Journal of Universal Computer Science, vol. 17, no. 3 (2011), 377-398
submitted: 25/5/10, accepted: 29/1/11, appeared: 1/2/11 © *J.UCS*

The Architectural Design of a System for Interpreting Multilingual Web Documents in E-speranto

Grega Jakus
(University of Ljubljana, Ljubljana, Slovenia
grega.jakus@fe.uni-lj.si)

Jaka Sodnik
(University of Ljubljana, Ljubljana, Slovenia
jaka.sodnik@fe.uni-lj.si)

Sašo Tomažič
(University of Ljubljana, Ljubljana, Slovenia
saso.tomazic@fe.uni-lj.si)

Abstract: E-speranto is a formal language for generating multilingual texts on the World Wide Web. It is currently still under development. The vocabulary and grammar rules of E-speranto are based on Esperanto; the syntax of E-speranto, however, is based on XML (*eXtensible Markup Language*). The latter enables the integration of documents generated in E-speranto into web pages. When a user accesses a web page generated in E-speranto, the interpreter interprets the document into a chosen natural language, which enables the user to read the document in any arbitrary language supported by the interpreter.

The basic parts of the E-speranto interpreting system are the interpreters and information resources, which complies with the principle of separating the interpretation process from the data itself. The architecture of the E-speranto interpreter takes advantage of the resemblance between the languages belonging to the same linguistic group, which consequently results in a lower production cost of the interpreters for the same linguistic group.

We designed a proof-of-concept implementation for interpreting E-speranto in three Slavic languages: Slovenian, Serbian and Russian. These languages share many common features in addition to having a similar syntax and vocabulary. The content of the information resources (vocabulary, lexicon) was limited to the extent that was needed to interpret the test documents. The testing confirmed the applicability of our concept and also indicated the guidelines for future development of both the interpreters and E-speranto itself.

Keywords: E-speranto, World Wide Web, multilingual documents, XML, interpretation, multilayered architecture
Categories: D.2.11, H.5, I.2.7

1 Introduction

In the two decades of its existence, the World Wide Web has been subject to continuous development. If the Web once used to be in the sole domain of scientists whose data exchange took place mostly in English, it is nowadays also an indispensable tool for the business world, media, and social networks, and its users come from virtually every corner of the world.

In the beginning, the access to information was limited by the so-called *digital divide* separating the users with access to modern information-communication systems from those that did not have such access. Nowadays, with internet access points available almost in every village, a much more pressing problem is the so-called *language divide*, which means that the information on web pages is not readily available to the majority of potential users because they do not understand the language of the page. Despite the facts that nowadays almost one quarter of the world population uses the internet [IWS, 10] and that almost half of the users comes from Asia, the vast majority of web pages are still solely in English.

Multilingualism is present in the World Wide Web in two ways. Most often it is reflected in the fact that web pages are translated merely into languages that – according to the owner's conviction – cover a large part of the target audience. The users that do not understand these languages can use one of the many web translation tools available online, such as [BabelFish, 10], [GoogleTranslate, 10], [Promt, 10] or [Systran, 10]. Although the translation quality in such cases is usually not the best due to the fact that one of the most prominent features of these translation tools is their responsiveness, they still enable a user to grasp the basic information on the web page in a language they understand. Web translation tools also in most cases do not enable the translation of longer texts. In addition, they often support only a limited set of languages and thus do not enable general multilingualism.

In machine translation, two issues regarding the quality and quantity of the translation tools are especially problematic. The first issue has to do with the quality of the computer's comprehension of the text written in a natural language. This is extremely difficult due to many polymorphisms, exceptions to the rules, and inconsistencies that are present in natural languages. The other issue is a scalability problem. In order to translate between n languages, we require $n(n-1)$ translation tools. If we wanted to provide automatic translation between the 6909 languages spoken in the world today [Paul, 09], we would have to make 47,727,372 translators.

The reasonable approach to solving the scalability problem is the use of an interlingua [Hutchins, 92]. This enables a two-phase translation process between two natural languages. Translating via an interlingua reduces the necessary number of modules of the multilingual system to $2n$, as each linguistic group requires only two modules that perform the transformation into an interlingua and vice versa. However, the use of an interlingua does not inherently solve the problem of the computer's comprehension of the text in a natural language. As this problem only occurs when translating from a natural language into an interlingua and not in the reverse process, the solution lies in the introduction of a formal computer-friendly language for describing multilingual documents. Such language would enable the author to create documents with the aid of specialized tools making automatic translation from a natural language avoidable. E-speranto was created to fulfil this need [Tomažič, 07]. If and when automated translators into E-speranto of sufficient quality are developed, E-speranto will also function as an interlingua in multilingual translation [Fig. 1].

In the context of E-speranto, the programs that perform the transformation from natural languages into E-speranto are called *translators*, and the programs that perform the transformations from E-speranto into natural languages are called *interpreters*. This terminology was adopted from the terminology of computer languages. As the document is created in E-speranto and is only displayed in the

chosen target language, this is called the interpretation of a document. Translation, however, means that the document that was originally written in a natural language is translated into E-speranto, which is analogous to translating the source code in the programming language into the executable code of the system where the program execution will take place.

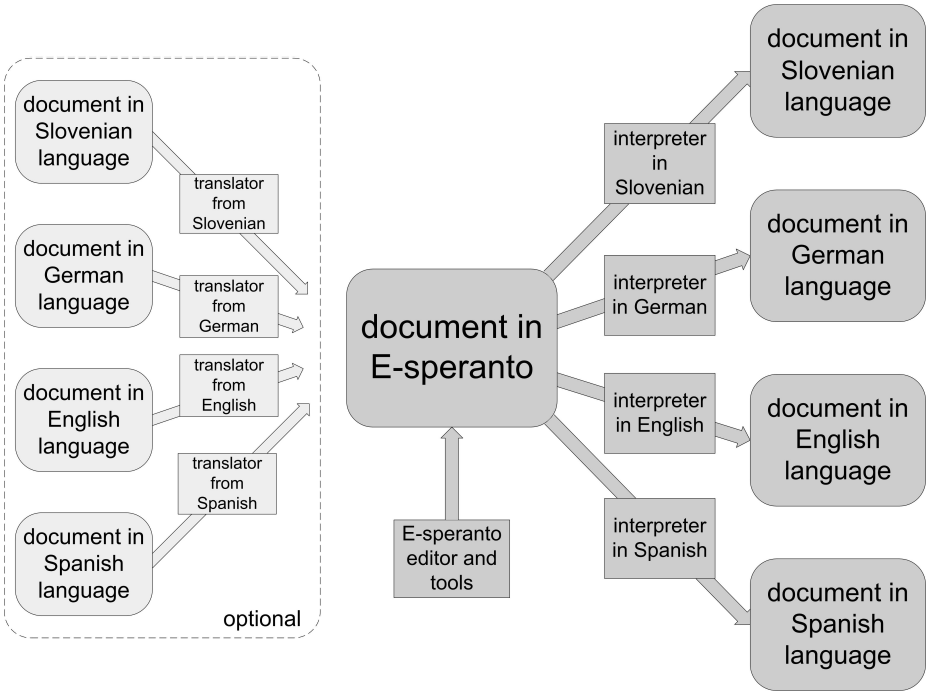

Figure 1: Generating multilingual documents on the Web with the aid of E-speranto

