# Interactive enconversion by means of the ETAP-3 system

I. Boguslavsky, L. Iomdin, V. Sizov

Institute for Information Transmission Problems,

Russian Academy of Sciences

Abstract.

A module for enconversion of NL texts into Universal networking Language (UNL) graphs is considered. This module is designed for the system of multi-lingual communication in the Internet that is being developed by research centers of about 15 countries under the aegis of UN. The enconversion of NL texts into UNL is carried out by means of a multi-functional linguistic processor ETAP-3, developed in the Computational linguistics laboratory of the Institute for Information Transmission Problems of the Russian Academy of Sciences. One of the major problems in the automatic text analysis is high degree of ambiguity of linguistic units. The resolution of this ambiguity (morphological, syntactic, lexical, translational) is partly ensured by the linguistic knowledge base of ETAP-3, but complete algorithmic solution of this problem is unfeasible. We describe an interactive system that helps resolve difficult cases of linguistic ambiguity by means of a dialogue with the human.

## 1. Introductory Remarks

ETAP-3 is a multipurpose NLP environment that was conceived in the 1980s and has been worked out in the Institute for Information Transmission Problems, Russian Academy of Sciences (Apresjan *et al.* 1992a,b, Boguslavsky 1995). The theoretical foundation of ETAP-3 is the Meaning ⇔ Text linguistic model by Igor' Mel'čuk and the Integral Theory of Language by Jurij Apresjan. ETAP-3 is a non-commercial environment primarily oriented at linguistic research rather than creating a marketable software product. The main focus of the research carried out with ETAP-3 is computational modelling of natural languages. All NLP applications in ETAP-3 are largely based on a three-value logic and use an original formal language of linguistic descriptions, FORET.

## 2. Briefly on ETAP-3

The major NLP modules of ETAP-3 are as follows:

- Machine Translation System
- Natural Language Interface to SQL Type Databases
- System of Synonymous Paraphrasing of Sentences
- Syntactic Error Correction Tool
- Computer-Aided Language Learning Tool

- Tree Bank Workbench
- UNL Deconverter and Enconverter.

The following are the most important features of the whole ETAP-3 environment and its modules:

- Rule-Based Approach
- Stratificational Approach
- Transfer Approach
- Syntactic Dependencies
- Lexicalistic Approach
- Multiple Translation
- Maximum Reusabilty of Linguistic Resources

In the current version of ETAP-3, its modules that process NL sentences are strictly rule-based. ETAP-3 shares its stratificational feature with many other NLP systems. It is at the level of the normalized, or deep syntactic, structure that the transfer from the source to the target language takes place in MT. ETAP-3 makes use of syntactic dependency trees for sentence structure representation instead of constituent, or phrase, structure. The ETAP-3 system takes a lexicalistic stand in the sense that lexical data are considered as important as grammar information. A dictionary entry contains, in addition to the lemma name, information on syntactic and semantic features of the word, its subcategorization frame, a default translation, rules of various types, and values of lexical functions for which the lemma is the keyword. The word's **syntactic features** characterize its ability/non-ability to participate in specific syntactic constructions. A word can have several syntactic features selected from a total of more than 200 items. **Semantic features** are needed to check the semantic agreement between the words in a sentence. The **subcategorization frame** shows the surface marking of the word's arguments (in terms of case, prepositions, conjunctions, etc.). **Rules** are an essential part of the dictionary entry. All rules operating in ETAP-3 are distributed between the grammar and the dictionary. Grammar rules are more general and apply to large classes of words, whereas the rules listed or simply referred to in the dictionary are restricted in their scope and only apply to small classes of words or even individual words. This organization of the rules ensures self-tuning of the system to the processing of each particular sentence. In processing a sentence, only those dictionary rules are activated that are explicitly referred to in the dictionary entries of the words making up the sentence.

## 3. ETAP-3 and UNL

It would be out of place to present here the whole UNL system, its underlying philosophy, language design, and the current state of system development. We refer the readers first of all to the publications by the author of UNL Hiroshi Uchida and other data that can be found at the UNL official site *http://www.undl.org*. Our purpose is to describe the UNL module of ETAP-3, and, in particular, the UNL enconverter, i.e. the system that receives a natural language sentence at the input and produces a UNL graph at the output.

