

Figure 13 illustrates the way TWS-HAMT carries out human-computer in this system configuration. The dialog box with the disambiguation question is produced by the system and the user is supposed to click the appropriate box for the correct interlingua text to be produced. After the automatic synthesis stage, the translation will appear, sentence by sentence, in the right-hand side editor window, for the user to check.

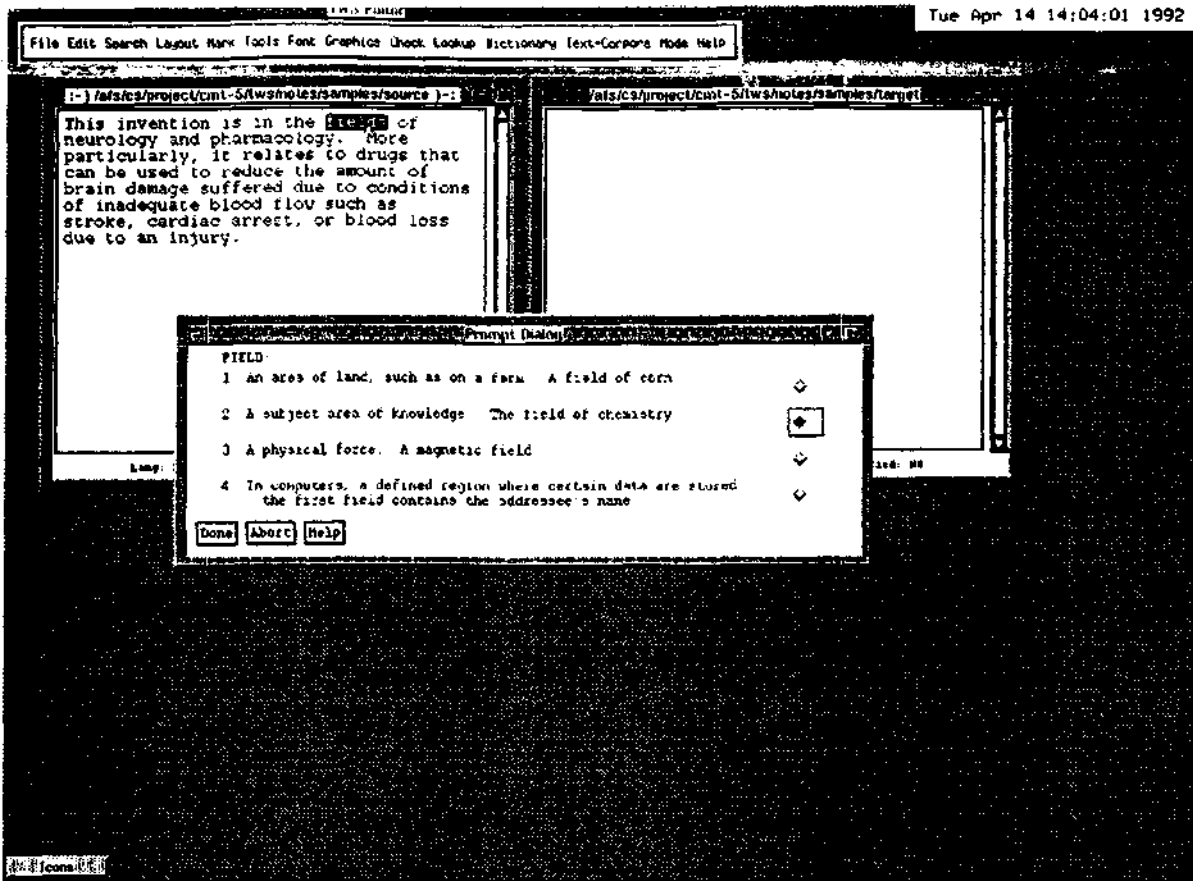


Figure 13: User Interaction in Human-Aided Machine Translation

6. Knowledge Acquisition Support

A number of interactive development environments have been reported (e.g., IRACQ (Ayuso et al., 1987), LUKE (Wroblewski and Rich, 1988) or ONTOS (Monarch, 1989)). All of them are meant to support a particular kind of NLP system. Thus, they trade generality and adaptability to new tasks for efficiency in solving the tasks of a given project.

6.1. Acquiring Textual Databases

The centerpiece of our approach to machine-aided human translation is the system's capability to suggest possible translations for fragments of input text which the operator can use, modify or reject.

The capability of producing suggested best-estimate translations for human judgment is predicated on the availability of a large number of previously translated texts in relevant sublanguages. Once such texts are obtained, they are organized in an internal archive which is used for a variety

ENGLISH

SPANISH

A 3-position keyswitch on the front panel of the processor module (Figure 1.1) lets you select one of three modes of operation.

