Interpretation of Multilingual Documents in e-speranto Using the Client-Server Architecture Model

Jakus, Grega; Varga, Emil; and Tomažič, Sašo

Abstract — ***The paper introduces an overview of a multilingual translation system based on e-speranto.***

E-speranto is a computer language intended for recording multilingual documents on the Web. It can also serve as an intermediate language, or interlingua, in multilingual translation. Its advantage over similar approaches is the compatibility with HTML and the intelligibility of documents both to computers as well as to people.

The development of the e-speranto-based translation system consists of four stages. They include the development of language, tools for writing documents directly in e-speranto, interpreters of e-speranto and translators from a natural language to e-speranto.

A proof-of-concept implementation of the translation system uses the components from the first three stages of the development. These components are organised using the client-server architecture model.

By using a Web browser, the user can request the interpretation of a multilingual document. On the server side, the requests are passed to the interpreter of e-speranto. The result of the interpretation is a HTML document that is returned to the client. Language-specific data, such as language rules, word and phrase dictionaries, are contained in a database. The database can be located on the client or on the server.

The interpretation of a document in e-speranto is realized in three steps that include semantic generation, lexical transformation and structural transformation.

The interpreters implement a modular architecture. The modules are divided into three layers of abstraction. A level of abstraction refers to the degree of abstractness of language structures that enter a certain module as data and on which transformations are performed. The procedures in the modules in the first layer perform language-independent operations that are common to rule-based machine translation. The procedures in the second layer are typical of a group of languages, while the ones in the third layer are language-specific.

Manuscript received April 28, 2009.

G. Jakus and S. Tomažič are with the Faculty of Electrical Engineering, University of Ljubljana, Slovenia. E. Varga is with the School of Electrical Engineering, University of Belgrade, Serbia (e-mails: grega.jakus@fe.uni-lj.si, emil.varga@gmail.com saso.tomazic@fe.uni-lj.si).

Index Terms — ***client, e-speranto, interpretation, multilingual translation, server***

1. INTRODUCTION

THE multilingualism is gaining importance in today's society when the World is in the process of globalization. The barriers and divides among different nations and cultures are blurring. A typical example of such a process in Europe is the uniting of nations and their economies in the European Union. While different markets are merging into single one, the language divides remain.

One of the demanding problems related to the multilingualism on the Web is the number of units needed for translation among different languages. The problem is burning since there are practically no political, geographical or any other kind of obstacles for the interaction among individuals. One can come across any language while using the Internet. In order to develop translators for all known 6900 languages that are spoken in the World today, about 47,610,000 of translators should be made.

One of the possible solutions to this problem is the use of an intermediate language or interlingua. Interlingua is an abstract presentation of the content that is independent from any natural language [1]. The record in interlingua must contain the whole information required for generating text in a natural language. Thus, the entire meaning we want to express in a natural language must be captured in interlingua. The advantage of using the interlingua is a two-phase course of translation between two natural languages. During the process, the modules that perform the conversion from a natural language to the interlingua (translators) are independent of those that perform the opposite conversion (interpreters). Moreover, the interpreters and translators of different languages are also mutually independent. The effect of this independence is the reduction of the number of units that would be needed in case of direct mapping among the individual languages. The cost of the latter approach is as high as $n(n-1)$, where n denotes the number of languages among which we want to translate. By using the interlingua approach, the cost of the interpreter

development reduces to *2n*, since only a translator and an interpreter for each language must be made.

2. Related Work

Numerous attempts of creating an interlingua-based multilingual system were conducted in the past. Some more not able implementations are presented in the next paragraphs.

DLT (*Distributed Language Translation*) [2] was a project of the development of a multilingual system in the 1980s that used an adapted version of Esperanto as an interlingua. The document written in Esperanto would be carried over the network and interpreted in a chosen language by the target computer. Although DLT presented a novel and interesting approach to machine translation, the results were not promising in practice.

The KANT system [3] is based upon controlled English (a language with a limited scope of vocabulary) and was created with the intention of translating technical documentation. It produces very accurate sentences, but due to the limited field of use it is not directly applicable for general multilingual translation.

UNL (*Universal Networking Language*) [4] is a computer language for recording and exchanging information and it is basically intended for communication on the Web. It supports 15 languages, which makes it currently one of the largest multilingual systems intended for use on the Web. Its main deficiency is the limited power of expressiveness [5] and somewhat lower degree of intelligibility of texts written in this language. The latter already proved as a disadvantage during the development of Internet standards in the past.

