From Graph to Tree: 
Processing UNL Graphs using an Existing MT System

Etienne BLANC & Gilles SERASSET
GETA, CLIPS-IMAG
BP 53, F-38041 Grenoble cedex 09
Etienne.Blanc@imag.fr Gilles.Sérasset@imag.fr

Abstract
We describe the transfer of an UNL graph into an equivalent tree, allowing to build UNL deconverters using existing MT systems based on tree processing.

1. Introduction
The strength of the Universal Networking Language lies precisely in its universal character. In particular it is not bound to a particular processing system, even though the Deco and Enco tools are particularly suited for the processing of the UNL language.

That means that existing MT systems may quite easily be adapted to UNL processing. We will explain here how GETA’s MT environment was used to build a UNL deconverter.

The first problem we were faced to lies in the fact that GETA’s MT environment uses trees and not graphs. This being the case for many MT systems, it is on this problem that the stress will be put.

But a short description of ARIANE, Getà’s MT environment, will be first useful.

2. ARIANE-G5, a generator of MT systems
ARIANE-G5 is a generator of MT systems, that is an integrated environment designed to facilitate the development of MT systems (Boitet, 1997).

These MT systems are written by a linguist using specialized languages for linguistic programming.

ARIANE is not devoted to a particular linguistic theory. The only strong constraint is that the structure representing the unit of translation (sentence or paragraph) has to be a decorated tree.

Fig 1 shows an overview of the structure of a transfer MT system designed using the ARIANE environment. This structure is highly modular. The three steps (analysis, transfer, generation) are performed through several lexical and structural phase, some of which are optional.

ARIANE runs presently on IBM mainframes.

3. Using ARIANE-G5 for UNL deconversion
Making use of the flexibility and modularity of ARIANE, we could easily build a UNL deconverter, whose structure is given Fig.2

The input module of this deconverter realizes a graph to tree structural transfer, external to ARIANE: the UNL graph, deep representation of the text meaning, is transferred into a tree, which is the only structure ARIANE may process. This input module achieves simultaneously the lexical transfer from the Uws to the French lemmas.

Starting from this deep representation of the text, we could have directly designed a generation step to generate the French text. But, having already at hand a French generator, we preferred to build a transfer step, internal to ARIANE, adapting our input tree to the structure required by this French generator.

A fundamental problem was of course to be sure that all the information contained in a UNL graph was effectively transferred into the input tree by the external transfer module.

We will begin by discussing this quite general problem of the transfer from a UNL graph representation into an equivalent tree representation. The following processing by ARIANE, particular to our MT system, will not be discussed here. Some details may be found in (Blanc, 2000).

We will then discuss the use of the restrictions of the UWs in the processing. Their main use is...
He doesn’t open the window.

[S]
;<SUZHOU_5>
agt(open(icl>do).@entry.@not,he)
obj(open(icl>do).@entry.@not,window.@def)
[/S]

Il n’ouvre pas la fenêtre.

Figure 1: the Ariane-G5 system as used for generating a transfer MT system.

Figure 2: The ARIANE-G5 system as used for generating an UNL deconverter.

Figure 3: An elementary UNL graph and its equivalent tree. From up to down: the source sentence; the corresponding graph; the structure of the graph; the structure of the tree; the decoration of the tree, the French sentence.
obviously to ensure an unambiguous lexical transfer. But their role is far more than that, as we will illustrate on a few examples.

We will end by briefly presenting an application of the reverse transfer, that is from tree to graph, which we use in our rustic, but complete, UNL graph editor.

4. From graph to tree

We will illustrate the graph to tree transfer by using several examples, beginning by the simplest one, where the graph has in fact a tree structure.

4.1. Elementary graph with tree structure

Fig. 3 shows an elementary graph, having a tree structure, and the corresponding tree at the output of the external transfer module.

Graph and tree have the same geometry, but the presence of a additional node, a « restriction node », in the tree (node 2). Such nodes appear during the lexical transfer. This lexical transfer transforms the UW attached to a node of the graph into a French word attached to the corresponding node of the tree, the correct French word being deduced from the headword and the restriction of the UW.

But beside this role in the lexical disambiguation, the restriction of the UW may be very useful in the further processing, as will be shown below. So, after translating the UW into a French word, we kept the restriction of the UW in the tree. For a technical reason, it was more convenient to keep in a daughter node specially created for this purpose rather than to attach it to the node itself.

The decoration of a node (other than a « restriction node »), such as node 3 for instance, comprises 3 parts:
- the UNL relation of the arc pointing towards the corresponding node in the graph : RSUNL(OBJ)
- the attributes of the node : VARUNL(DEF)
- an instance number : INST(2)
- and the lexical information attached to the French word in the French dictionary : CAT(CATN), GNR(FEM).

