# CLAU – A Service-Oriented System for Complex Language Alignment: Architectural Aspects

Claudiu Mihăilă, Corina Forăscul, and Sabin C. Buraga

*Abstract*—**In the last years, parallel corpora have become an effective framework to study how well the linguistic phenomena and, more specifically, annotation schemata can be applied when importing the annotations from one language to the other(s). In the case of automatic import, the evaluation and correction are better to be performed by linguists using specific software. The paper proposes CLAU – a service-oriented interactive application allowing users to import, evaluate, correct, and share XML-based annotations in parallel texts. The design, general architecture, and implementation are discussed. Also, two use cases are presented: temporal annotations in parallel texts and how CLAU facilitates social Web interactions between language scientists.**

***Index Terms***—**Parallel text processing, cross-language studies, service-oriented architecture.**

## I. Introduction

DUE to their extensive use in many NLP applications, in comparative language study, and their importance in language industries, parallel corpora are continuously created, improved and exploited [20]. Linguistic resources in a target language can be easier developed based on the linguistic knowledge (annotations) encoded in a source text, and using multilingual technologies – such as alignment at various text levels and annotation transfer –, provided that the appropriate text preprocessing tools exist for the source language. In a multiple language setting, it is easier and cheaper to correct automatically imported annotations than to create them from scratch. The individual language users should then use either a set of language specific rules, or directly to manually correct the annotations imported from the source language. Recent experiments with transferring annotations – word senses, syntactic and semantic relations, collocations [18], temporal information [9], coreference chains [15] – show that word alignment technology is a successful solution for the transfer of information from the tokens in one language to their translation equivalents in the other language.

Based on the existing NLP tools, briefly presented in Section II, *CLAU – the Complex Language-Alignment User-oriented system* – will be presented, available as an open-source collection of Web services working with XML-annotated parallel corpora. The users can configure CLAU according to their needs when developing and/or correcting parallel annotations, and they can collaborate in a cross-lingual and cross-cultural environment.

The paper describes – in Sections III and IV – the design, the general architecture, and several implementation solutions of the proposed system. CLAU is developed as an interactive application allowing users to import, correct, and evaluate annotations in parallel texts. The CLAU parallel text alignment system will use certain techniques implemented as local or external Web services, following the Service-Oriented Architecture (SOA) methodology described in Section 3 of this paper.

Section V presents two case studies proving how well the annotator can handle and/or improve annotations in parallel texts when using CLAU. These case studies show the impact of using a system like CLAU in the activities related to the annotations of temporal information in parallel texts, and in the support for social Web interactions. By using CLAU, less-specialized users of NLP tools are able to surmount different difficulties and redundant work regarding parallel corpora.

The paper ends by presenting the main conclusions and also gives important future research directions.

## II. State of the Art

Next to the abundance of linguistic resources, there are many specific tools for their exploitation and use [20]. Depending on the type of resource to work with – be it monolingual, parallel or comparable corpora, lexicons or dictionaries of various types, the exiting tools have different characteristics:

- Can be used locally or remotely (through a server),
- Can accept various file-formats or annotation standards,
- Can permit or not a collaborative and/or cross-lingual work,
- Can or cannot be configured according to the user's needs,
- Can have different types of licenses, which have impact on their use and further development by other members of the research community.

Manuscript received October 21, 2008. Manuscript accepted for publication February 19, 2009.

Claudiu Mihăilă is with Faculty of Computer Science, “Al.I. Cuza” University of Iaşi, 16, General Berthelot, 700483, Iaşi, Romania (e-mail: claudiu.mihaila @ info.uaic.ro).

Corina Forăscu1 is with Faculty of Computer Science, “Al.I. Cuza” University of Iaşi, 16, General Berthelot, 700483, Iaşi, Romania and with the Romanian Academy Research Institute for Artificial Intelligence, Romania (e-mail: corinfor@ info.uaic.ro).

Sabin C. Buraga is with Faculty of Computer Science, “Al.I. Cuza” University of Iaşi, 16, General Berthelot, 700483, Iaşi, Romania (e-mail: busaco@ info.uaic.ro).