Since ETAP-3 is an advanced NLP system based on rich linguistic knowledge, it is natural to maximally re-use its linguistic knowledge base and the whole architecture of the system in this new application. Our approach (described in detail in Boguslavsky *et al.* 2000) is to build a bridge between UNL and one of the internal representations of ETAP, namely Normalized Syntactic Structure (NormSS), and in this way link UNL with all other levels of text representation, including the conventional orthographic form of the text.

The level of NormSS is best suited for establishing correspondence with UNL, as UNL expressions and NormSS show striking similarities. The most important of them are as follows:

1. Both UNL expressions and NormSSs occupy an intermediate position between the surface and the semantic levels of representation. They roughly correspond to the so-called deep-syntactic level. At this level the meaning of lexical items is not decomposed into the primitives, and the relations between lexical items are language independent.
2. The nodes of both UNL expressions and NormSSs are terminal elements (lexical items) and not syntactic categories.
3. The nodes carry additional characteristics (attributes).
4. The arcs of both structures are non-symmetrical dependencies.

At the same time, UNL expressions and NormSSs differ in several important respects:

1. All the nodes of NormSSs are lexical items, while a node of a UNL expression can be a sub-graph.

2. Nodes of a NormSS always correspond to one word sense, while UWs may either be broader or narrower than the corresponding English words.

3. A NormSS is the simplest of all connected graphs - a tree, while a UNL expression is a hyper-graph. Its arcs may form a loop and connect sub-graphs.

4. The relations between the nodes in a NormSS are purely syntactic and are not supposed to convey a meaning of their own, while the UNL relations denote semantic roles.

5. Attributes of a NormSS mostly correspond to grammatical elements, while UNL attributes often convey a meaning that is expressed both in English and in Russian by means of lexical items (e.g. modals).

6. A NormSS contains information on the word order, while a UNL expression does not say anything to this effect.

These differences and similarities make the task of establishing a bridge between UNL and NormSS far from trivial but feasible.

The architecture of the UNL module within ETAP-3 is represented in Fig. 1.

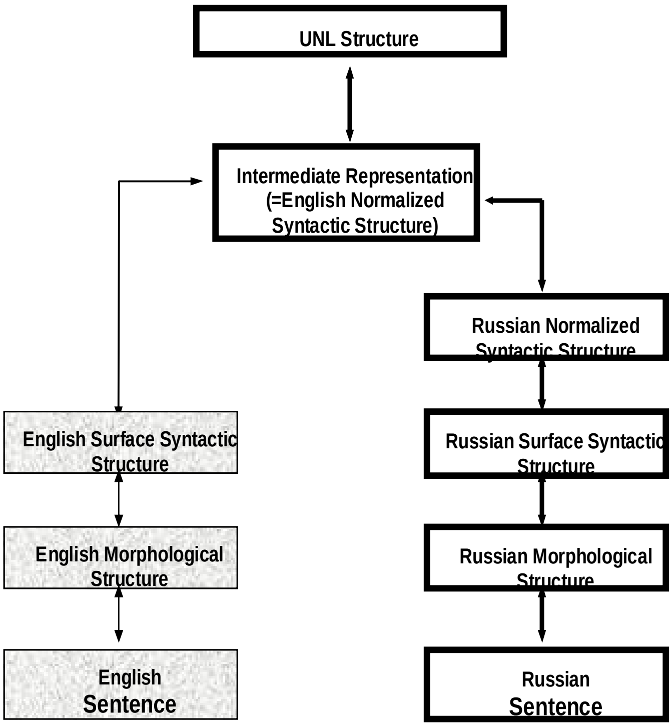


Fig. 1

As shown in Fig. 1, the interface between UNL and Russian is established at the level of the English NormSS. In the generation task, at this point, ETAP's English-to-Russian machine translation facility can be switched which carries through the phases of transfer and Russian generation. This architecture allows obtaining English generation for relatively cheap, as ETAP has a Russian-to-English mode of operation as well. Some experiments in this direction have been carried out which proved quite promising. Below, we will consider this scheme in the opposite direction – from the NL sentence to the UNL graph.