The remainder of the paper is organized in the following way: the next section presents the most important approaches to the development of formal languages intended for multilingual communication and the approaches to generating the content in natural languages based on the record in these languages. The third section briefly presents E-speranto, focusing on the features that are of vital importance when designing the architecture of E-speranto interpreters. The next sections present the design of the interpretation system, focusing on the architecture of interpreters and the structure of information resources. The architecture presented in this paper was used to create an actual prototype interpreter which is described in more detail in the fifth section of the paper. Subsequently, the interpreters and information resources were integrated in a system functioning as a proof-of-concept for a multilingual Web based on E-speranto. This system is presented in more detail in the sixth section. The paper ends with our findings, the simplifications that were made when developing the system, and future work.

2 Related Work

In the past, many researchers tried to create formal languages and systems based on languages that enabled multilingual communication. The majority of these languages function as an interlingua. The record in an interlingua contains all the information required to generate documents in any given natural language by including the entire meaning to be expressed. The content in the interlingua is thus presented in an abstract form independent of any natural language. Some of the most important implementations are presented below.

DLT (*Distributed Language Translation*) [Schubert, 88] was a research project that focused on developing a multilingual system based on an adapted version of Esperanto which functioned as the interlingua. The document in Esperanto was intended to be transmitted over the network and then interpreted in a chosen natural language on the target computer. During the development of the system, the researchers established that Esperanto is not appropriate for an interlingua despite the relative consistency of its grammar. Esperanto has many features similar to those of natural languages, which causes lexical and structural inconsistencies typical for direct translations between natural languages.

The interlingua KANT [Nyberg, 92] is based on English with a limited lexicon. KANT was created with the purpose of translating technical documentation. The system produces accurate translations, but the interlingua cannot be used for general multilingual communication due to the limited scope of its application.

Rosetta [Rosetta, 94] uses an interlingua based on Montague grammar [Thomason, 74]. Rosetta's intermediate representation structure is defined by the isomorphic grammars of all the languages supported by the system. As the interlingua inherently contains the features of supported natural languages, this significantly simplifies its interpretation into natural languages. However, as the interlingua contains only the elements of a limited number of natural languages, it remains to be non-universal and the system itself non-scalable, as we would need to change the interlingua if we wished to add a new natural language into the system.

UNL (*Universal Networking Language*) [Uchida, 99] is a language for the representation and exchange of information on the internet. UNL is a successor of the ATLAS-II [Uchida, 89] and PIVOT [Muraki, 87] interlinguae. In UNL, the content has the form of a graph which is expressed in the shape of linear expressions in concrete syntax. UNL enables the authors to write directly in this language; however, as the language itself was never intended for this use, it is difficult to understand it.

Although the course of interpretation in different systems can differ significantly, most approaches to forming a natural language have a modular design with two basic transformation steps – the lexical and structural transformations [Hutchins, 92]. The *lexical transformation* includes the choice of the lexical units in the target language and their morphological features that match the record in interlingua. The *structural transformation*, on the other hand, is the mapping of the language structures between the source and the target language.

A typical example of a system with such a design is ARIANE [Boitet, 97] – the framework for the development of machine translation systems. Several interlingua interpreters were developed on the basis of the ARIANE framework, for example in French [Blanc, 00], [Sérasset, 00]. ARIANE inherently differentiates between

algorithms and linguistic content. A similar approach is employed also by the TG/2 framework [Busemann, 98], [Busemann, 02]. Separating the interpretation into lexical and structural transformation is also present in the interpreters of Rosetta [Rosetta, 94], KANT [Nyberg, 92], UNL [Blanc, 02], [Dhanabalan, 03], [Singh, 07], [Spall, 07] and Mikrokozmos [Temizsoy, 98].

Due to the high level of abstractness of an interlingua, the mapping into the structures of the target language often becomes very difficult, as these differ substantially from the structures in the interlingua. In some systems, this gap is bridged by the *semantic generation* phase. In this phase, the interpreter generates the syntax structures of the target language matching the meaning of the concepts in the interlingua. The semantic generation is, for example, present in the SUSY [Freigang, 86], UNL [Blanc, 00] and partially also in the DLT [Schubert, 88] systems.

The paradigm of separating data from the interpretation procedures calls for the use of various information resources in the form of databases. Most often these databases are bilingual dictionaries, transformation rules, grammar rules of the target language, lexicons, vocabularies, and dictionaries of phrases and idioms. In addition, the systems can also use databases containing real world knowledge or pragmatic knowledge of a specific field. The actual data and their organization in different databases depend on a specific system.

3 The Generation of Multilingual Documents in E-speranto

E-speranto is a design of a formal language for generating multilingual documents on the Web [Tomažič, 07], [Omerović, 07]. Its syntax is based on XML (*eXtensible Markup Language*) [XML, 10] and is compatible with HTML (*HyperText Markup Language*), which is why the content written in E-speranto can be included in web pages. The grammar and syntax rules of E-speranto are presented in the XML Schema [XMLS, 10].

E-speranto was named after Esperanto, the auxiliary language created at the turn of the 20th century and based on linguistic elements from different Indo-European languages. Esperanto is based on sixteen rules that have very few exceptions [Amerio, 02]. The most important features of Esperanto are the consistency of its grammar and vocabulary, unambiguousness and an appropriate level of expressiveness, all of which are also the key features of languages for multilingual communication. Esperanto is the result of a broad linguistic knowledge and years of work, which is why it was used as the basis for the development of E-speranto – an electronic version of Esperanto.

In Esperanto, the grammatical features, such as for example gender or number, are expressed with the help of affixes (suffixes and prefixes), while E-speranto, on the other hand, uses an explicit record based on XML constructs. This type of document is still intelligible and can at the same time be parsed by a computer.

In E-speranto, the meaning is presented by the concepts, relations among the concepts, and concept attributes [Fig. 2]. The relations express the roles of the concepts within the message. The attributes mostly present the relationship of the author of the message to the concepts. Besides the information included in the concepts and relations, the attributes also include all the additional information necessary for the transfer of a specific meaning into a natural language without any

loss of the more subtle details (e.g. the quantity of the concept, the temporal frame, the role of the author, etc.).

