Adaptivity in Web-Based CALL

Galia Angelova[1], Svetla Boytcheva[2], Ognian Kalaydjiev[1]
Stefan Trausan-Matu[3], Preslav Nakov[1] and **Albena Strupchanska[1]**

Abstract. This paper presents the design, implementation and some original features of a Web-based learning environment - STyLE (Scientific Terminology Learning Environment). STyLE[4] supports adaptive learning of English terminology with a target user group of non-native speakers. It attempts to improve Computer-Aided Language Learning (CALL) by intelligent integration of Natural Language Processing (NLP) and personalised Information Retrieval (IR) into a single coherent system.

1 Introduction

Adaptive hypermedia is a relatively new research field with just a few recent examples of its successful use [1]. In contrast to traditional static systems that provide the same page content and set of links, the adaptive hypermedia aims at tuning the content or the hyperlinks to the user/usage data and the environment. Adaptive educational hypermedia aims mostly at distant learning via the Web as well as appropriate authoring tools [1]. However, building an adaptive Web-based educational application is not an easy task [2]. Almost all of the thousands available courses are networks of static hypertext with www-interface to traditional (intelligent) tutoring systems. There is only one example of a Web-inspired technique (LM matching) which could already be identified as an *original* technology. So, [2] considers the adaptive Web-based CALL as a relatively young field still searching for its own topic and essence.

In contrast to the recently launched Web-orientation, CALL is a popular area but no universal solutions are attained so far regarding the most desired features like learner-system communication in NL, adequate processing of learner's language errors and adaptive strategies for structuring the tutoring materials in NL. CALL systems are often perceived by learners and teachers as dumb and inflexible, which is demotivating for the learner and restricts the independent use of CALL systems considerably [3]. However, supporting free NL input requires integration of complex NLP techniques, esp. parsing and checking the semantic correctness of the learners' NL answers. A number of prototypes try to cope with the (almost free) NL input but "so few of these systems have passed the concept demonstration phase" [4].

The prototypes in [4] contain mostly modules for checking students' competence in vocabulary, morphology, and correct syntax usage (parsers). The most sophisticated semantic analysis is embedded in BRIDGE/MILT which *matches* the learner's utterance (a lexical conceptual structure) against the prestored expected lexical conceptual structures. More recent systems (CASTLE in RECALL [5] and SLALOM [6]) still focus on spelling, morphological, and syntactic errors. Another example is CIRCSIM-Tutor [7], which expects quite short answers, permissively extracts whatever is needed and ignores the rest. To conclude, the present CALL solutions especially for semantic analysis are far from being perfect.

This paper presents an innovative hybrid approach for building an adaptive Web-based CALL system where, most generally, personalised IR is tuned to the content of the LM (but not to the sequence of student search and browsing activities, which is the typical user modelling task in adaptive IR today). Another novelty of STyLE is the integration of advanced NLP techniques for maintaining student input as free text. Section 2 describes the whole project paradigm. Section 3 deals with the mechanism for checking the semantic correctness of the learners' NL utterances. Section 4 presents the pedagogical agent planning adaptive reactions. Section 5 describes dynamically generated Web-pages adapted to the current LM. Section 6 sketches a relevance filter for documents found in the Internet. Section 7 summarises user evaluation and section 8 gives the conclusion.

2 LARFLAST project environment

LARFLAST (Learning Foreign Language Scientific Terminology) aimed at the development of an adaptive Web-based terminology learning environment with focus on the harmonic integration of advanced NLP techniques and innovative learning solutions for system-student communication via an Open Learner Model (OLM) [8]. The potential users are students with background in economics, business or management who study English as a foreign language. From pedagogical perspective, the main objective of the project was to develop and to test with actual users a distributed Web-based learning environment which will enhance learners' performance, increase his/her autonomy and provide motivational and informational stimuli over and above what a traditional paper-based approach to learning terminology normally offers. In general the project attempts to find some balance between innovative research and practical needs. Fig. 1 presents the system components, resources and their architecture. The kernel **pedagogical resource** is a set of exercises specially designed to check user comprehension of the domain. There are exercises with fixed-choice answers and exercises requiring short