## 4. Interactive enconversion.

One of the most difficult problems in the automatic analysis of NL texts is the ambiguity of linguistic units. In ETAP-3, there is no single stage of processing expressly dedicated to disambiguation. The sentence is gradually disambiguated at different stages of processing on the basis of the restrictions imposed by the linguistic knowledge of the system. Examples:

1) lexical meanings with different grammatical properties: *We have tea in the garden* - *We were having tea in the garden,* but: *I have a pack of tea* - **I was having a pack of tea.*

2) lexical meanings with different syntactic properties: *Children grow fast* – *Children grow vegetables in the garden.*

3) grammatical meanings with different syntactic properties: *represented* – past participle (*countries represented in the UN discuss the resolution*) vs. past indefinite (*he represented his country*)

4) different syntactic structures: *the accusation of the minister* – 'the minister accused somebody' vs. 'somebody accused the minister' (the type of ambiguity not sufficiently accounted for in UNL!). In the sentence *The accusation of the minister by the parliament*, syntactic context provides a clue for disambiguation.

5) different translations of the same lexical meaning: *Wash your hands* – Rus. *Vymoj ruki,* but *Wash the tablecloth* – Rus. *Postiraj skatert'.*

All these and many other cases are successfully disambiguated by ETAP-3 thanks to the linguistic knowledge it is supplied with. However, in many cases linguistic knowledge of the system is insufficient for disambiguation. Of course, this may be due to the incompleteness of grammar and the dictionaries of the system. Should it be the case, this obstacle could in principle be overcome. In the long run, the linguistic knowledge base could be made virtually complete. Unfortunately, however, incompleteness of linguistic data is not the main obstacle. It is well-known that in very many cases the disambiguation performed by humans is not based on their linguistic knowledge alone. To a large extent, humans heavily employ their extra-linguistic competence in the outer world.

To give a simple example, suppose that a machine translation system has to translate a title from a recent article on the BBC site,

(1) *AIDS threatens economic collapse.*

It is very likely that the sentence will be wrongly understood as ‘AIDS poses a threat to economic collapse’ rather than ‘AIDS threatens (some countries) with economic collapse’, and, consequently, yield a wrong translation, for the simple reason that the system may lack the resources needed to distinguish the syntactic structure of this sentence from that of the sentence

(2) *AIDS threatens economic prosperity.*

Indeed, in order to make sure that the original sentence is parsed correctly, the system must know that the noun *collapse* instantiates the instrumental slot of the verb *to threaten* and not its object slot as in the second sentence. However, to provide adequate word lists for different slots of particular verbs is hardly possible because such lists will inevitably intersect in multiple ways; cf. ambiguous phrases like *threaten changes, threaten a revolution,* or *threaten the reduction.* On the other hand, any human who happens to read the BBC article will at once know what the original sentence (1) means.

It is therefore highly desirable that a rule-based NLP system be supplemented with an interactive tool that could, at certain pivotal points of language parsing, ask for human intervention and use this assistance to disambiguate some, or all of the ambiguous elements of the text being processed. Much work in this direction has already been done, first of all by the GETA group in the ARIANE environment (MIDDIM 1996).

It is exactly this interactive tool that we present in this paper. It should be stressed that the interactive tool will only be activated for the cases of ambiguity that cannot be resolved automatically and therefore require human intervention.

We will illustrate our approach with one English example. The sentence

**Примечание [L.1]:** Не понял, зачем нужна ссылка на русский язык.

(3) *We made the general remark that everything was fine*

is ambiguous between (at least) two interpretations:

(3a) ‘we made the general observation that everything was fine’

(3b) ‘we made the general say that everything was fine’

Obviously, meanings (3a) and (3b) are translated differently into other languages and should receive two different UNL-representations.

As mentioned above, one of the salient features of ETAP-3 is the fact that it has a MULTIPLE TRANSLATION option that can produce multiple (ideally, all possible) translations of each sentence. This option allows obtaining two different lexico-syntactic structures of sentence (3) and consequently two different translations into UNL. These structures, given in Fig. 2 and 3 below, conveniently visualize lexical and syntactic differences between (3a) and (3b). Note that syntactic links are represented as labeled dependency relations between the words of the sentence. The lexico-syntactic structure of a sentence is a tree in which every word (except for the root node) is connected by an incoming dependency relation with some other word. The root has no incoming relations but only outgoing ones.

