Terminological Grid and Free Text Repositories in Computer-Aided Teaching of Foreign Language Terminology

Galia Angelova, Albena Strupchanska, Ognian Kalaydjiev, Svetla Boytcheva, Irena Vitanova

Bulgarian Academy of Sciences,
Institute for Parallel Processing
25A Acad. G. Bonchev Str., 1113 Sofia, Bulgaria
{galia,albena,ogi}@lml.bas.bg

Sofia University "St. Kliment Ohridski"
Faculty of Mathematics and Informatics
and Faculty of Economics
svetla@fmi.uni-sofia.bg, itv@gmx.co.uk

Abstract

This paper describes the language resources used in the project Larflast[1] and their role in the Web-based prototype STyLE (Scientific Terminology Learning Environment) which supports adaptive learning of English financial terminology with a target group of adults, non-native speakers with intermediate level of English proficiency. Larflast attempts to improve the language learning process by intelligent integration of advanced natural language technologies (deep semantic analysis of free utterances and personalised information retrieval) into a single coherent system. The learning environment STyLE is implemented as a self-tuition workbench which offers a number of drills testing learner's comprehension of financial terminology and assessing his/her knowledge. User evaluation showed the positive and negative features of our approach in general and STyLE in particular. The conclusion is that language technologies have a long way to go, until they find the proper wrappings for integration of advanced applications and the necessary resources into useful systems.

1. Introduction

Computer-Aided Language Learning (CALL) is a hot area of research but no universal solutions are attained so far regarding the most desired features like learner-system communication in Natural Language (NL) and adequate processing of learner's language errors. The market applications of 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 (Murphy & McTear, 1997). However, supporting free NL input requires integration of sophisticated techniques for semantic analysis, esp. parsing and checking the semantic correctness of the learners' answers. A number of prototypes try to support (almost free) NL input but "so few of these systems have passed the concept demonstration phase" (Holland & Kaplan & Sams, 1995). The early prototypes in the classical collection (Holland & Kaplan & Sams, 1995) 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 in a dialog based on question-answering scenario (Dorr et all, 1995). The authors point out that the syntactic and semantic correctness of the student utterances have to be checked as well as the appropriateness of the answer at the given dialog point (therefore matching to expectations is a good solution). More recent systems like CASTLE in (RECALL 1997) and SLALOM (McCoy et all. 1996) still focus on spelling, morphological, and syntactic errors. Another example is CIRCSIM-Tutor (Glass, 2000) which expects quite short answers, permissively extracts whatever is needed and ignores the rest. Recent systems rely on (spoken) dialog, partial and/or incremental analysis, and combine rule-based and data-driven approaches (see e.g. (VanLehn et all 2002) and (Rose et all, 2002)) without much progress in checking the correctness and the appropriateness of the learners' utterances. To conclude, the present CALL solutions especially for semantic analysis are far from being perfect.

This paper presents the language resources in STyLE where, most generally, semantic analysis is systematically approached and personalised Information Retrieval (IR) is dynamically tuned to the content of the learner model. STyLE integrates formal semantic techniques for maintaining student input as free text. Up to our knowledge, STyLE is the only system that attempts proving the appropriateness of the learner utterance in real time, based on predefined minimal and maximal expected answer. Focusing on domain knowledge we invested much effort in the acquisition on the conceptual resources which were encoded as conceptual graphs (Sowa, 1984). STyLE is a coherent environment where the student accomplishes three basic tasks:

(i) reading teaching materials,
(ii) performing test drills and
(iii) discussing her own learner model with the system.

An initial user study (Vitanova, 1999) investigated how erroneous answers appear in terminology learning. Errors are usually caused by the following reasons:

- **Language errors** (spelling, morphology, syntax);
- **Question misunderstanding** which 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;

---

[1] *Learning Foreign Language Scientific Terminology*, a Copernicus'98 Joint Research Project, funded by the European Commission in 1998-2001, with partners CBLU Leeds, UK; UMIST Manchester, UK; LIRM Montpellier, France; Academy of Sciences, Romania; Simferopol University, Ukrain; Sofia University and Virtech Ltd., Bulgaria.