Un conmutador de 3 posiciones, situado en el panel frontal del procesador, le permite seleccionar uno de los tres modos de operacion

INTERLINGUA

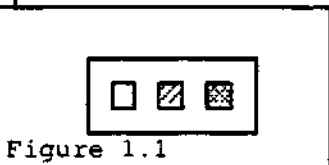
FRENCH

```
(*allow
(agent %user)
(theme
(*select
(agent %user)
(theme %mode-1)
```

Un commutateur a cle a 3 positions sur la face avant du module processeur (Figure 1.1) permet de choisir l'un de trois modes de fonct-

GRAPHICS

GERMAN



Ein dreistufiger Schalter des Prozessormoduls erlaubt es, eine von drei moeglichen Betriebsarten zu waehlen

Figure 14: The Structure of a Corporate Document Archive.

of search and match operations resulting in the display of the "best fit" translated sentence for each sentence of input. The structure of this archive is illustrated in Figure 14. The management system for this database is currently under development.

6.2. Acquiring Ontology: Ontos

A central knowledge source in knowledge-based MT systems is a model of the world, often called "ontology." Ontology acquisition is a very time-consuming process. An ontology acquisition tool, ONTOS, has been developed at CMU (see, e.g., Monarch and Nirenburg, 1988). A new version of the ONTOS graphical editor has been incorporated in TWS and is described in what follows.

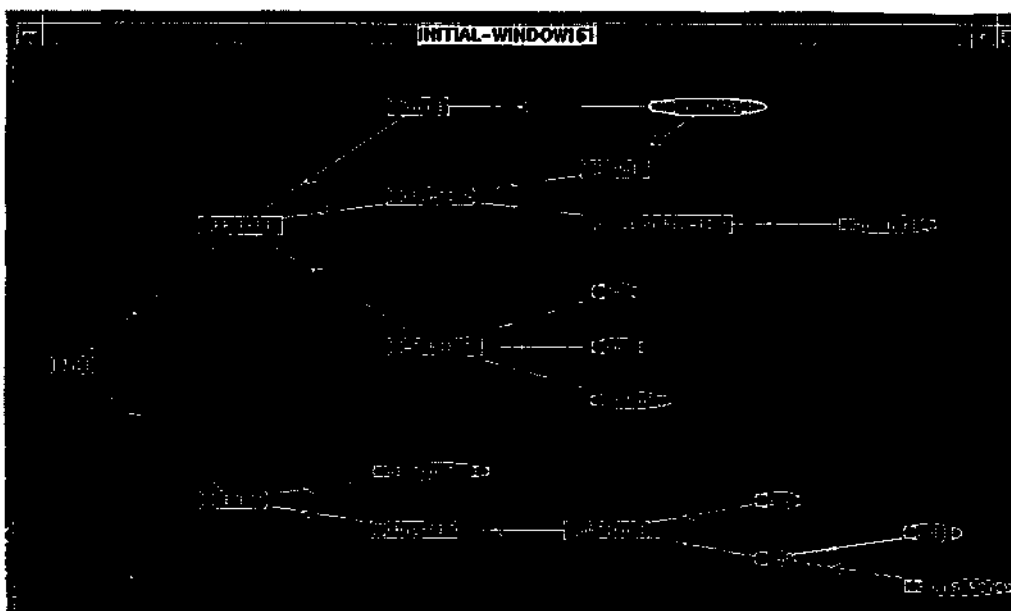


Figure 15: A sample ontology.

The new ONTOS GRAPHER can, in fact, be used as a general-purpose graphics tool for manipulating any kind of network representations. An important feature is the ability to handle full graphs, rather than trees. Graphical objects can be manipulated independently of the underlying conceptual entities. The FRAMEGRAPHICS package (Joseph, 1990) was used in implementing the grapher. Two types of graphic objects are supported — nodes and links. Nodes represent conceptual units while links highlight connections among them. The ONTOS grapher allows a large number of shapes and patterns for both nodes and links, in order to simplify recognition by the user.

Various browsing and acquisition actions can be performed by selecting objects, and then clicking on a menu item or pressing a key. The actions include creating, deleting, copying and updating nodes and links. The user can also edit the representations of conceptual objects corresponding to the various nodes and links.

Figure 15 illustrates the grapher interface. It displays a simple ONTOS ontology.

A sample use of the Browser menu is illustrated in figure 16. This menu allows the user quickly to browse through the knowledge base without the need to create a large number of nodes.

The TWS GRAPHER compares favorably with other graphers, notably the AGGREGRAPHS (see Dannenberg et al., 1991) which is part of the GARNET system and the ISI GRAPHER (see Robins, 1988). Unlike AGGREGRAPHS, which can only *display* a knowledge base, our grapher lets the user edit it. The ISI GRAPHER supports only a single type of link, while the TWS grapher can support an unlimited number of links.