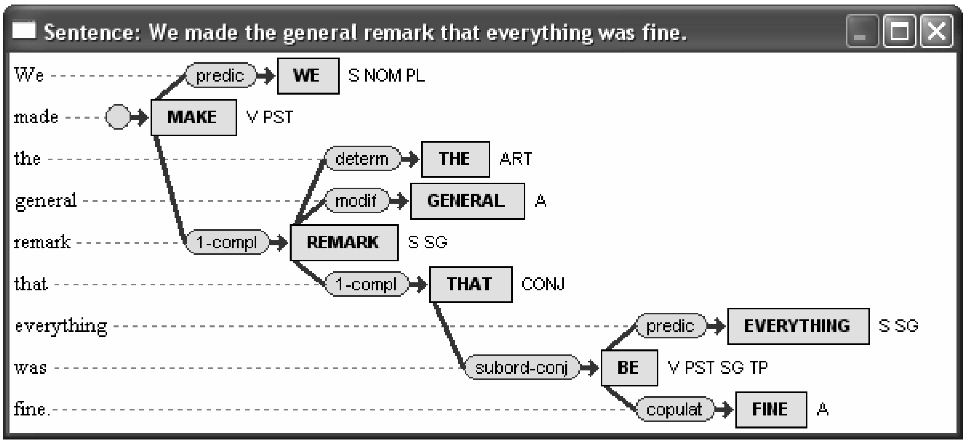


Fig. 2

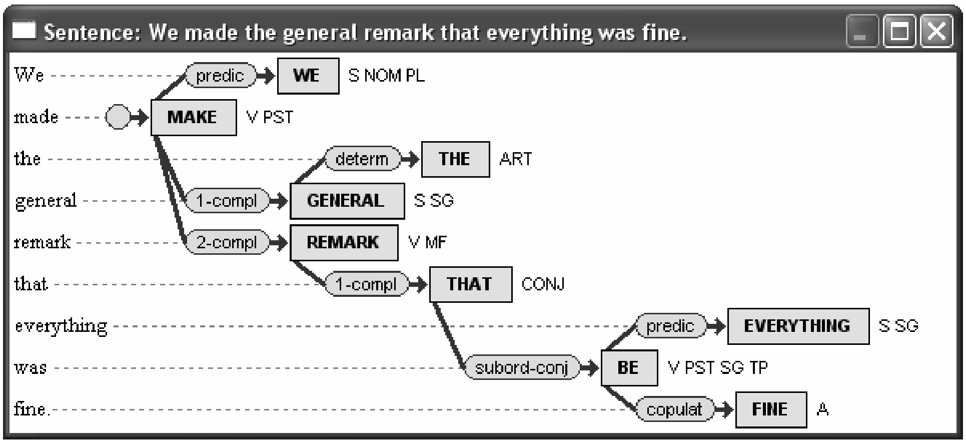
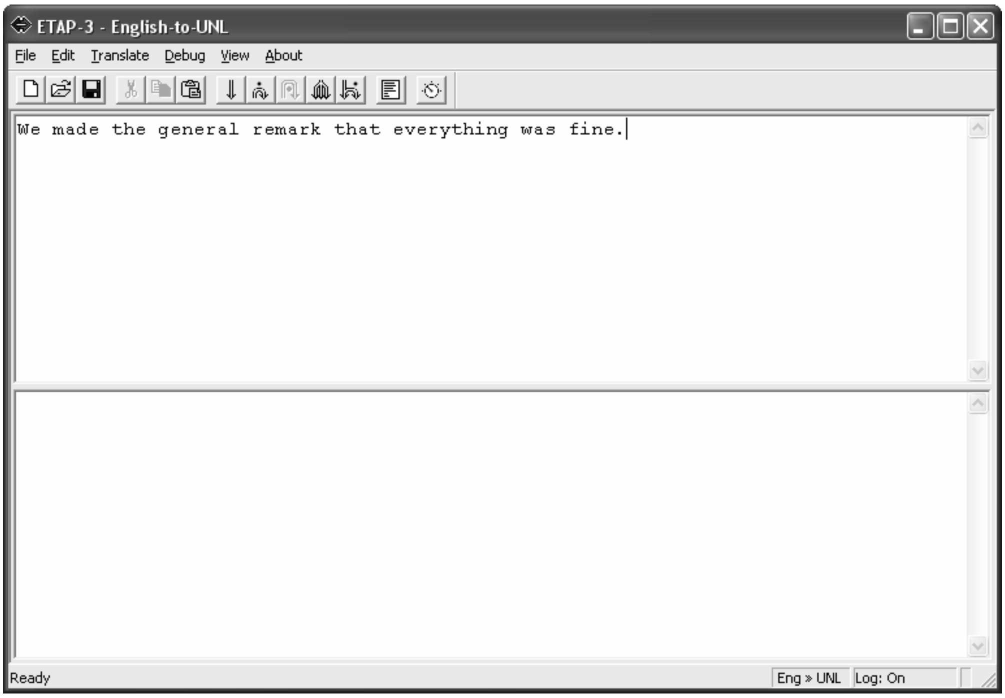