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

This classification influenced considerably the design of the knowledge-based tutoring environment STyLE which assists non-native English speakers in English terminology learning. More details about STyLE components, functionality, architecture, and implementation are given in (Angelova et all, 2002).

The paper describes only the resources and technologies developed by the Bulgarian team in Larflast and is structured as follows. Sections 2, 3 and 4 discuss the three kinds of language resources in STyLE and give hints about their role in the learning process, their volume and the relevant technologies using the resources. Section 5 presents the evaluation results and the conclusion.

2. Terminology as a Conceptual Resource

The learning environment STyLE contains terminology organised as a conceptual hierarchy linked to the lexicon. We consider the distinction between the conceptual and the lexical resources as very important, since it imposes differences in the internal representation, the techniques providing the internal processing and the role of the two resources in the learning process. There are two important requirements imposed on the conceptual representation: firstly, it should be clear and intuitive enough to be shown to the learner with pedagogical purposes and should allow for simple graphical visualisation and secondly, it should be sofisticated enough to serve as an input to the natural language understanding component, providing the semantic analysis of the learner's answers. Acquiring the domain knowledge in this project was an effort-consuming manual activity which required proper goal-oriented combination of middle and upper models from well-known knowledge resources like CyC, WordNet, MikroKosmos, Sensus etc. We show that in a practically situated task-dependent paradigm, most ontological choi-ces like granularity of concept types, choice of conceptual relations, engineering of the explicit and implicit hierarchy, etc. are influenced by the task requirements.

Ontological Choices for Acquisition of the Type Hierarchy

Looking for more universal principles and solutions, knowledge acquisition aims at the elaboration of a knowledge base fitting to the specific project goals. We consider the choices described below as task-dependent because there might be other ways to model the same domain. Acquiring the domain model, we try to answer questions like: *which concepts, relations and facts are important for the STyLE learner*? as well as *how should knowledge be encoded in order to better satisfy the specific project requirements?*

One of the reasons to support explicitly a type hierarchy is that some fragments of the domain knowledge are shown to the learner (visualised as domain facts) when student's misconceptions are detected. This means that the student observes almost directly the internal structure of the knowledge base. Because of this project-specific aspect, we partition the types in the ontology according to the features which seem to imply the most important characterstics and differentiation to be communicated to the learner (a foreigner who studies English financial terms). So we omit types that are considered insignificant to the student. Let us consider Fig. 1 which presents a fragment of the type hierarchy for SECURITY. Another possible classification for SECURITies can be built with respect to the issuing authority. But we consider the distinction BOND-STOCK as the central one to be taught to our learners and therefore ISSUING_AUTHORITY is connected to SECURITY as a feature of the concept.

We choose label-terms whenever possible. Most financial terms are noun phrases (NPs) containing more than one word. All concept types in Fig. 1 are real terms in financial dictionaries, which are to be considered in the terminology learning course (but there are also some labels, such as PRODUCT_OF_FINANCIAL_MARKET, that are not real-life terms). It might be misleading to arbitrarily synthesize "dummy labels" for providing a more ordered ontology, because the visualisation to the learner might give rise of wrong impressions and misconceptions about external collocations of financial terms. So, we prefer to synthesize somewhat explanatory dummy labels (phrases like ISSUED_BY_A_COMPANY instead of COMPANY_SECURITY). To summarise, in the hierarchy we place either label terms, or explanatory dummy labels.

Encoding Different Kinds of Partitions in One Hierarchy

There are many ways to partition a domain, at least because of the different goals and the numerous possible view-points that might exist. The compact hierarchy in Fig. 1 encodes several kinds of partitions in one lattice, by assigning one *isa_kind* clause per partition. We use a predicate *isa_kind*/4 (see examples in Fig. 1):

isa_kind(PartitionedType, [Subtype(s)],
[PartitionKind(s)], 'PartitionName').

The fourth argument of *isa_kind* is a text string to be shown when displaying the "legend" of the partition color to the learner. A visualisation fragment is given in Fig. 2 which uses the interface of the knowledge acquisition tool in Larflast (Dobrev&Toutanova, 2000) Focusing on a single concept, graphical representation of different classification perspectives with different colors is shown as a simple and natural way for system-learner communication.