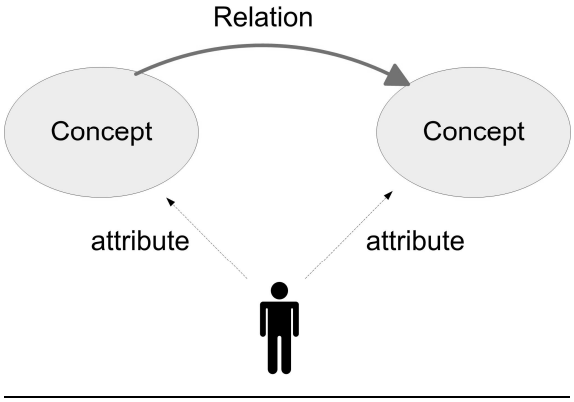

Figure 2: The basic communication model in E-speranto

In concrete syntax [Fig. 3], the concepts are recorded with the lexical units from the Esperanto vocabulary. Their features are expressed with XML attributes, after which they are linked with relations and assigned to an appropriate class. Classes have a vital role in E-speranto, both from the semantic and grammatical point of view:

1. Assigning to appropriate classes enables the distinction between the concepts as regards their role in the meaning of the message, while at the same time establishes the frame of their interpretation, as is evident from [Tab. 1].
2. Each individual class has its own subgrammar defining the syntactic and semantic limitations of the class, for example:
   - the available subordinate classes;
   - the range of attributes used to describe the concept in more detail;
   - the range of relations used to link the concept with other concepts.

```
<sentence original="E-speranto is a design of a formal language."
          feelings="declarative" organization="simple">
   <subject detail="proper_name" number="singular">
      <word>e-speranto</word>
   </subject>
   <predicate        detail_predicate="main"        mood="indicative"
voice="active"
              tense="present" person="third">
      <word>esti</word>
      <predicate                    detail_predicate="predicate_noun"
number="singular">
         <word>dezajno</word>
         <object relation="composition_element" number="singular">
            <word>lingvo</word>
            <attribute detail_attribute="qualitative">
               <word>formala</word>
            </attribute>
         </object>
      </predicate>
   </predicate>
</sentence>
```

Figure 3: A shortened version of the first sentence of this section in E-speranto ("dezajno"="design", "lingvo"="language", "esti"="to be", "formala"="formal")

Tense or person can, for example, be assigned only to the concept belonging to the class *predicate*. The class *subject* has a pre-set semantic relation to the class *predicate* (i.e. the agent or the doer of the action), and the class *object*, for example, allows the usage of semantic relations linking several nominal objects.

class	***interpretation***	***features***
predicate	The concept that represents an action or state.	tense, person, mood, etc.
subject	The concept with the semantic role of the doer of the action (the agent, deep subject).	number, gender, etc.
object	The concept that is involved in the action or state, but is not its doer (the recipient).	number, gender, semantic relation, etc.
adverbial	The concept describes the circumstances of the action or state.	number, semantic relation, etc.
attribute	The concept describes the features of the other concepts.	type, etc.

Table 1: The concept classes in E-speranto

4 Interpreting E-speranto in Natural Languages

Interpreting E-speranto is to a large extent similar to interpreting (and translating) computer languages; however, it has some very important specific features. E-speranto needs to be interpreted in several target languages, while computer languages, on the other hand, are normally translated only into the executable code of the system where program execution will take place. Interpreting into more than one target language requires a strict separation between the procedures and the data needed when interpreting. The procedures are thus an integral part of the interpreters, while the data constitutes the information resources.

4.1.1 The Data Structure

The interpretation procedure begins with the parsing of the source document and the construction of the data structure. It is commonly accepted [Hutchins, 92] that the data structure of an interlingua should clearly indicate the relationships between substructures.

When parsing programming languages, one of the most common approaches is the use of an abstract syntax tree (AST). AST is a tree representation of the syntactic structure of the content in the source language. The tree is abstract due to the fact that it does not encompass all the details of the document in the source language, but merely the ones that reflect the meaning.

Due to the facts that the E-speranto syntax is based on XML syntax and that the structure of a document in this language can be logically represented as a tree, the representation using the AST is suitable also in the context of E-speranto. In addition, the compatibility of E-speranto with XML standards also enables the use of established technologies such as DOM (*Document Object Model*) [DOM, 10] when constructing the data structure. [Fig. 4] shows AST constructed from the content from [Fig. 3]. This AST is used by the prototype interpreter presented in one of the following subsections.

The tree representation is a specialized case of a graph representation, which is especially suitable when the data structure represents simple sentences (such is the case with the prototype interpreter). However, in the future, the representation could change, for example into a graph with cycles (to express the reference of pronouns).

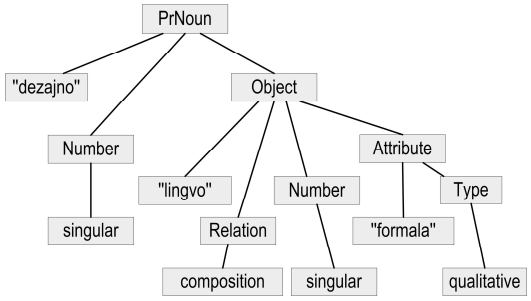

Figure 4: A simplified representation of the noun phrase "design of a formal language" in the form of a tree structure

4.2 Information Resources

From the point of view of information resources, it is not possible to establish a clear boundary between the generation and interpretation processes in the context of E-speranto, as some resources are actually used in both. [Fig. 5] shows the organization of the information resources when generating and interpreting a document in E-speranto. The following sections focus on the resources needed in the interpretation process – both the language-dependent (i.e. the lexicon and the rules of transformational grammar) and the language-independent components (i.e. the E-speranto grammar, vocabulary and real world knowledge).

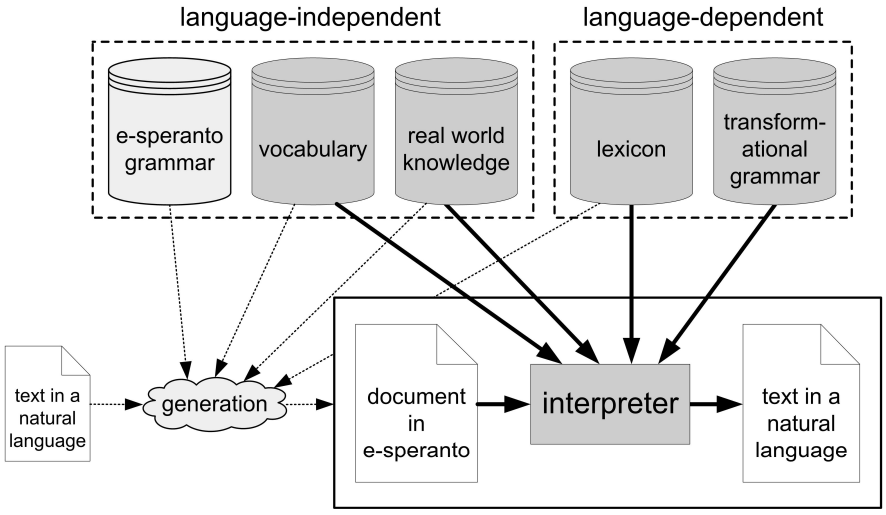

Figure 5: The resources for the creation and interpretation of a document in E-speranto

4.2.1 Lexicon

A lexicon is a compilation of the lexical units in a given natural language and their grammatical features. The lexicon includes:

- the mappings between the lexical units that describe the concepts in E-speranto and their equivalents in a natural language;
- the specific features of lexical units of the target language (e.g. grammatical categories, noun gender, noun number, prepositional case, etc.) and the restrictions in their use that are not encompassed in the general grammatical rules;
- the explanation (description) of the concept represented by the lexical unit in the natural language and the examples of its use. Both lexicon features are especially important when generating a document in E-speranto. With the aid of the development environment, they help the user choose the right lexical units (concepts) from the vocabulary of E-speranto and therefore implicitly determine the meaning of the concepts.