Fig. 3

In Fig. 2 *general* is an adjective (cf. label A to the right of the gray rectangle with the name of the word) and *remark* is a noun (cf. label S). In Fig. 3, *general* is a noun (cf. label S) and *remark* is a verb (cf. label V). Accordingly, in Fig. 2 the adjective *general* serves as a modifier of the noun *remark* (cf. label *modif* on the link that connects *general* to *remark*) and article *the* is also attached to *remark*. In Fig. 3 the noun *general* attaches article *the* and serves as the first complement of the verb *make* while the verb *remark* is its second complement (cf. labels *1-compl* and *2-compl* on the corresponding links). Besides that, there is a purely lexical ambiguity not shown in these structures. The word *fine* is ambiguous between an adjectival meaning (as in *fine weather*), a nominal one (as in *to pay a fine*) and a verbal one (as in *You will be fined*).

ETAP-3 is able to identify these ambiguities but in a general case cannot automatically decide which of the options is appropriate in a particular context. As mentioned above, this task can be reliably solved only in co-operation with the human. Let us switch on the Interactive disambiguation mode of ETAP-3 and participate in the dialogue proposed by the system. Fig. 4 shows the initial state of the English-to-UNL option with the English sentence input in the upper window.

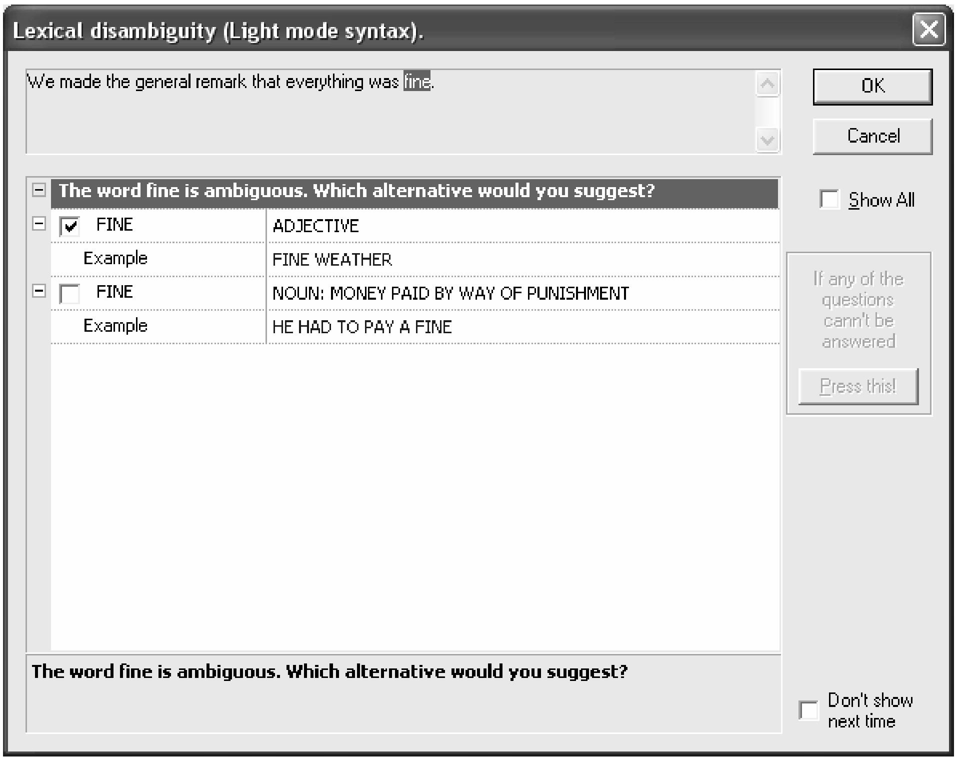


```
We made the general remark that everything was fine.
```

Fig. 4

The first occasion for the system to ask a question is the moment in the parsing when the root node of the structure should be selected. If a word that a system has chosen as a root node is ambiguous and the system cannot resolve this ambiguity, the user is asked for assistance. In our example, the only candidate for the root node (*made*) is unambiguous and no need for human intervention arises.