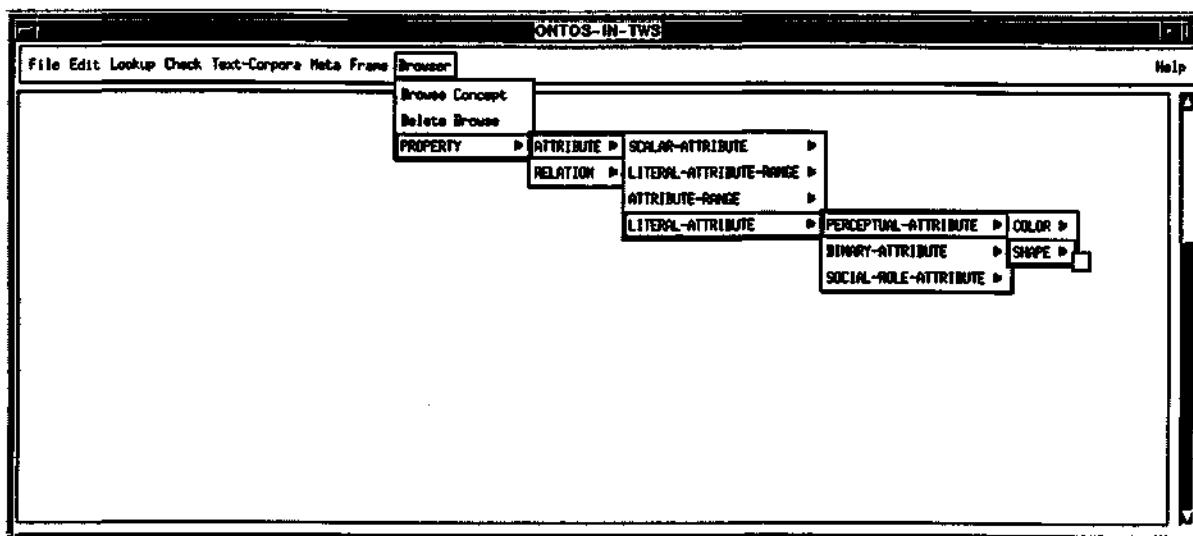


Figure 16: The Browser. Nested menus provide easy access to the Ontology.

Possible complement verb type(s)

Finite
Nonfinite
Present Participle
Past Participle

Is the auxiliary -to- used?

Always

Sometimes

Never

Embedded clause complement type:

COMP

XCOMP

Done

Abort

Help

Figure 18: Determining Syntactic Behavior of English Verbs

SMELL				
1				
Grammatical Category	Syntactic Features			
<CAT (VALUE V)>	<SYN>			
User Information	Syntactic Structure			
<STUFF (DEFN use olfactory sense voluntarily) (EXAMPLES smell this liquid...what do you think it is?) (TIME-STAMP Ingrid Feb 12 90)>	<SYN-STRUC (LOCAL <<ROOT %VAR0> (SUBJ <<ROOT %VAR1> (CAT N))>> <OBJ <<ROOT %VAR2 OPTIONAL> (CAT N))>>>>			
Orthographic Info	Semantic Structure			
	<SEM (LEX-NOP <VOLUNTARY-OLFACTORY-EVENT <AGENT (VALUE %VAR1)> (SEM <MOR# %ANIMAL #BIRD #REPTILE #AMPHIBIAN))>>			
Phonological Info	Lexical Relations			
Morphological Info	Pragmatic Info			
<MORPH (TRREG (%V+PAST# SMELL OPTIONAL) (%V+PAST-PART# SMELL OPTIONAL))>				
Quit	Load Entry	Modify Entry	Next Sense	Previous Sense

Figure 19: The Snapshot of a Nascent English Lexicon Entry

Select one of the following
verb-types

Promise
Persuade
Seem
Try
Believe
Other

Done Abort Help

Figure 17: Determining Subcategorization Classes for English Verbs

6.3. Lexicon Acquisition

Acquisition of machine-tractable lexicons is as important for machine translation development as acquisition of ontologies. Within TWS, we have developed a module called LAI, the Lexicon Acquisition Interface to support this task. LAI consists essentially of a set of user questionnaires organized in decision trees and presented to the user through TWS dialog boxes and an entry browser window. The questionnaires "walk" the user through the list of morphological, syntactic, semantic and other properties required for a particular lexicon entry format. The system records the information obtained from the user in a specified formalism and displays the nascent lexicon entry in a browser window. Sample questionnaire dialog boxes and a browser window are illustrated in Figures 17, 18 and 19. This type of functionality has been implemented and used in a number of NLP systems. See Knight, 1991, for a survey.