*CLaRK*[1], constantly developed since 2001 [17, 12], is an XML-based software system for the development of linguistic resources such as: corpora, dictionaries, and ontologies. Concerning the corpora (monolingual or multilingual), CLaRK permits editing, manipulation, searching, and transforming them, with a minimum human intervention. The system, implemented in Java, can be used only locally, hence a collaborative work is not (yet) possible. CLaRK can be configured to suit user needs. Conceived for working in a monolingual environment, CLaRK incorporates facilities for working with parallel texts, provided that the parallel annotations already exist.

One newer set of NLP tools, offering support for less-studied languages, is available remotely as Web service[2] at the RACAI[3] Institute. The linguistic Web services for English and Romanian [19] implement essential NLP operations such as POS tagging (with its prerequisites sentence and token splitting), lemmatization, chunking, word linking, WordNet lookup, languages identification, diacritics insertion (for Romanian) and Romanian Wikipedia indexing and searching. The access to the web services is research license-based.

*SAM* – the Script Annotation Manager [10] is an interactive system allowing the user to annotate and process parallel texts, XML encoded, based on a common vocabulary. Implemented as a plug-in for the open source Eclipse platform, SAM provides means to select parts of the text, then label this selection with an appropriate category and optionally enter a description for this annotation.

*GATE*[4] (General Architecture for Text Engineering) [5, 6] is a free open-source[5] architecture, framework and development environment for creating, evaluating and embedding NLP software and XML resources. It can integrate a large amount of built-ins in new processing pipelines that can be put to work on single documents or corpora. The user is instructed to select the GATE resources (modules) as parts of the needed processing chain, which then works on an input file, and returns an XML annotated output file. Developed since 1995, it includes support for monolingual manual annotation, performance evaluation, information extraction, (semi)automatic semantic annotation, and many other tasks. GATE is integrated with other broadly-used NLP software, such as UIMA, Wordnet, Weka, Lucene, Sesame, and Minipar.

*ALPE* – Automated Linguistic Processing Environment – [4] is a meta-system for dynamical building of NLP architectures. The model used by the system is a hierarchy of XML annotation schemas in which the parent-child links are defined by subsumption relations. The hierarchy is augmented with processing power by marking the edges with names of processors, each realizing an elementary NL processing step, able to transform the annotation corresponding to the parent node onto that corresponding to the child node. Still in development, ALPE allows a user to automatically utilize existing processing paths or to configure new ones on-the-spot, by exploiting the annotation schemas at intermediate steps.

Together with *UIMA* [8], the last two systems are going to be used together or separately[6] in the CLARIN[7] pan-European project, a large-scale collaborative effort to create, coordinate and make language resources and technology available and readily useable especially in the domains of Humanities and Social Sciences.

From the above systems, SAM is targeted for parallel corpora, but it does not take into account the possible alignments between the two texts, and CLaRK gives the possibility to work with parallel corpora. All the others could be used in CLAU for obtaining the initial annotations, and then, based on the alignment, CLAU enhances the (semi)automatic transfer of the annotations between individual language files.

## III. System Design

The main purpose of the CLAU system is to reduce the superfluous work, especially of linguists, when analyzing, evaluating or correcting annotations in parallel corpora.

The CLAU system is built, like GATE, following the Model-View-Controller architectural pattern, in order to isolate the business logic from the user interface, thus resulting in an application where it is easier to modify either the visual appearance of the application or the underlying business rules without affecting the other.

The architecture of CLAU, depicted in Fig. 1, aims to be modular and service-oriented (see Section 3.1). By following this approach, it is feasible to externalize some services, which already exist, like persistent storing, collaborative services, text preprocessing, machine translation, etc. Another advantage is that it is easier to add new services to the system once they are necessary. Plus, a service-oriented architecture ensures a high degree of cross-platform interoperability.

The data to be processed – in this case, the texts to be annotated – are represented in an internal model. Views and editors are accessing the model via a controller class. Changes within the application are propagated by the Event Dispatching Thread, a Java internal mechanism, to the model and to the view.

The core components included in the CLAU system are the XML Synchronization, Automatic file Alignment, Statistics, Storage, and User Interface modules. The design of each module is discussed in the following subsections.