Generation of a text from an interlingua can be carried out in different manners [1]. Most often it is based on language rules (rule-based) that define the conversion from a source presentation to the target one. Another widely used approach is based upon the semantic and pragmatic knowledge of a certain field (knowledge-based). The statistical and example-based approach are not generally used in generation of texts from an interlingua because they both require a bilingual corpus which is, however, hard to create when one of the languages is an interlingua.

The majority of the approaches of the rule-based natural language generation has a modular design with two basic steps of conversion. They represent the lexical transformation (conversion of the lexical units) and structural transformation between the language structures in both languages [1]. When generating text from an interlingua, the *semantic generation* is also present. During this step, the deep syntactic structure of the content is generated according to the semantics in the interlingua representation.

An example of a well-established framework that uses similar design is ARIANE [6]. ARIANE is a flexible framework for the development of machine translation systems between language pairs which can also be an interlingua and a natural language. The system separates the algorithms from the linguistic content as the parameterization of algorithms. Several interpreters of the interlingua are based on ARIANE, for example in French [7], [8].

3. E-speranto

E-speranto [9], [10] is a formal computer language for recording multilingual texts. Its main goal is to overcome language divides in the Internet. A document in e-speranto is interpreted in a chosen language, when the user requests the document to be displayed.

The basic syntax of e-speranto is based on XML (*eXtensible Markup Language*) which is an important technology on the Web today. XML is compatible with HTML and therefore e-speranto can be incorporated into Web pages. Syntactic and grammatical rules are taken from Esperanto, but are expressed explicitly by means of metadata.

The main advantage of e-speranto over similar approaches is especially the intelligibility of documents both to computers as well as to people and compatibility with existing Web technologies, such as XML and HTML.

The development of the e-speranto-based translation system consists of four stages. They include the development of language, tools for writing documents directly in e-speranto, interpreters of e-speranto in a natural language and translators from a natural language to e-speranto. When the interpreters of e-speranto are developed, the displaying of Web pages in e-speranto will be possible. When the translators from natural languages to e-speranto are available, also multilingual translation between different pairs of languages using e-speranto as an interlingua will be feasible.

4. Interpretation of E-speranto in Natural Languages

The interpretation of a document in e-speranto is based on tree transformations. A tree consists of symbols that denote concepts, concept attributes and the relations among concepts. It contains enough information to perform the interpretation on the semantic, morpho-lexical and syntactic level. A tree is represented in the syntax of Mathematica programming language in which the core of the prototype interpreter INES (the **IN**terpreter of **E**-**S**peranto) was made.

Because e-speranto is a computer language, we can draw some parallels with the interpreters and translators of computer programming

```
<sentence original="E-speranto is a design of a computer language." feelings="declarative" organization="simple">
  <subject detail="personal_name" number="singular">
    <word>E-speranto</word>
  </subject>
  <predicate detail_predicate="main" mood="indicative" voice="active" tense="present" person="third">
    <word>be</word>
    <subordinate>
      <predicate detail_predicate="predicate_noun" number="singular">
        <word>design</word>
        <subordinate>
          <object detail_object="of_genitive" number="singular">
            <word>language</word>
            <subordinate>
              <attribute>
                <word>computer</word>
              </attribute>
            </subordinate>
          </object>
        </subordinate>
      </predicate>
    </subordinate>
  </predicate>
</sentence>
```

Figure 1: Record of a sentence in e-speranto. The basic building element in e-speranto is a clause. A clause is a semantic unit that corresponds to a sentence in a natural language. Clauses are composed of sentence elements introduced by XML tags. The grammatical characteristics are expressed explicitly by means of XML attributes. The concepts representing the essence of e-speranto are marked in English for the sake of better intelligibility.

languages regarding the interpretation. Namely, the process of interpretation is similarly divided into several stages. We distinguish:

- lexical and syntactical analysis of the source code,
- generation of the intermediate code,
- code optimization, and
- compilation phase.

The lexical and syntactic analyses are performed every time during the composition of a document in e-speranto and are provided by the development environment. For this purpose the development environment based on the Eclipse platform was developed. The built-in XML editor performs the verification of the document conformity with the e-speranto grammar, syntax and vocabulary.

The other phases are realized in the INES interpreter. The conversion of the e-speranto document into the expressions of the Mathematica language is analogous to the generation of the intermediate code. The phase of code optimization in the classical translators corresponds to the adaptation of the intermediate representation structure to the form that is used in the process of interpretation in INES (compare Figures 1 and 2). The phase of optimization is important since it enables, to a certain degree, the independence of the tree structures in INES from the changing grammar and syntax of e-speranto, as the latter is still being developed.