---

[1] Academy of Sciences, 25A Acad. G.Bonchev Str, Sofia, Bulgaria
[2] Sofia University, FMI, 5 J. Baurchier Str., Sofia, Bulgaria
[3] Politechnical University Bucharest, 313 Splaiul Independenti, Romania
[4] STyLE was developed in the Copernicus'98 JRP LARFLAST, November 1998- October 2001. Partners: CBLU Leeds, UK; UMIST Manchester, UK; LIRM Montpellier, France; Academy of Sciences, Romania; Simferopol University, Ukraine; Sofia University and Virtech Ltd., Bulgaria

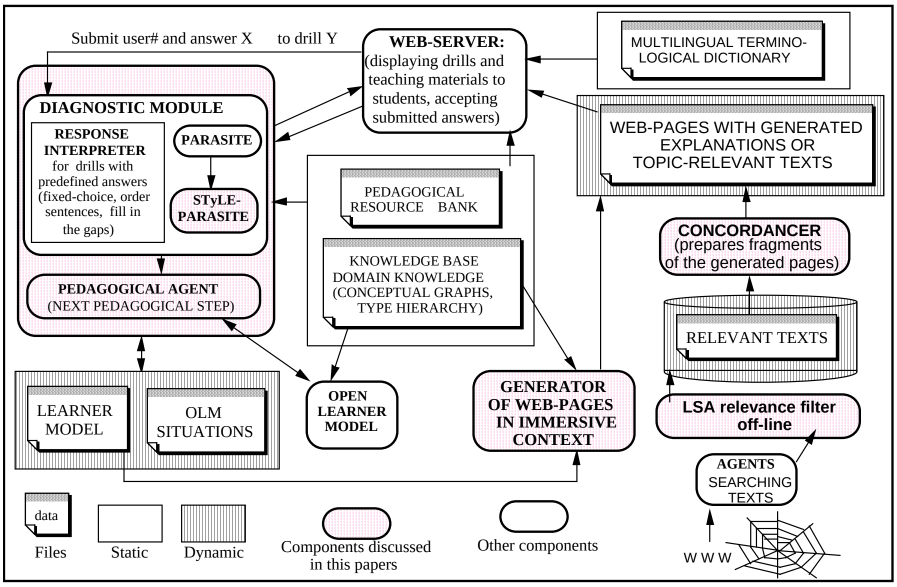

Figure 1. Main STyLE components

answers in free English, the later being particularly interesting for us in this paper. Domain knowledge is encoded in the **knowledge base (KB)** of conceptual graphs and the type hierarchy has a particularly important role for tuning the system behaviour to learners' reactions. Annotations relate the exercises to KB items, i.e. it is explicitly predefined which KB concept and relation are tested by every drill (in this way the system recognises domain misconceptions due to wrong learners' knowledge). Student performance in the exercises is controlled by the **Diagnostic module (DM)**. Answers in free English are linguistically analysed by the NL Understanding component **Parasite** [9]. Specially implemented prover **STyLE-Parasite** checks whether the linguistically correct student's answer is correct as an answer to the particular exercise performed at the moment (see section 3). All student reactions are reflected in the **LM** and special cases of misconceptions are stored for later discussion by the **OLM**. According to the LM, the **Pedagogical Agent (PA)** plans what is to be done next (see section 4). In case of decision to generate educational pages in immersive context, the generator **WebGen** is called remotely for generation of explanations tuned to the current LM (see section 5). Another STyLE component, an original implementation of **Latent Semantic Analysis (LSA)** measures the relevance of the www-collected documents to the considered terms in the financial domain (section 6). The relevance measure is on one hand a filter reducing the amount of financial texts to be considered on the fly and on the other hand, an important means for providing adaptivity at certain learning situations: PA uses the relevance measure to suggest appropriate readings as a step towards personalised IR, and WebGen extracts examples of terms-in-context from the most relevant texts. The modules considered below are original contributions of the authors; while other STyLE components and resources are not discussed in this paper.