[1] http://www.bultreebank.org/clark/

[2] http://nlp.racai.ro/webservices/

[3] Romanian Academy Research Institute for Artificial Intelligence: http://www.racai.ro/

[4] http://gate.ac.uk

[5] http://sourceforge.net/projects/gate

[6] http://www.clarin.eu/events/web-services-architecture-in-clarin

[7] http://www.clarin.eu/

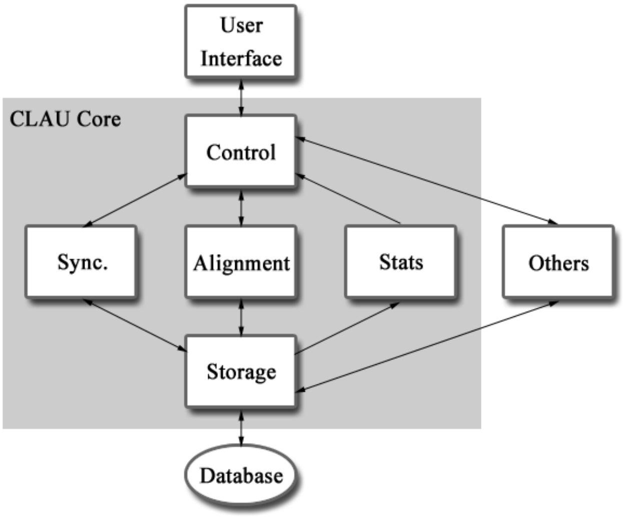


Fig. 1. The CLAU general architecture, showing the bi- or unidirectional interconnections between modules.

### A. *Service Oriented Architecture*

SOA (Service Oriented Architecture) refers to the design of a complex distributed system. SOA is a design methodology, aimed at maximizing the reuse of multiple services – possibly implemented on different platforms and using multiple programming languages. In a SOA-based context, the services generally have some important characteristics [7, 11]:

- Services are individually useful – they are autonomous (for example, a specific alignment algorithm is implemented by a service that can be independently invoked);
- Services must be loosely coupled – services discover the needed information at the time they need it. The benefits offered by this characteristic are: flexibility, scalability, ability to be easily replaced, and fault tolerance;
- Services can be composed to provide other services. This promotes the reuse of existing functionality (in our case, the alignment service is a composite service);
- Services can participate in a workflow. An operation performed by a service will depend on the messages that are sent or received – this aspect means service choreography. In our context, there are many examples of workflows, especially concerning editing and the synchronization of text annotations;
- Services can be easily discovered, eventually in an automatic manner. Therefore, services must expose details (and additional meta-data) such as their capabilities, interfaces, policies and supported protocols. Other details such as the programming language or the information about the platform are not useful for consumers and – usually – are not revealed.

Using SOA architecture is beneficial for our system since it allows easily adding new features without modifying the existing ones. Because these are based on existing services, the code reuse is maximal, and then development and testing time is minimal.

### B. *User Interface Module*

The CLAU user interface (as in Fig. 2) was conceived to be most advantageous in the process of annotating parallel texts. The system allows the creation of more editors at the same time. The user may open, view and modify two or more, up to ten, texts in different languages at once. This limit was imposed in order for the user to be able to read easily the contained texts. The text editors are responsible for presenting the textual data, while the tables in the lower part show the markup of the selected word, in the form of attribute-value pairs.

The interactive factor of this application is represented by many context-sensitive popup menus, integrated in various visual components. Through this mechanism, advanced users may benefit from the ergonomics of the designed interface.

On the one hand, the text editors are synchronized with the annotation views, in the lower part. As soon as the user moves the caret from one word to another, the view refreshes the information regarding the annotation of the current word. We have decided to remove the annotations from the exposed text, ensuring that it is more readable.

On the other hand, the text editors are synchronized with each other. The movement of the caret in one editor triggers the automatic selection of the corresponding word in the other editors.

Also, a graphical representation of the word alignment between two texts is offered to simplify the modifications that may occur. The user is able to add, delete or update a connection between words.