The compilation phase is the main step of the interpretation in a selected natural language. In INES this phase is realized in three steps.

In the first step, the so-called *semantic generation* is carried out. For each e-speranto concept a syntactic structure is generated that represents the same meaning in a natural language as the corresponding e-speranto concept. This is in accordance with the *principle of compositionality* which states that the meaning of a phrase is a function of its constituents and their position in a phrase. In the interpretation of e-speranto, this phase is important for the generation of common phrases and idioms.

In the second step (the lexical transformation), lexical units are formed on the basis of lexico-morphological attributes of concepts in e-speranto and linking of these concepts to the lexical units in a natural language (e.g. e-speranto record {*'lingua', tense: plural*} is interpreted as '*languages*' in English).

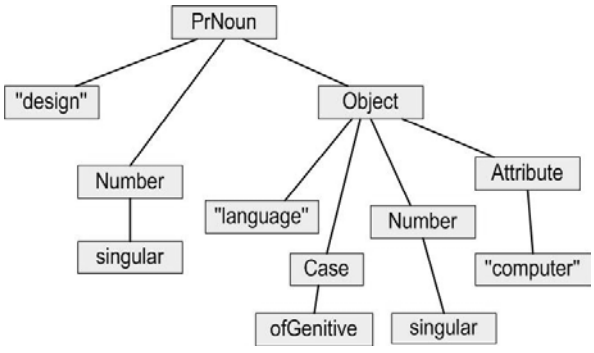

Figure 2: Simplified representation of the noun phrase "*design of a computer language*" from the sentence in the Figure 1 in the form of the tree structure of the symbolic programming language Mathematica.

In the third step, the structural transformation is carried out. The tree that comprises lexical units is rearranged according to the purpose of the message (e.g. the mood of the sentence), the syntax of the target language, etc. This step also includes the processing of punctuation and conjunctions. Finally, the tree is transformed to a sequence of lexical units representing a sentence in a natural language.