4.2.2 The Rules of Transformational Grammar

This database includes the mappings between the grammatical categories of E-speranto and the target natural language. In addition, the database contains the morphological, syntactical and semantic rules of the target language. These rules define the acceptable combinations of categories and the features of sentence elements that constitute the interpretation of a document in E-speranto. [Fig. 6] shows an example of a rule in the prototype interpreter presented in the following sections.

```
defaultOrder:={
   Cases[element,_xSubject],
   Cases[element,_xAux],
   Cases[element,_xMod],
   Cases[element,_xMain],
   Cases[element,_xPrNoun],
   Cases[element,_xAtt],
   Cases[element,_xObject],
   Cases[element,_xAdv]
 };
```

Figure 6: An example of a rule defining the order of individual sentence elements in the prototype interpreter. The rules are implemented on the basis of pattern matching. The interpreter checks whether the data structure matches a specific pattern and then implements the rule related to the pattern

4.2.3 Vocabulary

The E-speranto vocabulary includes all the lexical units in E-speranto. In addition, it also includes the features that often influence the process of interpretation (the so-called selectional restrictions). The latter for example denominate the concreteness[1] of a concept, its "animacy"[2] (i.e. in the case of animals, plants and people), the concept of movement, etc. Some of these features are usually a part of the lexicon; however, as the information included in these features duplicates due to the fact that the same information is recorded in every language, this information is only recorded once in the information resources of E-speranto, i.e. in the vocabulary.

4.2.4 Real World Knowledge

Real world knowledge is a language-independent component containing the information that is not necessarily included in the text or in the E-speranto

[1] A concrete concept denominates an object that actually exists in the real world (for example a book, a table, a computer), while an abstract concept describes the features of the objects that have been extracted from the physical objects (for example love, courage, pride).

[2] The "animacy" of the concept very often defines the grammatical form of a suitable word in the natural language. Slavic languages, for example, often use a different noun case when the noun refers to an animate object (for example an animal or a person, but not, however, a plant) [Stankiewicz, 86, pp. 127-142].

vocabulary, but that is in some cases vital for the correct encoding of the meaning in E-speranto and its subsequent interpretation into a natural language. Real world knowledge is especially important when it comes to generating documents in E-speranto, because it helps to resolve the ambiguities of the natural language or to point to nonsensical meanings. It can also be used in order to improve the quality of interpretation or it can be combined with the interpretation itself for automatic reasoning.

Lately, ontologies in the context of the so-called semantic web [Barners-Lee, 01] have become an important data resource of real world knowledge. Ontologies are the formal representations of the concepts and relations between them within a specific domain. Ontologies include all the knowledge about a domain and thus in a way define the domain.

4.3 The Architecture of the E-speranto Interpreters

The separation of algorithms from the data enables the interpreters to be constructed for many different natural languages by simply "parameterizing" the same algorithms with the linguistic content of the target language. However, this approach is somewhat questionable due to the following reasons:

- It is extremely hard to determine the algorithms general enough to interpret the language structures within the abstract form in any given natural language. Such mappings would need to take into account all the specificities of the target languages.
- Even if such mappings could be defined, the interpretation would include many redundant steps due to the fact that most of the defined procedures would have no influence on the chosen language.
- An interpreter with all of the above-mentioned procedures would be extremely complex and difficult to maintain.
- When interpreting in real time, as is for example the case with web pages, the redundant procedures would cause a longer response time and would thus create a negative user experience.

In order to avoid the aforementioned weaknesses and to make the best of the similarities between individual languages and thus reduce the production cost of the interpreters for these languages, we suggest an approach based on the modular architecture of interpreters and the grouping of modules in layers [Fig. 7]. Individual layers contain modules with the procedures that are carried out on the structures at the same level of abstraction in relation to the target language. In each interpretation phase, the implementation is transferred between the modules in different layers, whereby a specific module at any given layer is in general compatible with several different modules in the following layer. The modules in the first layer are language-independent. At each additional layer, a more language-specific content is processed.

The multilayer architecture can be linked to the abstraction created by separating the common features of a group of languages from the specific ones. Some features of the Slavic subgroup (the linguistic group of the authors of the present paper) are for example:

- *fusional morphology* (Slavic languages tend to create new words by combining several different morphemes);
- *the preservation of case* from the Proto-Indo-European language (most Slavic languages have seven cases);
- *the existence of separate perfective and imperfective verb forms*;
- *a large number of inflections in different parts of speech* (the concord of tense, mood, person, number, gender, case, etc. between different parts of speech);

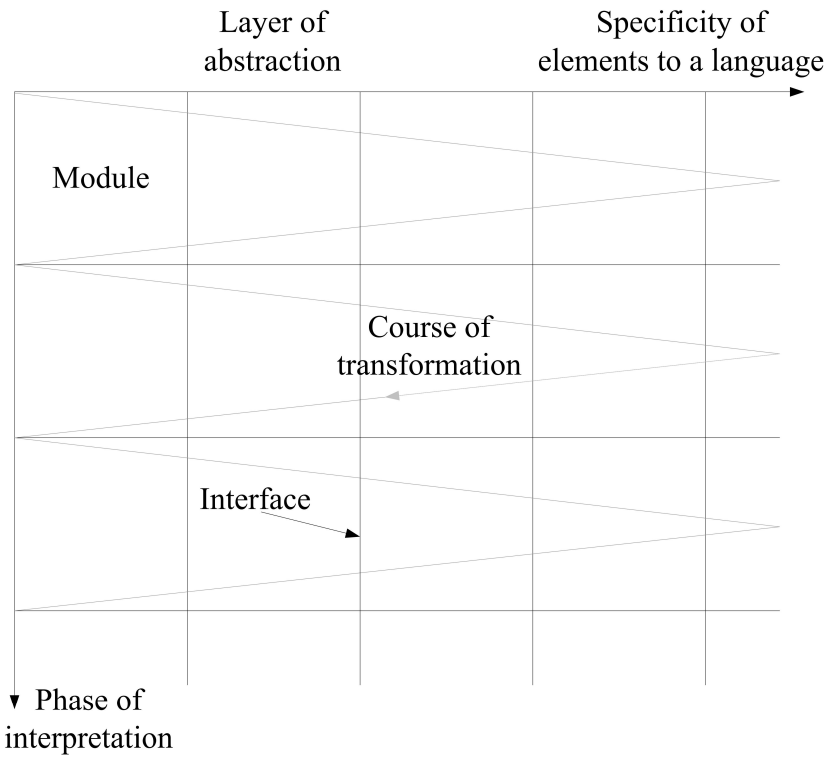

Figure 7: The multilayering of the interpreter based on the abstraction of the structures subject to the procedures within a specific module

The leading language in the world of electronic communication is English, a language belonging to the Germanic subgroup of the Indo-European languages. When comparing the features of the English language to the already mentioned features of the Slavic languages, we can establish the following:

- in English, parts of speech have very few inflections (these are substituted by changing the word order or by using other parts of speech, for example prepositions);
- unlike the so-called synthetic languages (e.g. the Slavic languages) which use morpheme inflection to express different notions, English is an analytic

language, in which individual language units are usually made up of a single morpheme.