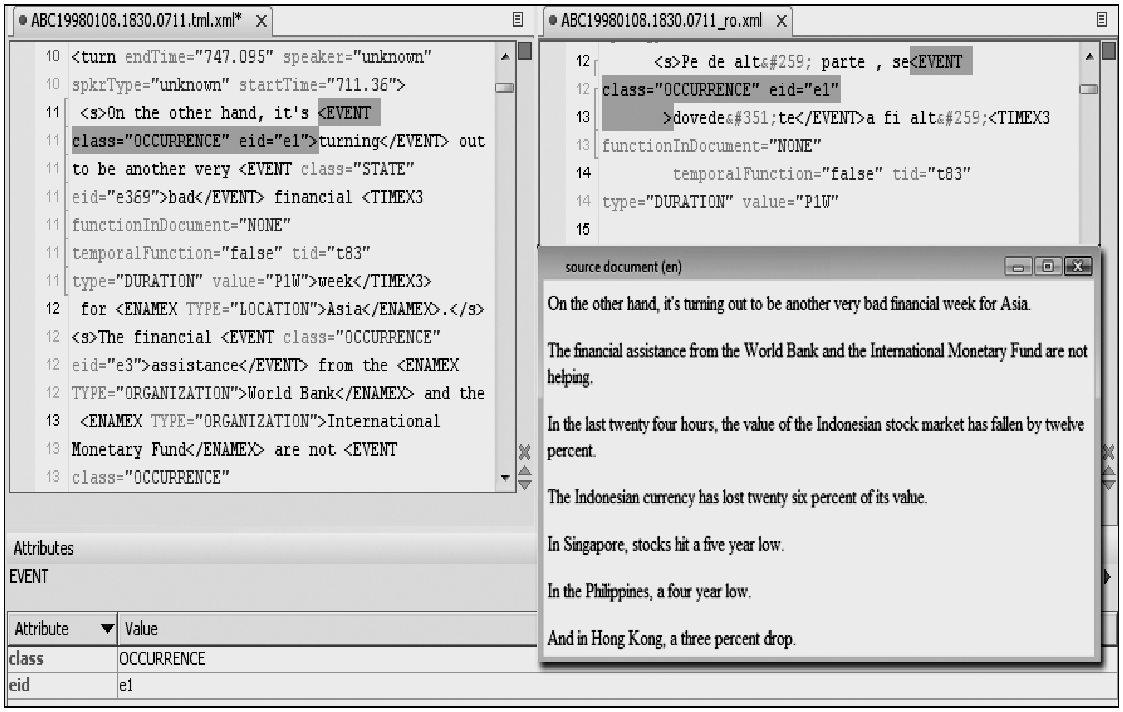
We are using the Eclipse platform [16] which also offers facilities for the XML editing and validating, along with visual indicators, showing the corresponding begin/end markup, attribute auto-complete option, should an annotation schema (e.g., DTD or XML Schema) be used.

### C. *XML Synchronization Service*

By XML synchronization we understand the fact that any alteration of a file in the parallel corpus leads to the automatic alteration of the corresponding entity in the other file(s) of that corpus. Also, the annotations in one file may be exported to the corresponding plain text file in the other language(s), based on the word alignment between the two texts.

For the XML synchronization module, we have considered to take advantage of the Eclipse platform, too.

Based on the word alignment, a correspondence between the tags of different annotated texts is automatically created. There are cases when this correspondence between tags is not a bijective function, and such cases are reported as problematic situations, to be eventually corrected by the user(s). Should it be just a linguistic phenomenon (e.g. untranslatable words or phrases) or simply a mistake (missing or incorrect translations), the users are free to take any decision – they may let the challenging situation as it is, or write the same sentence/fragment differently, where achievable.



```
<turn endTime="747.095" speaker="unknown"
spkrType="unknown" startTime="711.36">
 <s>On the other hand, it's <EVENT
class="OCCURRENCE" eid="e1">turning</EVENT> out
to be another very <EVENT class="STATE"
eid="e369">bad</EVENT> financial <TIMEX3
functionInDocument="NONE"
temporalFunction="false" tid="t83"
type="DURATION" value="P1W">week</TIMEX3>
 for <ENAMEX TYPE="LOCATION">Asia</ENAMEX>.</s>
<s>The financial <EVENT class="OCCURRENCE"
eid="e3">assistance</EVENT> from the <ENAMEX
TYPE="ORGANIZATION">World Bank</ENAMEX> and the
 <ENAMEX TYPE="ORGANIZATION">International
Monetary Fund</ENAMEX> are not <EVENT
class="OCCURRENCE"
```

```
     <s>Pe de alt&#259; parte , se<EVENT
class="OCCURRENCE" eid="e1"
      >dovede&#351;te</EVENT>a fi alt&#259;<TIMEX3
functionInDocument="NONE"
       temporalFunction="false" tid="t83"
type="DURATION" value="P1W"
```

On the other hand, it's turning out to be another very bad financial week for Asia.