LM is accessible as a file from the main server. Most generally it is filled in by DM while tracking user's performance (either by the **Response Interpreter** or by Parasite and STyLE-Parasite which process the free NL answers). LM has four kinds of clauses kept as Prolog facts to describe learners' familiarity is with domain knowledge:

- **know** – the learner knows a domain fact; inserted in LM in case of *correct* answer;
- **not_know** – the learner doesn't know a domain fact; inserted in LM in case of *wrong* answer;
- **self_not_know** – inserted for "*don't know*" answer;
- **know_wrongly** – learner's knowledge is considered wrong (eventually, might need corrections); inserted in LM in case of *partially correct* answer.

LM clauses have seven arguments (example in section 5):

- **user name**, defined while logging in,
- **label of a KB concept**, focused and tested in the drill that has just been performed;
- **set of KB facts**, which encode the tested knowledge;
- **drill identifier**;
- **counter** how many times the user passes trough this test item;
- **terms** indicating linguistic and conceptual mistakes (from STyLE and STyLE-Parasite), see next section;
- **unique index** for tracking the whole dialog history.

3. Checking student answers in NL

An initial user study [10] investigated how erroneous answers appear in terminology learning. Errors are usually caused by the following reasons:

- **Language errors** (spelling, morphology, syntax);
- **Question misunderstanding** - causes wrong answer;
- **Correct question understanding**, but **absent knowledge of the correct term**, which implies usage of paraphrases and generalisation instead of the expected answer;

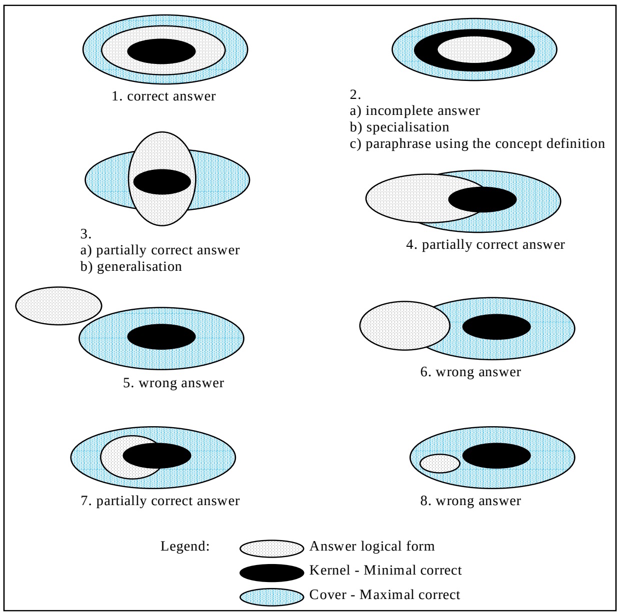

Figure 2. Comparison and inference with answers' logical forms

- **Correct question understanding**, but **absent domain knowledge**, which implies specialisation, partially correct answers, incomplete answers and wrong answers.

In principle Parasite covers errors due to the first two cases while the prover STyLE-Parasite maintains errors due to the last two cases. The expected answers are simple declarative sentences although Parasite handles complex sentences as well as simple discourse consisting of several sentences. Analysing the English input and its linguistic consistency by applying a dynamic logic, Parasite returns a logical model of the correct answers or indications of four kinds of errors (recorded as 6th argument in LM): *(i) unknown word*, *(ii) morpho*, *(iii) syntax* and *(iv) wrong*. However, knowing that an utterance is linguistically correct is not enough in CALL, for instance "*John loves Mary*" is linguistically correct but does not answer the question "*who does trade stocks on the primary market*". Therefore a second step is necessary in order to find out whether the given utterance is reasonable as an answer to the exercise being performed.