When developing interpreters for a group of languages, their common features can be treated at a layer that is separate from the layer with the actual features of specific languages in the group.

5 The Development of the Prototype Interpreter

The above-presented approach to the architecture was used when creating the prototype E-speranto interpreter called INES (*INterpreter of E-Speranto*). We used two programming languages with different programming patterns. The connection of the INES interpreter to the Web was implemented using the object-oriented Java language. The kernel of the interpreter was created in the symbolic programming language *Wolfram Mathematica*, whereby a combination of rule-based programming and symbolic pattern matching was used in addition to the usual interpreter pattern [Gamma, 95, pp 274-288].

Similarly to the interpretation and compilation of programming languages, the interpretation of E-speranto is divided into several stages:

- the lexical and syntactical analyses of the source code,
- the creation of the intermediate code,
- the code optimization and
- the compilation stage.

During the development stage, the lexical and syntactical analyses of the document in E-speranto are covered by the development environment and performed when generating the document. The INES interpreter performs the other functions. The creation of the intermediate code corresponds to the parsing of the E-speranto document into an abstract syntax tree. The code optimization of the classical translators and interpreters corresponds to the adaptation to the AST form specific for the execution environment and the implementation of the interpreter (compare the content in [Fig. 3] and [Fig. 4]). The optimization stage is very important, because it enables the independence of the INES interpreter data structures from the E-speranto grammar and syntax, which are still in the development stage.

The compilation stage is the central part of the interpretation. In the INES interpreter, this stage is divided into three phases: semantic generation, lexical transformation and structural transformation, all of which were already briefly described in the second section of the paper.

The interpretation in these phases is performed by the modules arranged into three layers of abstraction [Fig. 8]. The first layer is comprised of modules that dictate the course of interpretation and are completely independent from the target language. The algorithms in these modules, for example, enable the movement along the tree structure. When traversing the tree structure, individual subtrees are identified according to the class of the root element and transformed by the modules that belong to the next layers.

The second layer consists of the modules closer to the linguistic groups. These modules mostly perform the transformation of individual subtrees. Generally speaking, a subtree corresponds to a sentence element or its part in the sentence of a natural language[1]. The structure presented in [Fig. 4] for example shows a subtree corresponding to a subject complement.

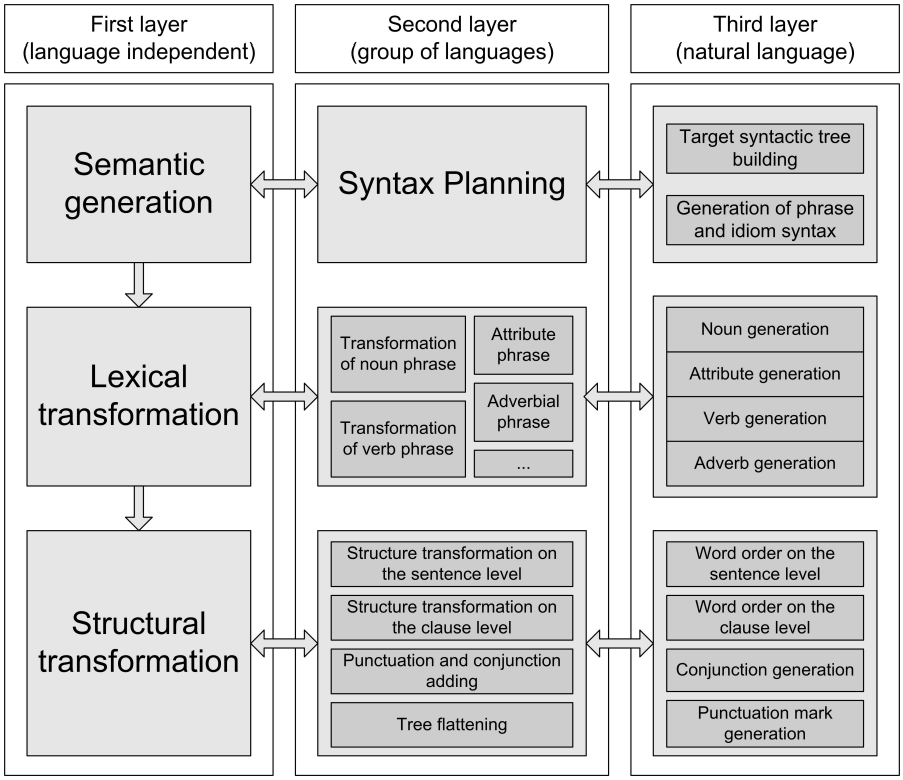

Figure 8: The implementation of the interpreter architecture presented in [Fig. 7] in a prototype interpreter

The modules in the third layer map the parts of the tree structure into the elements of a natural language in a way specific to the target language. Two examples of such a transformation are the substitution of E-speranto concepts and their attributes with the lexical units of the target language or the re-arrangement of the subtrees in accordance with the order of the individual sentence elements in the target language. The access to language specific information resources is also performed in this layer.

[Fig. 9] shows an example of the interpretation of semantic relations connecting the object concepts from E-speranto in English, Slovenian and Serbian. The two relations from E-speranto (*compositionElement* and *recipient*) [Fig. 9a] are expressed

[1] In this sense, sentence elements are analogue to the programming language expressions.

with prepositions *of* and *to* in English [Fig. 9b] and with inflections in Slovenian and Serbian. In Slovenian [Fig. 9c], the suffix –a is added to the root *cvetj* (*cvetja=of flowers*) to express the composition relation, and the root *mat* is supplemented with the suffix –eri (*materi=to (his) mother*) to express the recipient. The inflection mechanism is similar in Serbian [Fig. 9d], only the roots and the suffixes differ.

Due to the similarity in expressing semantic relations in Slovenian and Serbian, their handling is implemented in a common module in the second layer of the INES interpreter. This module is also available to the interpreters of other languages belonging to the Slavic linguistic group. The roots and suffixes suitable for a specific natural language from this group are defined within the modules in the third layer.