The financial assistance from the World Bank and the International Monetary Fund are not helping.

In the last twenty four hours, the value of the Indonesian stock market has fallen by twelve percent.

The Indonesian currency has lost twenty six percent of its value.

In Singapore, stocks hit a five year low.

In the Philippines, a four year low.

And in Hong Kong, a three percent drop.

| Attribute | Value |
|---|---|
| class | OCCURRENCE |
| eid | e1 |

Fig. 2. CLAU screenshot showing the synchronization between the two XML editors (English and Romanian languages), the annotation attributes in the lower part, and the plain text.

### D. Automatic Alignment Service

The purpose of the automatic alignment service is to analyze the parallel texts and align them at various levels – paragraph, sentence, and/or word level, with a high level of accuracy. Of course, there are cases when words in one language cannot be aligned to words in another language, or where one word corresponds to two or more words at the same time. It is also likely to come across multiword expressions which, due to their cross-lingual structural differences, are the hardest to translate by a simple word-by-word approach. These are usually language and culture specific, or are part of a very specialized jargon.

The alignment service is a composite service which consists of several modules implementing specific alignment algorithms. Each such a module can be a standalone piece of software – most liable to future improvement –, implemented in a platform-independent language to increase portability. One possible tool that is prone to be used by CLAU is *GIZA++*[8], described in [13].

Also, the evaluation in the field of multilingual alignment, carried out within the ARCADE II project[9] [3], indicates the main alignment systems which have superior results, hence candidates for integration in CLAU.

### E. Statistics Module

The operations performed by the users are automatically recorded by the statistics module. Thus, an analysis of the human work is produced, showing the activity of specific user(s), be it linguistic/annotation tasks, or interaction through the CLAU environment with other users of the system.

There are three categories of statistics this module provides:

- All operations on files are recorded when a user works on two parallel texts: the performed actions like adding, modifying or deleting annotation tags or even text are traced and kept in log files. Also, the languages the user usually acts upon are added to the user profile, to be used when working in the CLAU collaborative setting.
- The traditional evaluation measures like precision, recall, and F-measure are available whenever one of the texts is (part of) a golden corpus.
- Inter-annotator agreement and collaborative degree are computed if two or more users are working on the same files. Through the use of such pieces of information, the system is able to create relationships between the users, based on the collaborative degree. Moreover, by using this kind of data, the users may relate one to another for exchange of opinions or solving different unforeseen issues.

Based on the output of this module, an experimented user can do further analysis and/or statistics to be used in research directions not (yet) included in the CLAU system.

### F. Storage Service

The storage module represents the link between the application and the database. It provides access to stored files for the XML synchronization, automatic alignment and statistics modules. The communication is bidirectional in the case of the XML synchronization and alignment modules, as

[8] http://www.fjoch.com/GIZA++.html

[9] http://sites.univ-provence.fr/veronis/arcade/index.html

they are influenced one by the other, and unidirectional in the case of the statistics module (from storage to statistics).

Options, such as relational database systems, XML-enabled, or native XML database systems, are feasible storage solutions for CLAU. Another suitable possibility – especially in the context of Linked Open Data initiative [1] – is to store information into an RDF triple store.

## IV. Implementation

A solution for the implementation of core CLAU services is the Java language that offers platform independency, compatibility with the XML documents management libraries (e.g., the native XML database[10], Saxon[11] XML processing library), and moderate requirements for installed interpreters on the client machine.

CLAU is implemented as a plug-in for the open source Eclipse platform[12]. The Eclipse workbench provides a robust extensible software infrastructure, which makes it possible to efficiently design and implement linguistic applications for various purposes. Additionally, the excellent XML editing and management facilities provided by the oXygen system[13] can be easily used as an Eclipse plug-in.