STyLE-Parasite checks the answers' correctness against the available domain knowledge and the expected answers. Most generally, STyLE-Parasite takes the logical model built by Parasite, "compares" it to the logical forms of the predefined expected minimal and maximal answers and makes the necessary inferences [11]. Fig. 2 presents the eight possible cases of intersections of the terms in the three logical forms and shows how STyLE-Parasite decides about the correctness of the input logical form (which strongly depends on the lexical choices and the syntactic structure of the concrete input). Since there might be many correct answers and their language expression varies considerably, it is not practical to compare the input to a single predefined correct logical form. Rather, STyLE-Parasite uses pre-stored maximal and minimal logical forms. Adding new terms to the maximal answer might be redundant or wrong. STyLE-Parasite inference is sound but not complete, because the conclusion "*(partially) correct learner utterances*" is indicated after the first correct binding of variables. STyLE-Parasite returns the following indications of semantic mistakes (recorded as 6th argument in LM): *(i) correct*, *(ii) more_general*, *(iii) more_specific*, *(iv) paraphrase* (usage of concept definition instead of the proper term), *(v) incomplete*, *(vi) partially_correct*, *(vii) wrong* and *(viii) combination of several mistakes*.

4. Pedagogical Agent

PA plans future learner's moves between *(i)* performing drills (active sequencing) and *(ii)* suggestion of readings (passive sequencing), where readings are texts from the Internet or specially generated Web pages. The planning is reactive and local [12]. Since considerations concern presentational and educational issues, according to the terminology in [13] we would classify the planner as performing some aspects of instructional as well as content planning. PA has two main strategies for active sequencing - *local* and *global*. The local strategy plans the movement between drills testing different characteristics of one concept. Its main goal is to create a complete view about learner's knowledge about this concept. This strategy chooses drills with increasing complexity when the learner answers correctly, and it gives again previously completed drills if the learner has performed poorly. The global strategy plans the movement between drills testing different concepts, according to their place in the financial ontology. PA chooses next learner's movement depending on: *(i)* the predefined drill's goals, *(ii)* KB items, *(iii)* concept weights defined in the drills' annotations and *(iv)* current learner's score.

PA in STyLE is a step towards personalised information retrieval and information filtering [14]. The idea is inspired by the Web-context, where main financial sites expose and frequently update relevant texts. So, if we want to show to the learner readings containing most relevant information, we have to support and continuously update a database of relevant text (the archives collected by Web spiders). The planner thus operates with *(i)* a text DB, containing financial texts collected from Internet; and *(ii)* a relevance measure, showing for each text the percentage of its relevance to the domain terms under consideration (see section 6). According to the LM terms, which are **unknown** or **known_wrongly** by the learner [14], the planner selects the text that is the most relevant to be displayed as a tutoring material (reading) in the particular learning situation. The text with higher relevance to all these terms will be used also by WebGen, when extracting examples of terms usages in a concordancer.

5. Adaptive Web pages

WebGen dynamically generates a collection of highly structured Web-pages for learning finance terminology. These pages are tailored to a specific learner, according to the information available in LM at the particular learning situation. WebGen is written in XRL, a frame-based knowledge representation language developed in CommonLisp. The generator is launched by a servlet on a different server whenever it is called from the main STyLE server. WebGen expects as input a non-empty LM.

The generated Web-pages explain the structure of a part of the domain ontology. The decision about which part has to be chosen is based on the LM sets of **known**, **not_known** and **known_wrongly** concepts. The idea of providing such a structure of Web-pages was derived from the study [10] which identifies conceptualisation ("build up the conceptual map of the field") as a crucial terminology learning activity.