In the LARFLAST project, we considered the ontological perspectives of **natural** and **role** partitions. Natural subconcepts are classified according to unchangeable features while roles are distinguished according to temporary features. For instance, in the world of finances, one DEALER can be a BULL and/or a BEAR for different clients, so the classification *DEALER is BULL and/or BEAR* is a role partition. We mix all partitions into one hierarchy, as shown in Fig.1, and distinguish them only by the corresponding *isa_kind* predicate. In a similar way, we mix the disjoint/joint partitions and the exhaustive/non-exhaustive partitions in the same hierarchy. The default partition in Fig. 1 is a joint and unexhaustive classification into natural types.

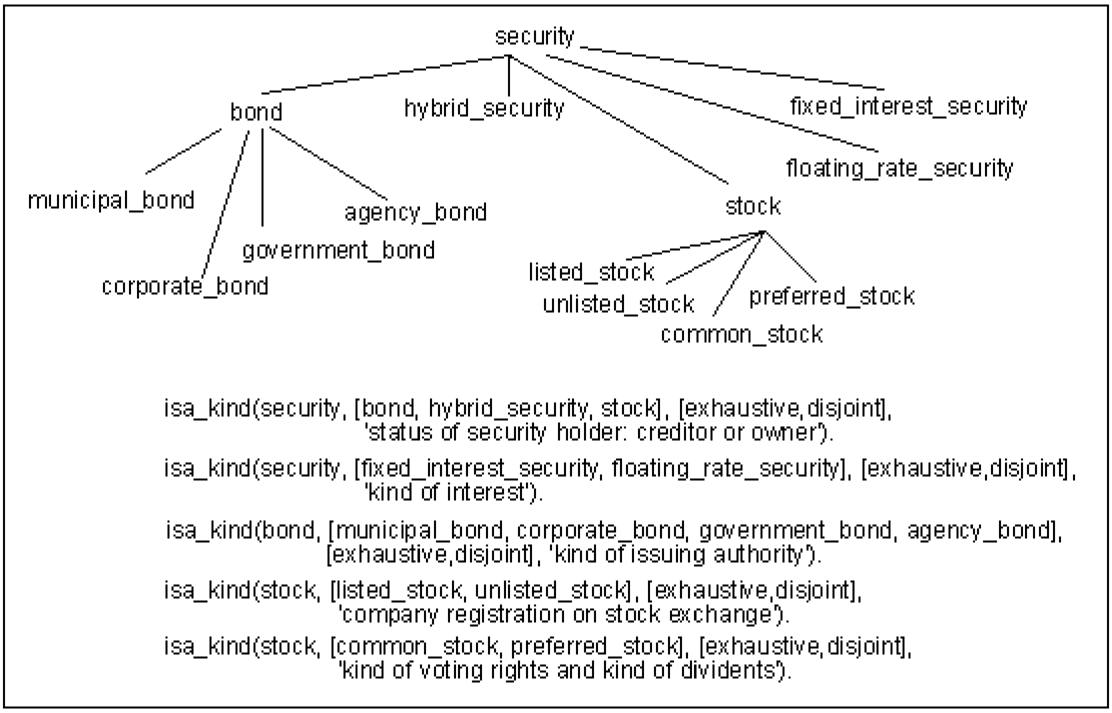

Figure 1. Ontology of terminological units and *isa_kind* perspectives.