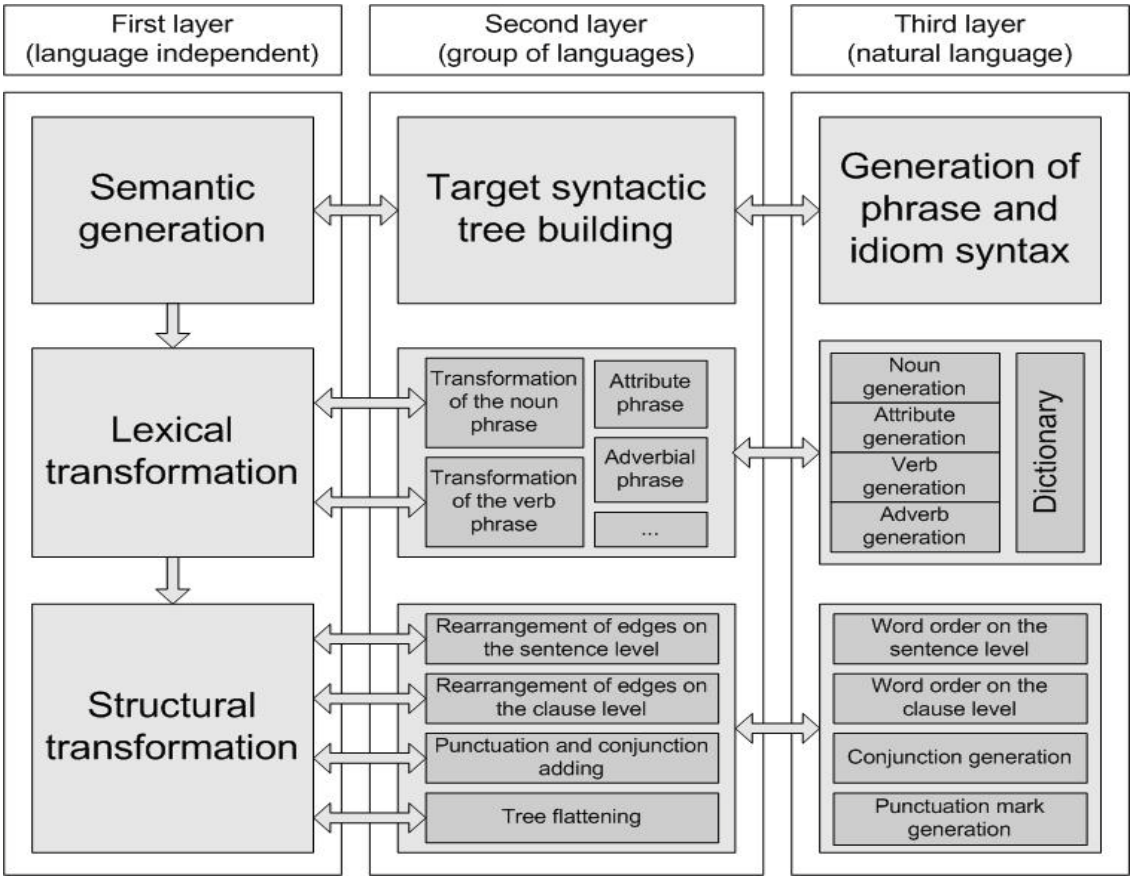

Figure 3: Scheme presenting operation of INES. Every phase of interpretation is divided into three layers. The procedures in the modules in the first layer perform language-independent operations that are common to rule-based machine translation. The procedures in the second layer are typical of a group of languages, while the ones in the third layer are language-specific.

5. *Architecture of the Interpreters*

The interpretation is carried out with the modules that are arranged into three levels of abstraction [11]. A level of abstraction refers to the degree of abstractness of language structures that enter a certain module as data and on which transformations are performed. The architecture of INES with some distinctive procedures on individual layers is shown in Figure 3.

The first layer comprises modules that dictate the course of interpretation and are independent of the target language. The layer is only aware of the fact that a sentence in a natural language contains the elements that express an action or activity (i.e. the predicate) and the holder of this action or activity (i.e. the subject). This layer also contains the algorithms for movements in the tree structure. Among the various possible methods the recursive *depth-first search* algorithm is implemented in INES. The individual subtrees are identified according to their type; their transformation is then performed by the lower layers.

The modules in the second layer are closer to the language families. These modules in general perform transformations of particular subtrees in accordance with their type. The type of a subtree is determined by the syntactic and/or semantic role the root element is performing according to the parent element in the tree representation. In general, a subtree of a certain type corresponds to a particular clause or its part in a sentence of a natural language. Figure 2 shows a subtree that corresponds to a predicate noun in a natural language.

The procedures with language-specific rules can be found in the third layer. These procedures map the parts of a tree structure to the elements of a natural language in a way that is specific to the language of interpretation. An example of such a transformation is the replacement of the e-speranto concepts and their attributes with the words of the target language or the rearrangement of the tree edges in accordance with the word order in which particular clauses appear in the target language. The access to word and phrase dictionaries is also implemented in this layer.

The characteristics that are common to a certain group of languages can be introduced on an abstractly higher level (e.g. on the second layer for a particular group of languages) than the actual characteristics of individual languages in this group (the third layer). The use of this concept reduces the cost of development for enabling some of the modules to be reused when developing the interpreters of related languages.

6. ARCHITECTURE OF THE SYSTEM PROTOTYPE

The implementation of the translation system prototype (Figure 4) is roughly divided into two parts: the client side and the server side. The server side is only present during the development of the e-speranto language and the interpreters. When the language is standardized, the interpreters will migrate from the server to the client. The functionality of the interpreters will be implemented in browsers through the plug-ins.

The performance of the system prototype can be tested on the project's Web site [12].

6.1. Client side

On the client side, the documents in e-speranto are composed. For this purpose an integrated development environment (IDE) based on the Eclipse platform [13] was developed. The IDE provides the user with the content assistance, document validation, access to the dictionaries and their effective use during the composition of documents.

By using a Web browser, the user requests the interpretation of a multilingual document that is located on the server.

6.2. Server side

On the server side, Apache Tomcat, a Java HTTP Web server, handles the requests for the interpretation of documents in e-speranto. The requests are passed to the INES. The interface between the Internet and the core of the interpreter is written in Java programming language.