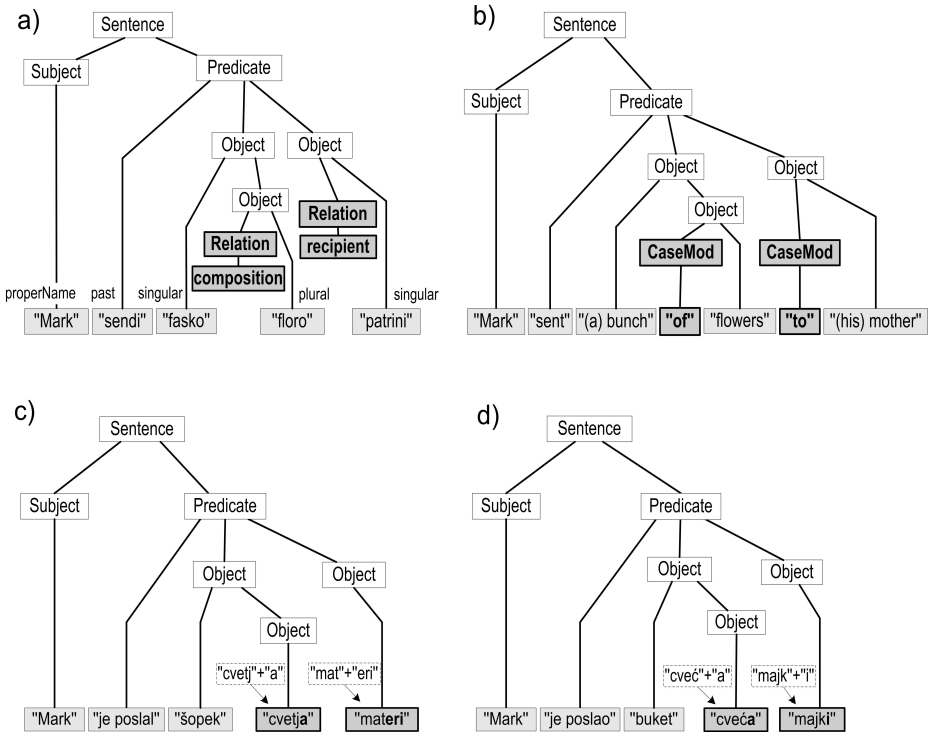

Figure 9: An example of the interpretation of relations involving object concepts in English, Slovenian and Serbian. For convenience, the attributes of the E-speranto concepts are not a part of the tree, but are presented next to the concepts

6 The Proof-of-Concept

The E-speranto interpreters built on the basis of the above presented architecture have been used to build a proof-of-concept of the multilingual Web based on E-speranto[1]

[1] The system can be tested on http://www.e-speranto.org/interpretation-demo/

[Jakus, 10]. Figures [Fig. 10] and [Fig. 11] show the structure and architecture of the system. In light of an easier implementation, we chose to use the interpreters on the server side, as this does not require the standardization of E-speranto and its support by the web browsers. If and when E-speranto becomes standardized, the interpreters will be transferred to the client side, where they will be present in the form of browser plug-ins.

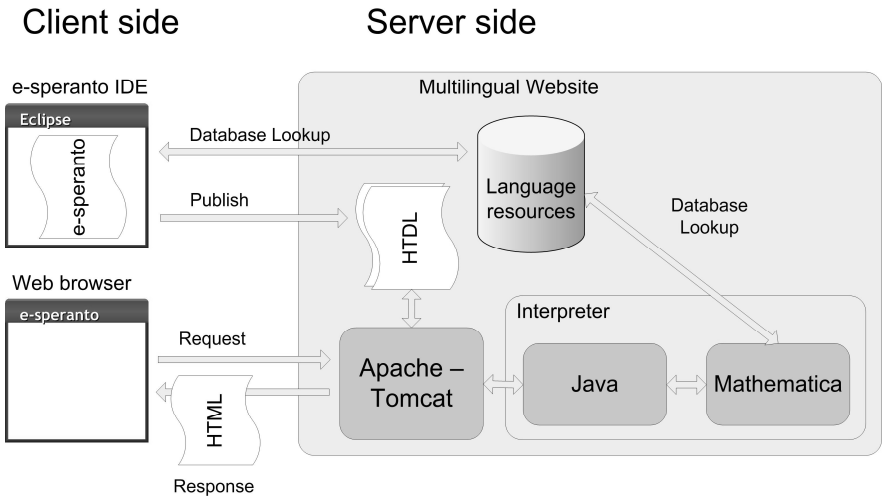

Figure 10: The proof-of-concept of a multilingual web based on E-speranto

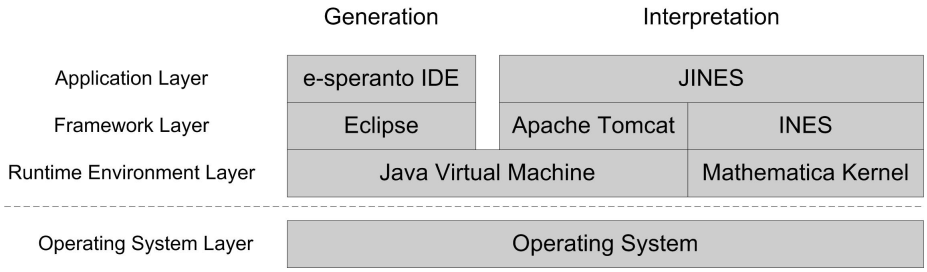

Figure 11: The architecture of the proof-of-concept. The system is built on two operating system-independent runtime environments

The user can create a document in E-speranto with the aid of the development environment and add HTML content. The new format was named HTDL (*HyperText Description Language*) [Fig. 12]. The user can publish such a document on a *multilingual web site*.

When the web server hosting the multilingual web site receives the request for the HTDL document, it forwards the request to the interpreter. The result of the

interpretation substitutes the E-speranto content on the web page. Due to the fact that the server responds with a “clean” HTML document, any web browser can be used as a client. The latter processes the HTML document and displays the web page with the content in the language of the user [Fig. 13].