The word that activates a dialogue on the lexical ambiguity is *fine.* Of the three options mentioned above, the verbal one is incompatible with the syntactic context, but the other two can perfectly fit into it. Therefore, the first option is automatically rejected by the system and the other two are offered to the user. Fig. 5 shows the dialogue window that appears when the user is asked for assistance in lexical disambiguation. In the upper part of the dialogue window the sentence is reproduced with the word at issue highlighted. Below, the user is shown the options not yet resolved by the system, among which he/she is asked to choose. Each option is provided with a short but clear and informative comment and/or a simple example. What the user should do is identify and click the appropriate option. Comments and examples are formulated in such a way that no special linguistic knowledge is required to choose among the options.



Lexical disambiguity (Light mode syntax).

We made the general remark that everything was fine.

The word fine is ambiguous. Which alternative would you suggest?

| | |
|---|---|
| ☑ FINE | ADJECTIVE |
| Example | FINE WEATHER |
| □ FINE | NOUN: MONEY PAID BY WAY OF PUNISHMENT |
| Example | HE HAD TO PAY A FINE |

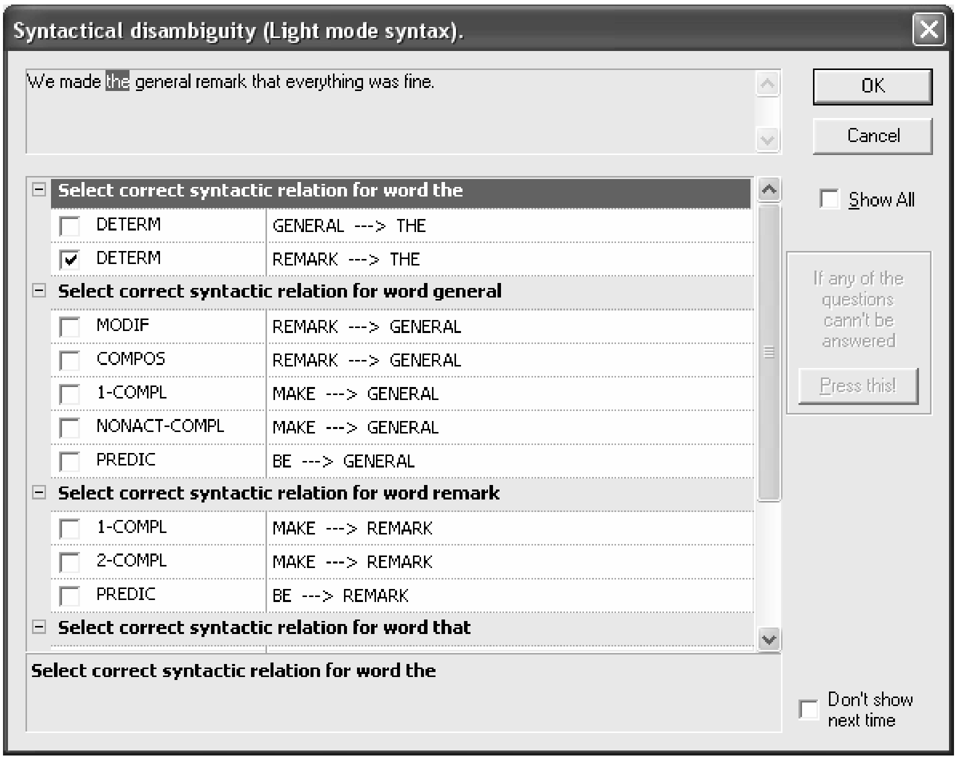
The word fine is ambiguous. Which alternative would you suggest?

Fig. 5

After having dealt with purely lexical ambiguity, the system passes to syntactic ambiguity or to complex cases when lexical and syntactic ambiguities come together. The syntactic ambiguity dialogue window represents words that have more than one alternative governors (while in a tree only one governor is allowed) and the parser has no means to make a choice. In Fig. 6, we see three situations of this type: article *the* can determine either *general* or *remark*, the word *general* can be subordinated by means of different syntactic relations either by *remark*, *make* or *general*, and *remark* can be linked either to *make* or to *be.* Note that *make* can subordinate *remark* by two different syntactic relations. The latter can be either the first complement of *remark* (as in *make the remark*), or the second complement (as in *make (the general) remark*). Obviously, some of these options rely on different part-of-speech characteristics of ambiguous words. For example, *remark* is a noun in *make the remark* and a verb in *make (the general) remark.*

For each word with alternative links, the user should choose one option and click the corresponding square. In Fig. 6 the phrase *the remark* is given priority over the phrase *the general.*

Often enough, we need not resolve all the ambiguities identified by the system. It may be the case that one choice made by the user is sufficient for the system to resolve the remaining ambiguities on its own. In our example, the resolution of any one of the ambiguities shown in Fig. 6 directly leads to automatic disambiguation of the remaining ones and to the construction of a UNL graph.