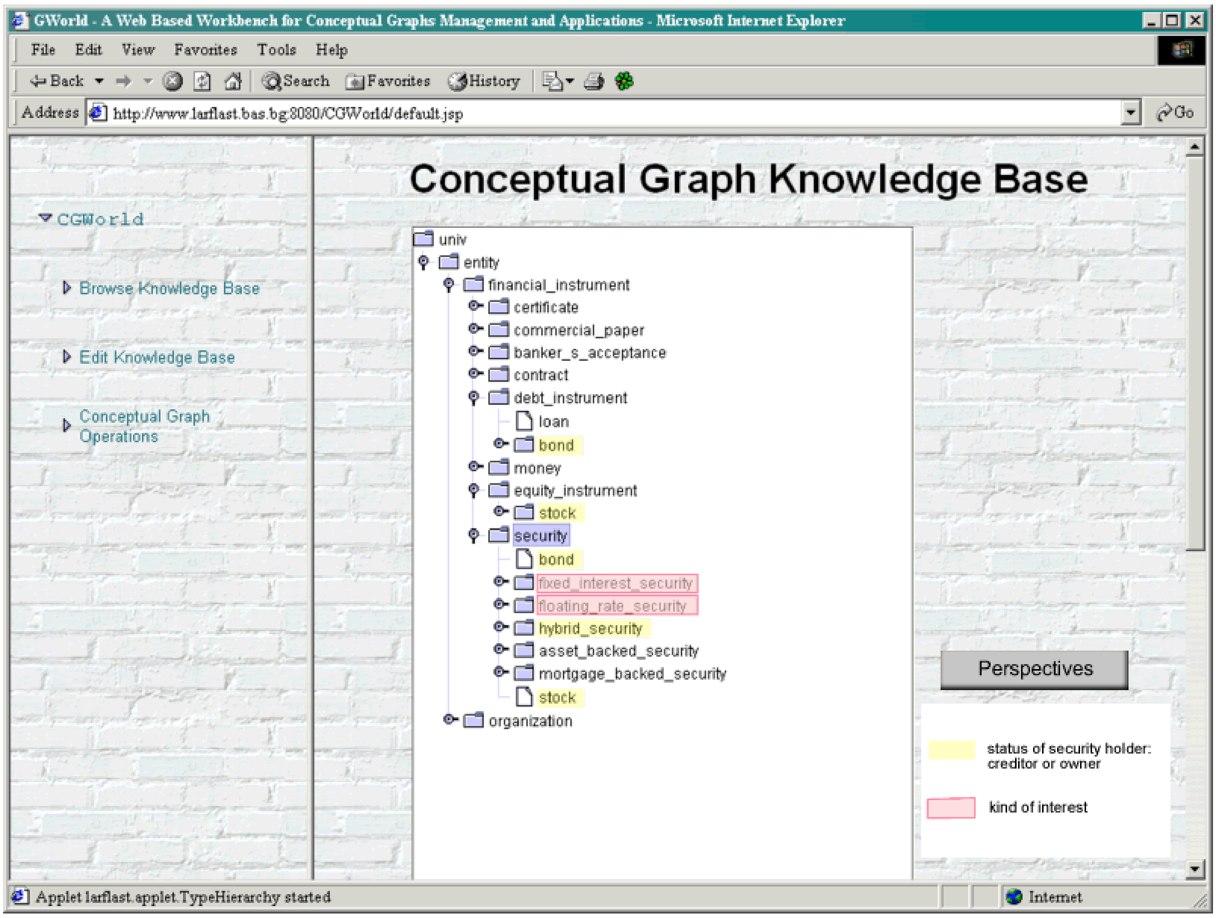

Figure 2. Visualisation of the ontology. Perspectives are marked by colors. Clicking on a type shows its partitions.

Our present knowledge acquisition experience shows that there are no simple choices in the extremely complex domain of finances. One needs mixed perspectives in the internal complicated hierarchy. However, the learner is shown a simplified and partial view to a small sub-hierarchy, which is relevant to the subject discussed at the current pedagogical situation.

Granularity of concepts and conceptual relations

STyLE integrates PARASITE as a drill-checking machine which means that the NL semantics is treated compositionally, word by word, with basic granularity of meanings as defined by word senses. Thus we need a formal technique for shifting the conceptual granularity, to assure that the domain semantics of "complex" objects is translated correctly to "one-word" meaning postulates. The shifting technique we use is implemented by ontological operations like type expansion and type contraction (Sowa, 1984). For instance, the type relation ISSUED_BY is defined as follows (the next proposition is called type definition):

relation ISSUED_BY(x,y) **is**
[ISSUE] -> (AGNT) -> [ISSUING_AUTHORITY: **y**]
-> (THEME) -> [SECURITY: **x**]

In this way we obtain facts with suitable "cascade" granularity: one encoding to be shown to the learner, for instance when visualise ISSUED_BY in the fact

[BOND]-> (ISSUED_BY)-> [COMPANY]

and another encoding with the corresponding word-by-word granularity, provided by type expansion (which corresponds to the granularity of the meaning postulates. The careful ontological elaboration at such a depth required much time-consuming efforts of knowledge engineers, language teaching experts and domain experts.