## V. Use Cases

There are many situations where a system like CLAU can improve the quality of the evaluations in parallel texts; therefore the time needed for such activities will significantly decrease. Moreover, being an interactive environment, it also enhances the social relationships between users. The situations described below provide evidence for the CLAU benefits.

### A. Temporal Information in Parallel Texts

As mentioned in the introduction, there are experiments involving the automatic transfer of temporal mark-up from English to Romanian [9]. The evaluation of the automatic import was initially performed manually, using directly the annotated files in the two languages: the evaluator browses separately the two files and corrects separately the errors in each file.

The use of an editor, like CLaRK, can improve the quality of the evaluation, by reducing the time needed for such an activity, and by ensuring a better coverage of the annotations that are to be evaluated – one can expect a user can skip or forget to evaluate some annotations.

The evaluation/correction activity is improved even more by using CLAU: next to parallel browsing, it gives the possibility to automatically transfer a modification in one file into the other. Moreover, all problematic situations of the automatic transfer are indicated to the user, hence checking and, eventually, modifying the annotations in the target language are directly performed.

If more than one user is working on the same corresponding files, the evaluations also show the agreement among the users, thus indicating the degree of applicability of the general temporal theory to another language than the one initially used for the development of the theory/annotation schema.

### B. Support for the Social Web-like Interactions

Another important facet of the proposed system is given by the support for social interactions, conforming to the so-called “Web 2.0” – or social Web – characteristics [14]. These “Web 2.0”-like approaches are not easily found in the existing implementations of similar tools.

The actual Web can be viewed as a platform that gives users the possibility/liberty to control their data. CLAU offers support for ad-hoc creation of a social community by viewing a specific alignment as a social object, in the same sense as photos or Web addresses act. Each performed/edited alignment can be set to be shareable among (groups of) users. The CLAU system can connect similar groups of scientists on the basis on their user profiles – e.g., interests, geographical location, known languages, etc. – and/or regarding the same activities executed within our application – for instance, same texts to be aligned or similar annotations written. Such recommendations could be performed by using classical machine learning techniques.

Additionally, tag-based facilities for identification, searching, classification, and aggregation of alignments are provided.

## VI. Conclusions and Further Work

In this paper, CLAU – a service-oriented system which facilitates the complex language-alignment processes – was proposed. The overall architecture of the application was detailed, following the principles of the Service Oriented-Architecture methodology for developing complex software. The design consists of several important core modules and was presented in the section 3. Also, different solutions of effective implementation were provided. Our approach is focused on open source technologies, including greater flexibility and usability of the developed system.

Several key use cases were also included, to prove certain facilities provided by the CLAU application, in order to augment the work of the language annotators.

Future immediate research directions, illustrated in Fig. 1 as the *Others* module, include the integration of other NLP tools, like those mentioned in section 2, which might be of use when annotating parallel corpora, the complete alignment to international standards and formalization of communication methods between services, and the extension and creation of new ones, on a collaborative basis.

Machine translation is of the utmost importance when dealing with multilingual parallel corpora; therefore, the integration in CLAU of such a service might be necessary in the future, especially with the increase in the number of languages in current corpora. The user(s) will then evaluate

[10] http://www.rpbourret.com/xml/XMLAndDatabases.htm

[11] http://saxon.sourceforge.net/

[12] http://www.eclipse.org/

[13] http://www.oxygenxml.com/

and correct both the translation, as well as the parallel annotations.

A more formal study on the subject of architectural aspects, especially concerning the external services that can be integrated, could be considered for the next stage development. We are planning to describe every operation that can be performed in terms of standard WSDL[14] documents, and to create the subsequent workflows regarding the most important activities within CLAU system.

Another important direction to follow is towards collaborative recommending: the proposed application can automatically correlate two text-alignments based on the detected languages, and can use these correlations to improve its recommendations – using collaborative filtering [2] or association rules.

## REFERENCES

[1] Bizer, C., Heath, T., Idehen, K., Berners-Lee, T., "Linked Data on the Web," in *Proceedings of WWW2008*, Beijing, China, ACM Press, 2008.

[2] Chakrabarti, S., *Mining the Web – Discovering Knowledge from Hypertext Data*. Morgan Kaufmann, San Francisco, 2003.