Syntactical disambiguity (Light mode syntax).

We made the general remark that everything was fine.

Select correct syntactic relation for word the

| | | |
|---|---|---|
| □ | DETERM | GENERAL ---> THE |
| ☑ | DETERM | REMARK ---> THE |

Select correct syntactic relation for word general

| | | |
|---|---|---|
| □ | MODIF | REMARK ---> GENERAL |
| □ | COMPOS | REMARK ---> GENERAL |
| □ | 1-COMPL | MAKE ---> GENERAL |
| □ | NONACT-COMPL | MAKE ---> GENERAL |
| □ | PREDIC | BE ---> GENERAL |

Select correct syntactic relation for word remark

| | | |
|---|---|---|
| □ | 1-COMPL | MAKE ---> REMARK |
| □ | 2-COMPL | MAKE ---> REMARK |
| □ | PREDIC | BE ---> REMARK |

Select correct syntactic relation for word that

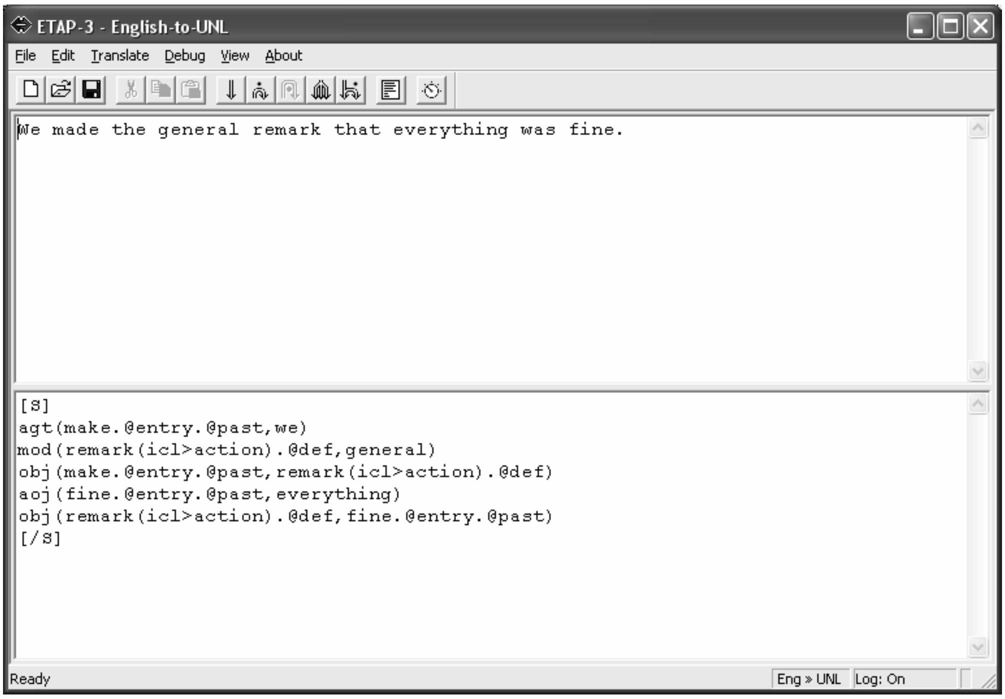
Select correct syntactic relation for word the