4.2. Graphs containing nodes with more than one mother node (or whose entry node have a mother node)

In a tree, the root node has no mother node, and the other nodes have only one mother node. This is of course generally not the case for a graph, where the entry node may have a mother node, and other nodes may have several mother nodes.

Let’s for instance consider the graph of fig. 4, where the entry node (« institute ») has a mother node (« establish ») the arc joining the first node to the second bearing the relation \textit{obj}:

\textit{obj(establish(icl>found).@past, institute (icl>facilities).@present.@entry)}

In order to get a tree, with a root node without mother node, the relation is inverted in the transfer module, and becomes

\textit{xxobj(institute(icl>facilities).@present .@entry, establish(icl>found).@past)}

where \textit{xxobj} represents the relation inverse of the \textit{obj} relation. The first graph element expresses that « institute » is the \textit{obj} of establish, whereas the second expresses the fact that « establish » has « institute » as \textit{obj}.

Another point should be noted about the transfer of this graph : the UW « United Nations University » is translated into an French compound word (nodes 4 and 5). The information relative to the compound word (for instance the \textit{aobj} relation) is attached to node 3, head of the compound word, whereas the information relative to its components are attached to nodes 4 and 5 ; the relation between the nodes of the compound word is noted as a UNL relation : the \textit{mod} relation between « Institute » and « Nations-Unies ». The entry node of the compound word (node 4) is given the semantic relation \textit{grp} (for grouping).

4.3. Graph with a closed circuit

Fig. 5 shows a graph containing a circuit. The transfer into a tree is achieved by « opening » the circuit by duplication of one of its nodes. Here the node « lecturer » has been duplicated. The information that both tree nodes represent the same original graph node is given by their instance numbers. And so the French sentence will read « Le conférencier a lu sa communication » and not « Le conférencier a lu la communication du conférencier », which would be obtained with 2 differences instances of the node « lecturer ».

4.4. Processing of the compound Uws

The compound UW in the graph of fig. 6 is processed as the French compound word of the graph of fig. 4 (cf § 4.2) : a « head » of the compound UW is created in the tree (node 6) ; this node is decorated with the information relative to the scope (semantic relation of the arc pointing towards the compound UW, attributes of the compound UW). As was the case for a compound French word (cf §4.2), the entry node of the scope is given the semantic relation : \textit{grp}.

Let’s remark that this compound UW has been duplicated due to the present of the closed circuit in the graph, but that in fact only the head node has been duplicated (node 13), the remaining of the compound node in the duplicate being restablished during the
ARIANE processing. As said in the preceding paragraph, the original node and its duplicate represent the same object, so that the French sentence reads « Il sait que tu ne viendras pas et il le regrette », and not « Il sait que tu ne viendras pas et il regrette que tu ne vienne pas »: the second occurrence of the compound UW has been replaced by a pronoun.

The United Nations University is an institute which was founded by the United Nations General Assembly in 1975.

The lecturer read his paper.

[5]
[<SUZHOU_2>]

agt(read(icl>do).@entry.@past,lecturer.@def)
obj(read(icl>do).@entry.@past,paper(icl>article))
pos(paper(icl>article),lecturer)

Le conférencier a lu sa communication.

Fig.5 Processing a graph with a circuit : duplication of the node « lecturer ».
He knows that you will not come and regrets it.

Il sait que tu ne viendras pas et il ne le regrette.

Fig. 6 Processing of a compound UW: a head node is created in the tree (node 6), which bears the information relative to the compound UW. Only the head node is duplicated (node 13) when duplication is required by the presence of a closed circuit in the graph.
5. Use of the restriction beside the lexical disambiguation

As said above, the role of the restrictions is not limited to the lexical disambiguation, it is the reason these restrictions are kept in the tree as « restriction nodes », as said in §4.1.

We will give some examples of this use of the restrictions.

5.1. Helping to determine the argumentary or the syntactic function

In the UNL language, one distinguishes the verbal concepts do, occur, be. For instance, the graph of fig. 3 contains the UW « open(icl>do) », whereas the graph of fig. 7 below contains the UW « open(icl>occur) ».

Both UWs are translated into French by the same verb, « ouvrir » (or in English by the same verb « to open »). But the subject syntactic relation for the French (or the English) verb corresponds to the agt relation in the case of the « open(icl>do) » UW, but to the obj in the case of the « open(icl>occur) » UW. That means that the restriction had to be tested in order to find the subject of the sentence.