Role of the terminological grid

STyLE prototype uses about 300 English terms in financial markets, organised in a lattice built according to the different perspectives of the *is-a* relation (hyponyms, hyperonyms). The terminological hierarchy is the resource supporting the adaptive navigation through the pedagogical material, while the pedagogical agent plans learner's moves by suggestions for *performing drills* or *further readings* in case of wrong and incomplete answers and necessity to acquire more knowledge on a certain topic. *Terms* as words are the topics to be thought and they are searched in the repository of relevant texts from Internet, which become *suggested readings* if the filtering procedures decide so. *Terms* as conceptual labels participate as well in the Learner Model, where indications about the student performances to all drills are kept (after performing drills, indications like *know*, *not_know*, and *know_wrongly* are stored for every user and every term, for details see (Angelova et all., 2002)).

3. Resources encoding lexical semantics

STyLE integrates the system Parasite (developed in UMIST by Allan Ramsay, see (Ramsay&Seville, 2000)). In this section we discuss briefly the lexicon entries, which describe the terms as words, and the meaning postulates which define the lexical semantics of the terms (as we said, we distinguish between the conceptual and lexical semantics). Other linguistic resources like English grammar rules are integrated in Parasite as well, to provide the syntax and semantic analysis, but we do not consider them here.

The lexicon entries are defined in Prolog clauses and terms are entered there as nouns, verbs and adjectives to describe their morphological features. In fact, as STyLE is implemented in Prolog, the coincidences of the labels provide the links between the concepts in the type hierarchy and the words in the lexicon entries. We are interested here in the meaning postulates which are encoded manually and define the lexical semantics of the general lexica and the terms in the closed world of the project. Let us consider first several simple examples of meaning postulates for common words expressing facts that *apples are fruits* (but also have the role of *food*) and *all birds fly*:

```
lexicalMP(forall(X, apple(X) => (X is fruit) & soft(X))  ).

lexicalMP(criterial(lambda(X, apple(X)), lambda(Y,food(Y)))  ).

lexicalMP(forall(X :: {bird(X)}, fly(X))  ).
```

Further examples of meaning postulates show the way we define the semantics of financial terms, for instance

```
lexicalMP(forall(P1:: {bank(P1)}, (P1 is institutions))  ).

lexicalMP(
forall(X::{budget(X)},
    plan(X) & financial(X) &
    exists(Y::{summarize(Y)},
      exists(I::{income(I)}, theta(Y,$object,I) &
        exists(E::{expenditure(E)}, theta(Y,$object,E) &
            exists(T::{period(T)}, theta(Y, $over, T))))))   ).

lexicalMP(
forall(X::{capacity(X)},
    maximum(X) &
    exists(Y::{produce(Y)},
     exists(Z::{firm(Z)}, theta(Y,$agent,Z) &
      forall(U::{unit(U)}, theta(Y,$object,U) & count(U,X)))))  ).

lexicalMP(
forall(P1:: {company(P1)}, (P1 is institutions))  ).

lexicalMP(
forall(X::{expenditure(X)},
    money(X) &
    exists(Y::{spend(Y)}, theta(Y,$object,X)))  ).

lexicalMP(
forall(X :: {export(X)},
   (good(X) or service(X))
   & exists(Y :: {sell(Y)}, theta(Y,object,X)
       & exists(Z :: {theta(Y, agent,Z)},
          exists(ZC :: {country(ZC)}, location(Z, ZC)
            &  exists(T::{buyer(T)}, theta(Y,$to,T)
               &  exists(TC::{country(TC)},
                  location(T, TC)
                  & not(ZC = TC)
                  & forall(D :: {to(X, D) & country(D)},
                     TC = D)))))))  ).
```

Please note that the meaning postulates impose a hidden hierarchy of lexical meanings which however is different from the conceptual hierarchy as the latter reflects all perspectives interesting for the learners and assigns labels to these perspectives, to provide multiple inheritance along the different lattice branches. No doubt the two lattices – the ontological and the lexical one - are similar at an abstract level, in the sense that they contain the same information, as the conceptual partition features can be alternatively encoded as attributes of the words in the lexical hierarchy. In our case, however, we included more knowledge in the ontology as we preferred to use the visualisation utilities elaborated especially for the project, and thus to show to the learner more information in graphical format.