[3] Chiao, Y.-C., Kraif, O., Laurent, D., Nguyen, T.M.H., Semmar, N., Stuck, F., Véronis, J., Zaghouani, W., "Evaluation of multilingual text alignment systems: the ARCADE II project," in *Proceedings of LREC-2006*, Geneva, 2006.

[4] Cristea, D., Forăscu, C., Pistol, I., "Requirements-Driven Automatic Configuration of Natural Language Applications," in Bernadette Sharp (Ed.): *Natural Language Understanding and Cognitive Science, Proceedings of the 3rd International Workshop on Natural Language Understanding and Cognitive Science – NLUCS 2006, in conjunction with ICEIS 2006*, Cyprus, Paphos. INSTICC Press, Portugal, 2006.

[5] Cunningham, H., Maynard, D., Bontcheva, K., Tablan, V., "GATE: A Framework and Graphical Development Environment for Robust NLP tools and Applications," in *Proceedings of the 40th Anniversary Meeting of the Association for Computational Linguistics (ACL'02)*, Philadelphia, 2002.

[6] Cunningham, H., Tablan, V., Bontcheva, K., Dimitrov, M., "Language engineering tools for collaborative corpus annotation," in *Proceedings of Corpus Linguistics Conference*, Lancaster, UK, 2003.

[7] Erl, T. *Service-Oriented Architecture: Concepts, Technology, and Design*. Prentice Hall PTR, 2005.

[8] Ferrucci, D., Lally, A., "UIMA: an architectural approach to unstructured information processing in the corporate research environment," *Natural Language Engineering* 10, No. 3-4 (2004)

[9] Forăscu, C., "Why Don't Romanians Have a Five O'clock Tea, Nor Halloween, but Have a Kind of Valentine's Day?" in A. Gelbukh (Ed.): *Computational Linguistics and Intelligent Text Processing, CICLing 2008, LNCS 4919*, Springer-Verlag, Berlin Heidelberg, 2008.

[10] Geilfuss, M., Milde, J.-T., "SAM – an annotation editor for parallel texts," in *Proceedings of LREC-2006*, Geneva, 2006.

[11] Josuttis, N., *SOA in Practice. The Art of Distributed System Design*. O'Reilly, Sebastopol, 2007.

[12] Monachesi, P., Simov, K., Mossel, E., Osenova, P., Lemnitzer, L., "What Ontologies Can Do for eLearning," in *Proceedings of IMCL 2008*, Amman, Jordan, 2008.

[13] Och, F. J., Ney, H., "A Systematic Comparison of Various Statistical Alignment Models," *Computational Linguistics*, Vol. 29, No. 1, 2003.

[14] O'Reilly, T., *What is Web 2.0 – Design Patterns and Business Models for the Next Generation of Software*. O'Reilly, Sebastopol, 2005.

[15] Postolache, O., Cristea, D., Orăsan, C., "Transferring Coreference Chains through Word Alignment," in *Proceedings of LREC-2006*, Geneva, 2006.

[16] Shavor, S., Fairbrother, S., D'Anjou, J., Kehn D., *Eclipse.* Addison-Wesley Professional, 2004.

[17] Simov, K., Peev, Z., Kouylekov, M., Simov, A., Dimitrov, M., Kiryakov, A., "CLaRK – an XML-based System for Corpora Development," in *Proceedings of the Corpus Linguistics 2001 Conference,* 2001.

[18] Tufiş, D., "Exploiting Aligned Parallel Corpora in Multilingual Studies and Applications," in Toru Ishida, Susan R. Fussell, and Piek T.J.M. Vossen (Eds.), *Intercultural Collaboration. First International Workshop (IWIC 2007),* LNCS 4568, Springer-Verlag, Berlin Heidelberg, 2007.

[19] Tufiş, D., Ion, R., Ceauşu, A., Ştefănescu, D., "RACAI's Linguistic Web Services," in *Proceedings of the 6th Language Resources and Evaluation Conference – LREC 2008*, Marrakech, Morocco. ELRA – European Language Resources Association, 2008.

[20] Véronis, J. (Ed.): *Parallel Text Processing: Alignment and Use of Translation corpora*. Series: Text, Speech and Language Technology, Vol. 13, Kluwer Academic Publishers, 2000.

[14] Web Service Description Language (WSDL) 2.0: http://www.w3.org/TR/wsdl20