<table>
<thead>
<tr>
<th>The window doesn't open.</th>
</tr>
</thead>
<tbody>
<tr>
<td>[S] ;&lt;SUZHOU_4&gt;</td>
</tr>
<tr>
<td>obj(open(icl&gt;occur).@entry.@not,window.@def)</td>
</tr>
<tr>
<td>[/S]</td>
</tr>
<tr>
<td>La fenêtre n’ouvre pas.</td>
</tr>
</tbody>
</table>

Fig 7 The obj relation of this graph corresponds to the syntactic subject relation in French or English

5.2. Treatment of an unknown UW

Fig. 8 gives the result of the processing of a graph comprising unknown words, that is UWs not present in the French dictionary.

Testing the restrictions of the unknown UWs rake(icl>do) and rake(icl>thing) indicates that the first one is a verbal concept, the second one a thing concept, which allowed a correct construction of the sentence.

In fact, even in the absence of the restrictions of the unknown UWs in the graph of fig. 8, the French sentence would have been correctly constructed (Fig. 9). Another treatment of the unknown word is indeed based on the context. For instance the first instance of the UW rake, being the origin of an agt relation, is a verbal concept

<table>
<thead>
<tr>
<th>He rakes the leaves with the big rake.</th>
</tr>
</thead>
<tbody>
<tr>
<td>[S] ;&lt;SUZHOU_6&gt;</td>
</tr>
<tr>
<td>agt(rake(icl&gt;do).@entry,he)</td>
</tr>
<tr>
<td>obj(rake(icl&gt;do).@entry,leaf(fld&gt;botany).@def.@pl)</td>
</tr>
<tr>
<td>ins(rake(icl&gt;do).@entry,rake(icl&gt;thing))</td>
</tr>
<tr>
<td>mod(rake(icl&gt;thing),big(mod&lt;thing))</td>
</tr>
<tr>
<td>[/S]</td>
</tr>
<tr>
<td>Il &lt;&lt;rake&gt;&gt; les feuilles avec le?</td>
</tr>
<tr>
<td>grand? &lt;&lt;rake&gt;&gt;.</td>
</tr>
</tbody>
</table>

Fig 8 The UWs are not present in the French dictionary, the sentence could nevertheless be correctly built using the restriction in order to determine the nature of the unknown UWs.

<table>
<thead>
<tr>
<th>He rakes the leaves with the big rake.</th>
</tr>
</thead>
<tbody>
<tr>
<td>[S] ;&lt;SUZHOU_6&gt;</td>
</tr>
<tr>
<td>agt(rake:01.@entry,he)</td>
</tr>
<tr>
<td>obj(rake:01.@entry,leaf(fld&gt;botany).@def.@pl)</td>
</tr>
<tr>
<td>ins(rake:01.@entry,rake:02)</td>
</tr>
<tr>
<td>mod(rake:02,big(mod&lt;thing))</td>
</tr>
<tr>
<td>[/S]</td>
</tr>
<tr>
<td>Il &lt;&lt;rake&gt;&gt; les feuilles avec le?</td>
</tr>
<tr>
<td>grand? &lt;&lt;rake&gt;&gt;.</td>
</tr>
</tbody>
</table>

Fig 9 The type of an unknown UW may be obtained by testing its context : rake :01 being the origin of an agt relation, is a verbal concept.

6. A simple but complete UNL graph editor based on tree to graph transfer

We used the inverse transfer, that is from tree to graph, to design a rustic, but complete UNL graph editor. We will not give here a complete description of this editor, but only show briefly its operation using fig. 10 to 12.

Fig. 10 shows a card of this Hypercard based graph editor. Using the buttons « + daughter » and « + sister » of the upper part of the card, a tree equivalent to the graph is build node by node in the upper field. (Its equivalence to a graph is obtained as shown above by duplicating nodes and using reverse relations.)
The UW of a newly created node may be either directly entered, or entered by the way of a language dictionary as shown on Fig. 11. Here the Russian word for « battery » was asked for, and the corresponding page of a Russian dictionary was then displayed. The various senses of the word « battery » are associated to various UWs. Clicking on the desired sense enters the corresponding UW into the new node.

The language dictionaries used are the constitutive parts of a hypertextual multilingual database having the UWs as pivot (Blanc 1999).

The decoration of a node (binary relations and attributes) are entered by selecting the node on the tree and activating the button « deco » on the upper part of the card. A selection window is then displayed (fig. 12), and relations and attributes are selected by simple clicking on their names.

When the tree is complete, activating button « arbre -> graphe » on the bottom of the card deduces the graph from the tree.

Fig. 10 A card of the UNL graph editor. First upper field: the structure of the equivalent tree is build node by node using the « + daughter » and « + sister » buttons. The UW of each node is selected using the dictionary window (cf fig. 11). The third field contains the decoration of the nodes, entered using the « deco » button and then the selection window of fig. 12. When the tree is complete, activating button « arbre -> graphe » on the bottom of the card deduces the graph from the tree.
Fig. 11 Selecting the UW of a node under creation.

Fig. 11 Selecting the binary relation and the attributes of the selected node.
7. References