Hypertext is a very suitable means for conceptualisation; domain ontology is mapped by WebGen into a network of hyperlinked Web-pages. Each concept is a Web-page and each relation is an arc, i.e. hyperlink. The ontology, the network of Web-pages, and the conceptual map to be induced in the mind of the learner have the same (semantic network) structure. The very clear links topology, easy to understand for everybody, is a prerequisite, especially in Web-organised tutoring pages, since it minimises the cognitive overload and potential disorientation when applying hypertext in learning. The uniform conceptual framework facilitates understanding. The generated Web-pages are cognitively adapted to the user since they are linked according to explicit (*is-a*, *part-of*, *agent*, *instrument*, etc.) or implicit (*similar*) relations [15]. Typical situations, examples of documents, metaphors, collocations, associated with each concept, strengthen the second, immersive dimension. STyLE assures a maximum effect of the conceptualisation process since it tailors the structure of Web-pages to the current status of the learner, derived from the LM. Figure 3 displays two Web-pages generated for the LM given below. Irrelevant clauses are deleted from this LM.

6. LSA relevance filter

Trying to dynamically retrieve documents from widely-known financial sites, STyLE uses advanced filtering to determine the most relevant documents to be recommended as "suggested readings" in a particular learning situation. In this way STyLE can enlarge and continuously update its text archive. The filtering process is off-line performed by an original implementation of LSA [16, 17]. It analyses all texts collected from the Web and generates a relevance measure for each text with respect to each of the terms in the domain KB. Only the documents whose proximity is higher than some threshold are kept and the others are discarded. Complex terms (consisting of more than one word) are placed as one term in the LSA matrix. We build the frequency matrix (excluding the stop-words and those met in just one document), transform it using the classic logarithm-entropy weighting, perform Singular Values Decomposition and keep just the top 100 singular values with the corresponding right and left singular vectors. They are used to determine the top most relevant documents for each of the domain terms. An annotation table supports fast access to the STyLE archive, containing the key terms together with a list of their best corresponding relevant documents. Practically we work only with terms tested in exercises, because only these terms can appear as **unknown** or **known_wrongly** in LM and therefore only for them relevant readings are suggested.