□ Show All

□ Don't show next time

Fig. 6

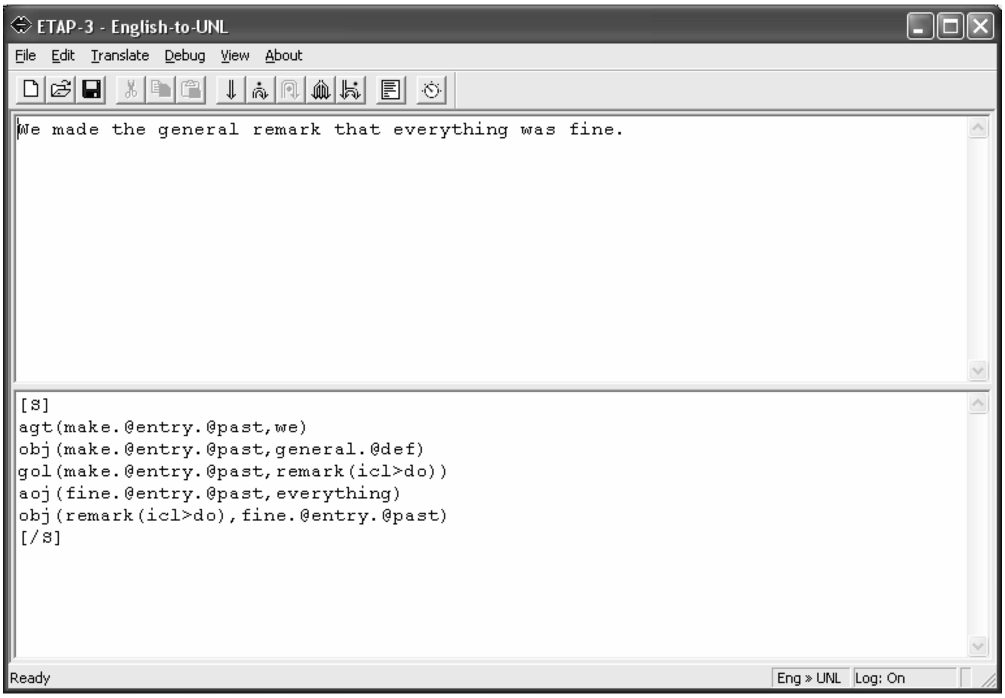
After the one choice made in Fig. 6, the enconvertor comes up with the UNL graph shown in Fig. 7. If, instead of selecting the phrase *the remark,* we had opted in Fig. 6 for the phrase *the general,* the result would have been different – see Fig. 8.



```
We made the general remark that everything was fine.
```

```
[S]
agt(make.@entry.@past,we)
mod(remark(icl>action).@def,general)
obj(make.@entry.@past,remark(icl>action).@def)
aoj(fine.@entry.@past,everything)
obj(remark(icl>action).@def,fine.@entry.@past)
[/S]
```

Fig. 7



```
We made the general remark that everything was fine.
```

```
[S]
agt(make.@entry.@past,we)
obj(make.@entry.@past,general.@def)
gol(make.@entry.@past,remark(icl>do))
aoj(fine.@entry.@past,everything)
obj(remark(icl>do),fine.@entry.@past)
[/S]
```

Fig. 8

## 5. Future work.

The interactive enconverter described above needs further improvement in the following directions.

First, the questions on the syntactic links should be supplied with clear and simple comments similar to the ones generated in the lexical ambiguity dialogue.

Second, the dialogue should be extended to the cases of UNL-related ambiguity. We mean here situations in which an unambiguous Russian or English word corresponds to more than one Universal Word.

Third, we are planning to supply a facility that allows to graphically visualize the output of the enconverter as a UNL graph and manually revise it by the human expert.

## References

Apresjan Ju.D., I.M.Boguslavsky, L.L.Iomdin *et al.* (1992a). Lingvisticheskij processor dlja slozhnyx informacionnyx sistem. (A linguistic processor for advanced information systems.) Nauka, 256 p. Moscow.

Apresjan Ju.D., I.M.Boguslavsky, L.L.Iomdin *et al.* (1992b). ETAP-2: The Linguistics of a Machine Translation System. // META, Vol. 37, No 1, pp. 97-112.

Boguslavsky I.(1995). A bi-directional Russian-to-English machine translation system (ETAP-3). // Proceedings of the Machine Translation Summit V. Luxembourg.

Boguslavsky I., N. Frid, L. Iomdin, L. Kreidlin, I. Sagalova, V. Sizov (2000). Creating a Universal Networking Language Module within an Advanced NLP System // Proceedings of the 18th International Conference on Computational Linguistics (COLING 2000), 2000, p. 83-89.

Boguslavsky I. (2001). UNL from the linguistic point of view. // Proceedings of The First International Workshop on MultiMedia Annotation. Electrotechnical Laboratory SigMatics, Tokyo, 2001, 1-6.

MIDDIM 1996 – Proceedings of the MIDDIM-96 seminar on interactive disambiguation. Le Col de Porte, 1996.