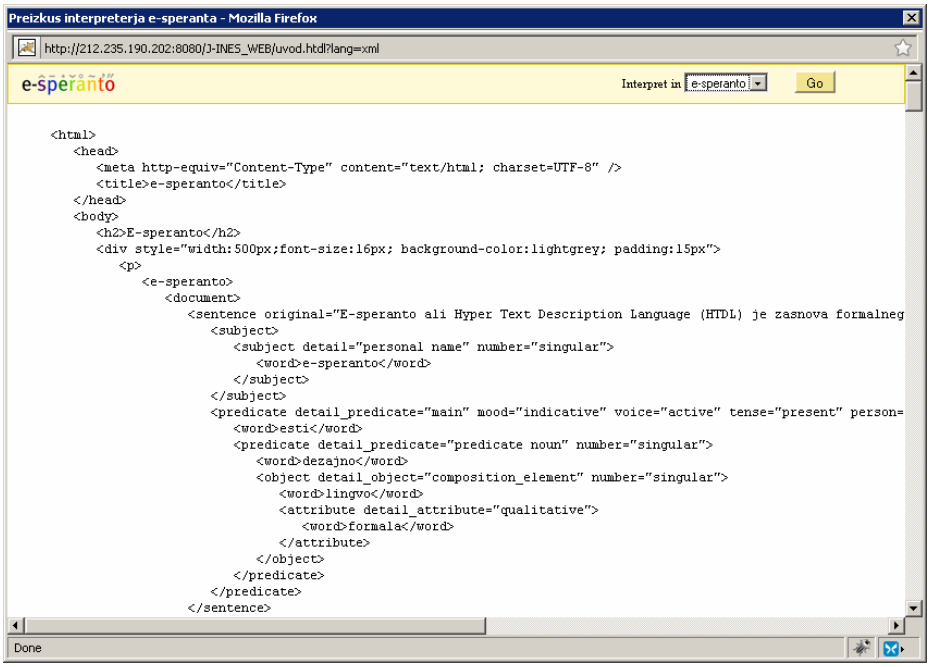

Figure 12: The source code of the web page in HTDL, a combination of HTML and E-speranto

7 Discussion and Further Work

In the implementation of the presented system we used several simplifications which are mostly connected with the information resources and the record in E-speranto:

- we limited the vocabulary of E-speranto,
- we used limited information resources,
- we limited ourselves to the interpretation of E-speranto in Slavic languages,
- we limited ourselves to the interpretation of simple sentences.

In the ideal case, the E-speranto vocabulary should be able to grasp all the meanings in all the natural languages in which we wish to interpret E-speranto, even though a specific meaning only appears in one of the above mentioned languages. The practical application, however, requires several compromises, which means we have to settle for a certain limit of the possible precision in expressing the meaning. This is why we decided to form the basic E-speranto vocabulary on the vocabulary of

Esperanto, although this means that the vocabulary will not match the vocabularies of all the target languages. Consequently, a concept in E-speranto might not have a related meaning in the target language and will have to be expressed in a different way, e.g. with a description, while in some cases it will not even be possible to express the meaning.

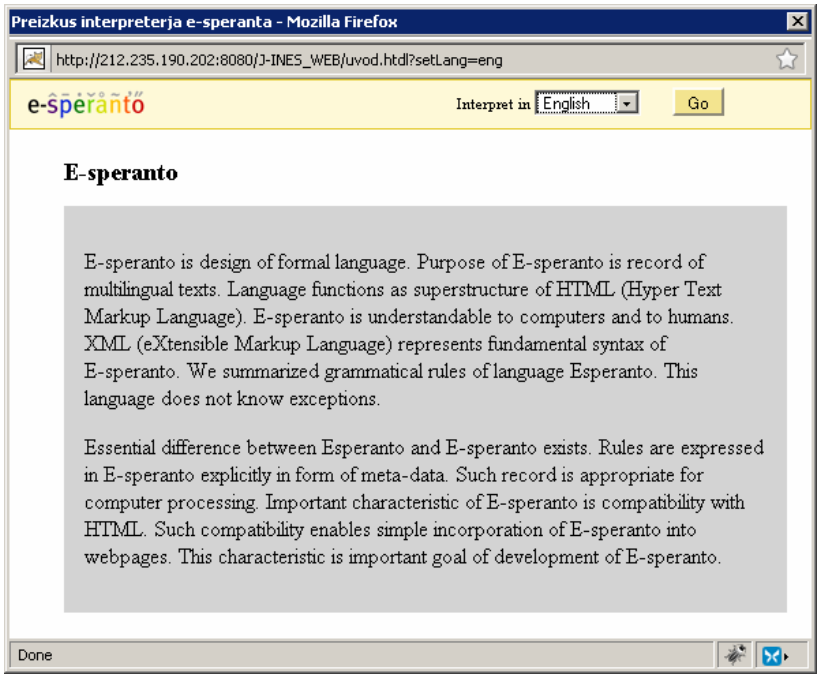

Figure 13: The result of the interpretation of a web page in E-speranto

As the E-speranto vocabulary contains all the most important information about the concepts that are needed for the interpretation, the database with real world knowledge was not yet incorporated in the system. However, the introduction of ontologies, such as, for example, DBpedia [DBpedia, 10], is planned as one of the future improvements of the system.

In addition, we had to settle for a record that can be interpreted into Slavic languages, especially Slovenian, Serbian and Russian. The limitations of this record lie especially in the set of attributes and relations – we focused merely on those that allow the mapping of meaning into the aforementioned languages without a loss in meaning. E-speranto can, with somewhat lower precision, be used for the interpretation in other, especially Indo-European languages[1]. For demonstration purposes, we also regularly develop and update the interpreter in English.

In order to simplify the initial development, we limited ourselves to the interpretation of simple sentences. In this way, the meaning mostly stays intact due to the fact that simple sentences can be used to express almost any meaning. This decision, however, requires that the source text be suitably adapted.

The development of the interpreters is closely related to the development of E-speranto. The results of the interpretation process within the proof-of-concept have given us valuable feedback that can be used when further developing E-speranto. Based on the proof-of-concept we, for example, established that the interpretation needs to take place on a much more abstract level that was first intended. Apparently, if E-speranto mirrors the grammar and syntax of natural languages, the authors often make the mistake of modelling E-speranto records on the records in their mother tongue.

In order for E-speranto to become an established language for the record of multilingual texts on the Web, many issues still have to be addressed – also as regards its interpretation. Within our project, we wish to perform more precise research about the content that will constitute individual layers in the interpreter architecture in order to enable the highest possible reuse factor. In addition, we plan to upgrade the system for the interpretation of multilingual documents with the tools for automatic testing and quantitative evaluation of the interpretation results.

8 Conclusion

One of the biggest obstacles standing in the way of the general use of systems for multilingual communication based on formal languages is the high cost of their production. One of the most important challenges is thus reducing the production cost by reusing the code [Hutchins, 09] or the information resources [Diaz, 00]. The use of E-speranto can greatly contribute in this area, since all it requires for its full functionality is the development of interpreters in individual languages. The interpreter architecture presented in the paper opens new possibilities for additional cost reductions, as it enables the reuse of some modules when interpreting from E-speranto into languages from the same linguistic group.

Although globalization is causing the vanishing of economic and political divides, language and cultural divides still persist. The authors of the paper hope that

[1] An example of lower precision is the absence of articles in the interpretation of the E-speranto document in Fig. 13. Slavic languages do not use articles to express the definiteness of nouns and therefore the attribute that would indicate such a feature was not yet included in E-speranto. Even if such an attribute was included, the rules for the use of articles in English prove to have many exceptions, which would require the incorporation of “real world knowledge” database into the system.

E-speranto and the technologies based on this language will help to overcome these divides in the area where other divides among nations almost do not exist anymore – in the World Wide Web.