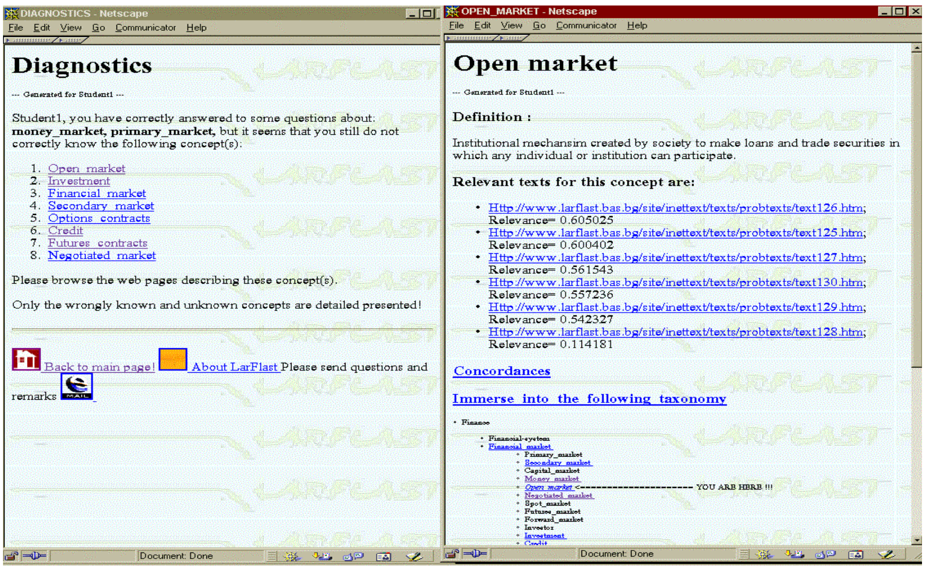

Figure 3. Dynamically generated pages for a particular LM status

know(student1,money_market,[[basic definition,2025,80],[basic definition,2022,80]],unit1_drill1,1,none,4).
not_know(student1,investment,[[basic definition,2006,80]],unit1_drill1,1,none,5).
know_wrongly(student1,secondary_market,[[basic definition,2004,80]],unit1_drill1,3,none,7).
not_know(student1,negotiated_market,[[basic definition,2008,80],[basic definition,2010,80]],unit1_drill1,2,none,9).
not_know(student1,futures_contracts,[[basic definition,2012,80],[role definition,2028,80]],unit1_drill1,1,none,11).
not_know(student1,credit,[[basic definition,2014,80],[role definition,2027,80]],unit1_drill1,2,none,13).
know_wrongly(student1,financial_market,[[basic definition,2001,80]],unit1_drill1,2,none,14).
not_know(student1,open_market,[[basic definition,2016,80]],unit1_drill1,1,none,16).
know(student1,primary_market,[[basic definition,2018,80],[basic definition,2020,80]],unit1_drill1,1,none,17).
not_know(student1,options_contracts,[[basic definition,2023,80]],unit1_drill1,1,none,18).
know_wrongly(student1,investment,[[basic definition,2006,80]],unit1_drill1,2,none,20).
know_wrongly(student1,open_market,[[basic definition,2016,80]],unit1_drill1,2,none,24).

7. Evaluation

The first user evaluation was completed in the end of 2001. STyLE was tested by *(i)* two groups of university students in finance with intermediate knowledge of English, *(ii)* their university lecturers in English, and *(iii)* a group of students in English philology. STyLE was evaluated as a CALL-tool for self-tuition and other autonomous class-room activities, i.e as an integral part of a course in “English for Special Purposes”. The learners could test their knowledge through the specially designed exercises, compare their answers with the correct ones using the generated feedback (immediate, concrete and time-saving, it comes in summary form which is crucial in order to accomplish the use of STyLE autonomously) and extract additional information from the suggested readings and concordancers.

Technically, from a learner’s perspective, STyLE is a set of Web-pages containing exercises and readings. Users reported particular satisfaction with the surrounding context of texts and terms usages organised in a concordancer. They classify as perfect the solution to have feedback about a given term immediately after they prompted erroneous answers to exercises where this term appears. What is still desirable is to restrict the genre of the suggested readings since the current texts are freely collected from the Internet and some of them cannot be used as teaching materials (LSA cannot recognise the text educational appropriateness since it considers the terms occurrences only; on the other hand it is well-known that no NLP techniques are available to accomplish such fine filtering). However, the teachers were very pleased to have concordancers with contiguously updated term usages; they would gladly see such a language resource in a further authoring tool, because searching suitable texts in Internet is a difficult and time-consuming task. Users considered the interface as “friendly”, “understandable”, and “easy to navigate” which means that they are satisfied with the adaptive approach.

Another evaluation focus was the free NL input, which attempted to provide complete NL diagnostics and is the most serious in CALL at present (up to our knowledge). Unfortunately the user perspective turned to be a different one. The learners were not impressed that for instance the sentence “*primary market operates with newly issued securities and provides new investments*” is correct since it is between the minimal answer “*primary market operates with newly issued securities*” and the maximal answer “*primary market operates with newly issued securities and provides new investments and its goal is to raise capitals*”. The main disappointment of learners and teachers is that STyLE cannot answer *why*, i.e. Parasite and STyLE-Parasite provide extremely comprehensive diagnostic about the error type but not about the error reason. Fortunately, all users liked the fact that there were numerous examples of terms usages in real texts whenever morphological or syntax errors were encountered in the free NL input. So we conclude with certain pessimism concerning the appropriateness of formal semantic approaches in CALL today and much optimism that data-driven corpus techniques, if properly applied, fit quite well to the adaptive CALL.