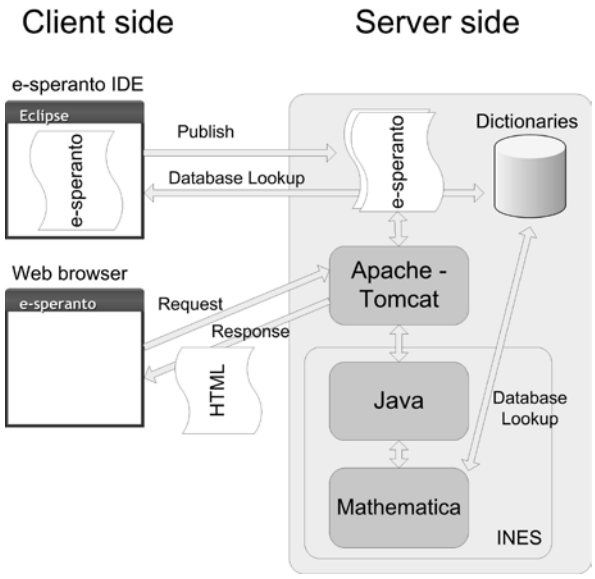

Figure 4: Implementation of the multilingual translation system based on e-speranto using the client-server architecture model.

The sentences in an e-speranto document are converted to symbolic trees. Beside the transformation of the data structures, the compilation of a "starter script" is carried out. The script comprises trees that need to be transformed, specifications of the modules that are needed to perform the interpretation, as also global variables and processing instructions. The module selection depends mostly on the language of interpretation.

6.3. Database

The database is primarily located on the server, but it can also be located on the client. Database serves as a source of language-specific content such as language rules, word and phrase dictionaries.

7. CONCLUSION

In this paper we introduced a brief overview of the multilingual machine translation system based on e-speranto. The most important features of the presented system are:

- the use of an interlingua that is compatible with HTML and it can be therefore used for recording multilingual content directly in Web pages;
- the use of an interlingua that is simple enough to be user-friendly, and is complex enough to have enough expressiveness and unambiguity;
- the support for composing documents in e-speranto by the tools developed on the Eclipse platform;
- the implementation of the architecture of the interpreters that allows the reuse of some modules and therefore it reduces the cost of the development of the interpreters into similar languages.

Our further research will be towards the following fields. We intend to define the format and the content of the dictionaries in detail. Moreover, we intend to upgrade the integrated development environment with features that would enable automatic publishing of the composed e-speranto documents on the Web server. We also want to perform a more thorough research in the optimal number of layers in the layered architecture of the interpreters and determine the content that needs to be placed into individual layers. In this context, we will pursue the aim of optimally high factor of module reuse when interpreting into similar languages.

REFERENCES

[1] Hutchins, W., Somers, H., "An Introduction to Machine Translation", *Academic Press*, New York, 1992.

[2] Schubert, K., "The Architecture of DLT – interlingual or double-dialect", *Floris Publications*, New Directions in Machine Translation, Holland, 1988.

[3] Nyberg, E., Mitamura, T., " The KANT system: Fast, accurate, high-quality translation in practical domains", COLING, 1992.

[4] Uchida, H., et al., "Universal Networking Language: A gift for a millennium", *The United Nations University, Tokyo*, Japan, 1999.

[5] Bugoslavsky, I., “ Some c ontroversial Issues of UNL: Linguistic A spects”, Uni versal Netw ork Language: Advances in Theory and A pplications, Res earch on Computer Science, 2005.
[6] Boitet, C., “GETA’s methodology and its current developments”, PACLING’97, M eisei Uni versity, O hme, Japan, 1997.
[7] Sérasset, G., B oitet, C., “ On UNL as the future ”html of the l inguistic c ontent” and r euse of existing NLP components i n UNL -related appl ications w ith the example of a UNL-French deconverter”, COLING, 2000.
[8] Blanc E., From the UNL hypergraph to GETA's multilevel tree, M T2000: m achine tr anslation and m ultilingual applications in the new millennium, Uni versity of E xeter, British Computer Society, 2000.
[9] Omerović S., J akus G., F ilimonova T., Tomažič S., “Zapis večjezičnih besedil v e-sperantu”, *Elektrotehniški vestnik*, Vol. 74, No. 3, 2007, pp. 151-157.
[10] Tomažič S., “Multilingual Web with E -speranto”, *IPSI Bgd Internet Research Society*, The IP SI B gD Transactions on Inter net Res earch, Vol. 3 , No. 2 , J uly 2007, pp. 13-15.
[11] Jakus G., Omerović S., Filimonova T., Tomažič S., “Classification of the language group characteristics into a multi-layered architecture of the interlingua interpreter in multilingual tr anslation”, *Elektrotehniški vestnik*, V ol. 75, No. 5, 2008, pp. 285-292.
[12] e-speranto Web Page, http:// www.e-speranto.org
[13] Eclipse, http://www.eclipse.org