References

[Amerio, 02] Amerio, F., Bonvecchiato, G., Fighiera, G. C.: Esperanto: Data and Facts, 2nd edition, FEI - Milan, 2002

[BabelFish, 10] Yahoo! Babel Fish - Text Translation and Web Page Translation, http://babelfish.yahoo.com, accessed 1.4.2010

[Barners-Lee, 01] Berners-Lee, T., Hendler, J., Lassila, O.: The Semantic Web, Scientific American Magazine, May 2001

[Blanc, 00] Blanc, E.: From the UNL hypergraph to GETA's multilevel tree, MT2000: machine translation and multilingual applications in the new millennium, University of Exeter, British Computer Society, November 2000

[Blanc, 02] Blanc, E., Sérasset, G., Tsai, W.: Structural and lexical transfer from an UNL graph to an equivalent natural language dependency tree, LREC-2002: Third International Conference on Language Resources and Evaluation, Workshop: First international workshop on UNL, other interlinguas and their applications, pp.14-18, Las Palmas, Canary Islands, May 2002

[Boitet, 97] GETA's methodology and its current developments, PACLING'97, Meisei University, Ohme, Japan, Proc. pp. 23-57, September 1997

[Busemann, 98] Busemann, S., Horacek, H.: Flexible shallow approach to text generation, Proceedings of the Ninth International Workshop on Natural Language Generation (INLG-1998), Niagara-on-the-Lake, Ontario, Canada, 1998

[Busemann, 02] Busemann, S.: Issues in generating text from interlingua representations, LREC-2002: Third International Conference on Language Resources and Evaluation, Workshop: First international workshop on UNL, other interlinguas and their applications, pp.1-7, Las Palmas, Canary Islands, May 2002

[DBpedia, 10] DBpedia web page, http://dbpedia.org, accessed 1.4.2010

[Dhanabalan, 03] Dhanabalan, T., Geetha, T. V.: UNL Deconverter for Tamil, Convergences ´03, International Conference on the Convergence of Knowledge, Culture, Language and Information Technologies, Alexandria, Egypt, 2003

[Diaz, 00] Diaz de Ilarraza, A., Mayor, A., Sarasola, K.: Reusability of wide-coverage linguistic resources in the construction of a multilingual machine translation system, MT2000: machine translation and multilingual applications in the new millennium: international conference at the University of Exeter, November 2000

[DOM, 10] Document Object Model, W3C, http://www.w3.org/XML/Schema, accessed 1.4.2010

[Freigang, 86] Freigang, K-H.: Research on machine translation at the University of Saarbrücken, Translating and the Computer 8, Conference proceedings, ed. Catriona, Picken, London, 1986

[Gamma, 95] Gamma, E., Helm, R., Johnson, R., Vlissides, J.: Design Patterns: Elements of Reusable Object-Oriented Software, Addison-Wesley, 1995

[GoogleTranslate, 10] Google Translate, http://translate.google.com, accessed 1.4.2010

[Hutchins, 92] Hutchins, W., Somers, H.: An Introduction to Machine Translation, Academic Press, New York, 1992

[Hutchins, 09] Hutchins, W.: Uses and Applications of Machine Translation, Presentation on 20 February 2009 at Westminster University

[IWS, 10] Internet World Stats, http://www.internetworldstats.com, accessed 1.4.2010

[Jakus, 10] Jakus, G., Varga, E., Tomažič, S.: Interpretation of multilingual documents in e-speranto using the client-server architecture model, The IPSI BgD Transactions on Advanced Research, IPSI BgD Internet Research Society, Vol.6, No.1, pp. 11-16, January 2010

[Muraki, 87] Muraki, K.: PIVOT: Two-Phase Machine Translation System, Machine Translation Summit, Hakone Prince Hotel, Japan, in MT Summit Manuscripts and Program pp. 81-83, September 1987

[Nyberg, 92] Nyberg, E., Mitamura, T.: The KANT system: Fast, accurate, high-quality translation in practical domains, COLING, 1992

[Omerović, 07] Omerović, S., Jakus, G., Filimonova, T., Tomažič, S.: Multilingual documents in e-speranto (in Slovenian), Electrotechnical Review, Vol. 74, No. 4, pp. 151-157, 2007

[Paul, 09] Paul, L. M. (ed.): Ethnologue: Languages of the World, Sixteenth edition. Dallas, Tex.: SIL International, 2009

[Promt, 10] PROMT Translation Software and Dictionaries, http://www.promt.com, accessed 1.4.2010

[Rosetta, 94] Rosetta, M. T. (pseud.): Compositional Translation, Dordrecht: Kluwer Academic Publishers, 1994

[Schubert, 88] Schubert, K.: The Architecture of DLT – interlingual or double-dialect, New Directions in Machine Translation, Floris Publications, Holland, 1988

[Sérasset, 00] Sérasset, G., Boitet, C.: On UNL as the future "html of the linguistic content" & reuse of existing NLP components in UNL-related applications with the example of a UNL-French deconverter, COLING, August 2000

[Singh, 07] Singh, S., Dalal, M., Vachhani, V., Bhattacharyya, P., Damani O. P.: Hindi Generation from Interlingua, MT Summit XI, Proceedings, pp. 421-428, Copenhagen, Denmark, September 2007

[Spall, 07] Spall, S. S., Bhatia, P.: UNL Punjabi Deconverter, LRIL-2007 National Seminar on Creation of Lexical Resources for Indian Language Computing and Processing, Centre for Development of Advanced Computing (C-DAC), Mumbai, India, 2007

[Stankiewicz, 86] Stankiewicz, E.: The Slavic Languages: Unity in Diversity, Mouton De Gruyter, November 1986

[Systran, 10] SYSTRAN - Online translation, translation software and tools http://www.systranet.com, accessed 1.4.2010

[Temizsoy, 98] Temizsoy, M., Cicekli, I.: A language-independent system for generating feature structures from interlingua representations, Proceedings of the Ninth International Workshop on Natural Language Generation (INLG-1998), Niagara-on-the-Lake, Ontario, Canada, 1998

[Thomason, 74] Thomason, R. (ed.): Formal Philosophy. Selected Papers by Richard Montague, New Haven, 1974

[Tomažič, 07] Tomažič, S.: Multilingual Web with e-speranto, The IPSI BgD Transactions on Internet Research, IPSI BgD Internet Research Society, Vol.3, No.2, pp 13-15, July 2007

[Uchida, 89] Uchida, H.: ATLAS II: A Machine Translation System Using Conceptual Structure as an Interlingua, in Proceedings of the Second Machine Translation Summit, Tokyo, 1989

[Uchida, 99] Uchida, H., Zhu, M., Della Senta, T.: Universal Networking Language: A gift for a millennium, The United Nations University, Tokyo, Japan, 1999

[XML, 10] Extensible Markup Language (XML), W3C, http://www.w3.org/XML, accessed 1.4.2010

[XMLS, 10] XML Schema, W3C, http://www.w3.org/XML/Schema, accessed 1.4.2010