STyLE contains about 150 logical expressions which are either distributed with the Parasite system or developed in Larflast. They describe the semantics of words expected in the utterances, which answer to especially designed drills where the student is allowed to write down free text. Every free text input is first processed by the system Parasite which checks the syntax and the semantic correctness of the learner's free text input. After a logical form is produced – which happens for linguistically correct utterances only – an additional prover called STyLE-Parasite checks whether the logical form of the answer is "between" the logical forms of the predefined minimal and maximal expected answers for the current drill (Angelova et all, 2002). The comprehensive diagnostics allows to recognise cases like answer generalisation, answer specialisation, paraphrases using the concept definition, incomplete answer, partially correct and wrong answer. This sophisticated tool makes STyLE a very powerful environment (from formal linguistic perspective), which goes very deeply into the semantic processing compared to other systems. Fig. 3 illustrates the diagnostics options.

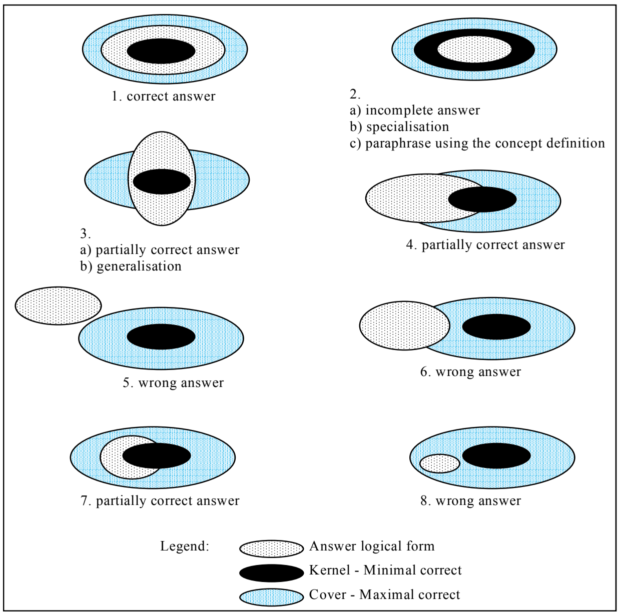

Figure 3. Diagnostics of conceptual errors

4. Collection of free texts

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. Agents - Web spiders search and deliver texts that correspond to some list of given keywords (the terminology covered by the prototype STyLE). These documents are stored on system's servers and periodically updated by newer documents with higher relevance scores. The filtering process is off-line performed by an original implementation of Latent Semantic Analysis (LSA) (Landauer et all, 1998; Deerwester et all, 1990)). It analyses all texts collected from the Web and generates a relevance measure for each text with respect to each of the terms in question. 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. Only the top most relevant documents for each of the domain terms are kept. 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 the learning module and therefore only for them relevant readings are suggested.

STyLE text archive contains 800 most relevant readings, which are html-pages containing mostly text (we excluded tables and other information that signals prevailing technical content). These texts are offered as suggested readings but are also used for building dynamic concordances which show samples of terms usages to the learner. The latter samples may be displayed in cases of language errors to drills where the student makes linguistic mistakes. Choosing this option (*view samples*) is up to the student. The dynamism of the text collection ensures the appearance of new samples, which makes the browsing interesting at every run.

5. Evaluation and Conclusion

Technically, from a learner's perspective, STyLE is a set of Web-pages containing exercises and readings. The embedded applications and components like Parasite, STyLE-parasite, generation of xml-pages, LSA, web-spiders run behind the scene or off-line. 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.

Users liked the feedback after performing drills, immediately after they prompted erroneous answers to exercises where this term appears. All of them evaluated positively the visualisation of the hierarchy as well as the surrounding context of texts and terms usages organised in a concordancer which is dynamically built and centered on the terms discussed at the particular learning situation. The teachers were very pleased to have concordancers with contiguously updated term usages; they would gladly see such a language resource integrated in a further authoring tool, because searching suitable texts in Internet is a difficult and time-consuming task. Indirectly, these positive reactions show that the idea to keep separately the conceptual representation is a fruitful one, as it allows for easy visualisation as well as for a terminology-centred design of the dialog, the navigation choices, the suggestion of further moves and so on.