8. Conclusion

The reported modules of STyLE were built in 2000-2001, in a number of programming languages and hardware platforms. The final manually-acquired KB covers 150 terms and contains about 220 concepts, 23 conceptual relations and 300 domain facts. The exercises consist of about 250 single items, each annotated with the related KB item. Parasite maintains a lexicon of 300 words and lexical semantics defined in 150 meaning postulates. All components except Parasite and STyLE-Parasite are domain independent, so they can be easily shifted to a KB in a new domain. This resource allowed us to test the main project ideas with real users.

Domain terminology is a starting point in LARFLAST project, while linguistic knowledge builds around it. STyLE improves the learning outcomes because of creativity implied in the contact and because of the special emphasis on domain knowledge and terminology. Its adaptivity (including innovative aspects of personalised IR with tuning to the LM status) is the most essential feature which will be kept with scaling the KB and shifting to other domains.

References

[1] Brusilovsky, P. *Adaptive Hypermedia*. In: User Modeling and User-adapted Interaction, 2001(1-2), pp 87-110.

[2] Brusilovsky, P. *Adaptive and Intelligent Technologies for Web-based Education*. Rollinger, Peylo (eds.) Kuenstliche Intelligenz, 1999 (4), 19-25.

[3] Murphy M. and M. McTear. *Learner Modelling for Intelligent CALL*. In: Proc. User Modelling 97, Springer, 1997.

[4] Holland, V., M. Kaplan, J. and M. Sams (eds*.) Intelligent Language Tutors: Theory Shaping Technology*. Lawrence Erlbaum Ass., UK, 1995.

[5] *RECALL*, a Telematics Language Engineering project (1997), http://iserve1.infj.ulst.ac.uk/~recall.

[6] McCoy, Pennington, Suri. *English error correction: A syntactic user model based on principled “mal-rule” scoring*. Proc. UM-96, 59-66.

[7] Glass, M. *Processing Language Input in the CIRCSIM-Tutor Intelligent System*. AAAI 2000 Fall Symp. on Building Dialogue Systems for Tutorial Applications. http://www.csam.iit.edu/~circsim

[8] Dimitrova , Self, Brna, *Applying Interactive Open Learner Models to Learning Technical Terminology*, Proc. UM-2001, LNAI 2109, 148-157.

[9] Ramsay, A. *Meaning as Constraints on Information States*, in Rupp, Rosner, Johnson (eds.) *Constraints, Language and Computation*, 1994, Academic Press, London, pp. 249-276.

[10] Vitanova I., *Learning Foreign Language Terminology: the User Perspective.* Larflast report 8.1, August 1999, delivered to EC.

[11] Boytcheva, Kalaydjiev, Strupchanska, Angelova, *Between Language Correctness and Domain Knowledge in CALL*, RANLP-2001, 40-46.

[12] McCalla, G. *The Fragmentation of Culture, Learning, Teaching and Technology: Implications for the AI in Education Research Agenda in 2010*. In IJAIE (2000), 11, pp. 177-196.

[13] Vassileva J., B. Wasson. *Instructional Planning Approaches: from Tutoring towards Free Learning*. Proc. EuroAIED'96, 1996, pp. 1-8.

[14] Kalaydjiev, O. An Intelligent Pedagogical Agent in CALL. In Proc. Doctoral Consortium UM-2001, pp. 283-285.

[15] Trausan-Matu, St., *Knowledge-Based, Automatic Generation of Educational Web Pages*, Proc. of Internet as a Vehicle for Teaching Workshop, 1997, pp. 141-148.

[16] Laudauer T., Foltz P., Laham D. *Introduction to Latent Semantic Analysi*s. Discourse Processes, 25. 1998.

[17] Deerwester, Dumais, Furnas, Laundauer, Harshman R. *Indexing by Latent Semantic Analysi*s. J. Am. Society Information Sciences, 41. 1990.