At present, our LAI supports the acquisition of lexicons only for English and for one particular lexicon format. In order to facilitate the creation of LAI questionnaires and browser windows for other languages and other kinds of lexicons, a special interface-building interface (IBI) has been implemented, which allows for interactive specification of menus and window widget sets (such as the dialog boxes and the browser window).

7. Conclusion

Translator workstations are at present in their infancy, but there is hope that in five to ten years their general availability on popular hardware platforms will forever change the nature of translation industry.

A user environment, such as a TWS, can never be considered completed. There are always new and exciting functionalities that can be added to any system to make it smarter, faster, more flexible, customizable or convenient in use. We have a long list of improvements to the system along the above lines. We also plan to have our TWS undergo more extensive practical tests by translators, which will give us insights into development priorities.

References

- Ayuso, D., V. Shaked and R. Weischedel. 1987. An Environment for Acquiring Semantic Information. Proceedings of Annual Meeting of ACL. Stanford, CA.
- Babatz, R., A. Backer, C. Beilke, T. Berlage, and M. Spence. 1991. CLM — A Language Binding for Common Lisp and OSF/Motif User Guide and Reference Manual Version 2.0, Technical report, German National Research Center for Computer Science.
- Bass, L., B. Clapper, E. Hardy, R. Kazman and R. Seacord. 1990. Serpent: A User Interface Management System. Proceedings of the Winter 1990 USENIX Conference.
- Brown, R. 1991. Augmentation. In: K. Goodman and S. Nirenburg (eds.) **KBMT-89: A Case Study in Knowledge-Based Machine Translation**. San Mateo, CA: Morgan Kaufmann.
- Brown, P., J. Cocke, J. della Pietra, S. Della Pietra, F. Jelinek, J. Lafferty, R. Mercer and P. Roossin. 1990. A Statistical Approach to Machine Translation. *Computational Linguistics*, 16, 79-85.
- Dannenberg, R. B., Marchal, P., Kosbie, D., Mickish, A. and Loyall, A. B. 1991. Aggregadgets & Aggrelists Reference Manual. Technical Report, Department of Computer Science, Carnegie Mellon University.
- Joseph, R. L. FRAMEGRAPHICS 1.0: A Frame Based Graphic System. 1990. Unpublished report, Department of Computer Science, Carnegie Mellon University.
- Kay, M. 1973. The MIND System. In: R. Rustin (ed.) **Natural Language Processing**. New York: Algorithmics Press, 155-188.
- Knight, K. 1991. Integrating Knowledge Acquisition and Language Acquisition. CMU PhD Thesis.
- Kugler, M., G. Heyer, R. Kesse, B von Kleist-Retzow and G. Winkelmann. 1992. The Translator's Workbench: An Environment for Multi-Lingual Text Processing and Translation. In: S. Nirenburg (ed.) **Progress in Machine Translation**. Amsterdam: IOS Publications, pp. 185-189.
- Macklovitch, E. 1989. An Off-the-Shelf Workstation for Translators. Proceedings of the 30th

American Translators Conference, Washington, D.C.

Moen, S. 1990. Drawing Dynamic Trees. *IEEE Software*, July.

Monarch, I. 1989. ONTOS: Reference Manual, Technical report, Center for Machine Translation, Carnegie-Mellon University.

Myers, B., Giuse, D., Dannenberg, R., Zanden, B., Kosbie, D., Pervin, E., Mickish, A., Marchal, P., 1990. Garnet: Comprehensive Support for Graphical, Highly-Interactive User Interfaces. *IEEE Computer*, Vol. 23, number 11.

Nirenburg, S. and C. Defrise. 1992. Application-Oriented Computational Semantics. To appear in Johnson, R. and Rosner M. (eds.), **Computational Linguistics and Formal Semantics**. Cambridge University Press.

Nyberg, E. H. The FrameKit User's Guide. 1988. Technical Report, Center for Machine Translation, Carnegie Mellon University.

OSF/Motif Programmer's Guide, Revision 1.1, Open Software Foundation, Prentice Hall, 1991.

Robins, G. The ISI Grapher Manual. 1988. Technical Report, ISI, University of Southern California.

Sato, S. and M. Nagao. 1990. Toward Memory-Based Translation. Proceedings of Coling-90. Helsinki. August.

Tomita, M. 1986. Sentence Disambiguation by Asking. *Computers and Translation*, 1:39-51.

Wroblewski, D.A. and E.A. Rich. 1988. Luke: An Experiment in the Early Integration of Natural Language Processing. Proceedings of Second Conference on Applied Natural Language Processing. Austin, TX.