We concentrated especially on the evaluation of 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 learners were not very enthusiastic regarding these modules, as they permit relatively restricted simply input and do not go beyond the human capacity of the teacher. 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. A possible improvement of the current paradigm for formal analysis is to switch to partial semantic analysis, which – at the level of the interface - will give more flexibility to the students to enter phrases instead of full sentences. What is still desirable regarding the filtering module is to restrict the genre of the suggested readings since the current texts are freely collected from the Internet and some of them should be used as teaching materials (LSA cannot recognise the text educational appropriateness since it considers the terms occurrences only; other supervised IR techniques like text categorisation might improve the filtering if they are properly integrated).

The conclusion is that teachers as well as learners like CALL systems that are easy to integrate in the typical educational tasks, i.e. the area of language learning has well-established traditions and the experimental software is well-accepted only if it is really useful and facilitates the learning process. Our feeling is that all attempts to integrate language technologies in CALL should be closely related to testing the laboratory software with real students. At the same time cooperation with teachers is an obligatory condition as the necessary pedagogical background is often missing in the research environments where normally the NLP applications and language resources appear. Language technologies have a long way to go, until they find the proper wrappings for integration of advanced applications and the necessary resources into useful CALL systems.

References

Murphy M. and M. McTear (1997). Learner Modelling for Intelligent CALL. In: Proc. User Modelling 97, Springer.

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

Dorr, B., Hendler, J., Blanksteen, S., and Migdaloff, B. (1995) On Beyond Syntax: Use of Lexical Conceptual Structure for Intelligent Tutoring. In (Holland & Kaplan & Sams, 1995), pp. 289-309.

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

McCoy, Pennington, Suri (1996). English error correction: A syntactic user model based on principled mal-rule scoring. Proc. User Modelling Conf.-96, pp. 59-66.

Glass, M. (2000) 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

VanLehn, K., Jordan, P., Rose, C., Bhembe, D., Bottner, M., Gaydos, A., Makatchev, M., Pappuswamy, U., Ringenberg, M., Roque, A., Siler, S. and R. Srivastava. The architecture of Why2-Atlas: A coach for qualitative physics essay writing. Proc. Intelligent Tutoring Systems Conf. 2002, LNCS vol. 2363, Springer, pp. 158-167.

Rose, C.P., Bhembe, D., Roque, A., Siler, S., Srivastava, R. and K. van Lehn. A hybrid language understanding approache for robust selectioon of tutoring goals. In Proc. of the 6th Inter. Conference on Intelligent Tutoring Systems, Spain, June 2002.

Sowa J. (1984) Conceptual Structures: Information Processing in Mind an Machine. Addison-Wesley, Reading, MA.

Vitanova I. (1999) Learning Foreign Language Terminology: the User Perspective. Larflast report 8.1, Sofia.

Angelova G., Boytcheva, S., Kalaydjiev, O., Trausan-Matu, S., Nakov, P. and A. Strupchanska (2002). Adaptivity in Web-Based CALL. Proc. ECAI-2002, IOS Press, pp.445-449.

Dobrev, P. and K. Toutanova. (2000) CGWorld - a Web-based workbench for conceptual graphs management and applications. Proc. ICCS-2002, Shaker Verlag, Aachen 2000, ISSN 0945-0807, pp. 243-257.

Ramsay, A. and H. Seville. (2000) What did he mean by that? Proc. Int. Conf. AIMSA-2000, Lecture Notes in Artificial Intelligence Vol. 1904, Springer, pp. 199-209.

Landauer T., Foltz P., Laham D. (1998) Introduction to Latent Semantic Analysis. Discourse Processes, vol. 25, pp. 259-284.

Deerwester, S., Dumais, S., Furnas, Landauer, Harshman R. (1990) Indexing by Latent Semantic Analysis. J. Am. Society of Information Sciences, vol. 41, 391-447.