Integration of Resources and Components in a Knowledge-Based Web-Environment for Terminology Learning

Svetla Boytcheva[1], Ognian Kalaydjiev[2], Ani Nenkova[2], and Galia Angelova[2]

[1] Department of Information Technologies, FMI, Sofia University,
5 J. Bauchier Blvd., 1164 Sofia, Bulgaria,
svetla@fmi.uni-sofia.bg
[2] Bulgarian Academy of Sciences, Linguistic Modeling Lab,
25A Acad. G. Bonchev Str., 1113 Sofia, Bulgaria,
{ogi,ani,galja}@lml.bas.bg

Abstract. This paper presents the design and currently elaborated components in the knowledge-based learning environment called STyLE. It supports learning of English terminology in the domain of finances with a target user group of non-native English speakers. [1] The components elaborated so far allow for the discussion of the Web-based learning environment, the approach to the building of a learner model, and the adaptive strategies for instructional and content planning depending on the learning situation. The paper emphasises on the specific aspects of learning terminology in a second language and checking the correctness of learner's performance within the application of STyLE.

1 Introduction

Designing tools for learning terminology in a second language is a task deserving special attention. Learning specific vocabulary in a foreign language requires the development of natural language learning environments where learners should be allowed to explore the co-relations between their language capabilities and domain knowledge. Such environments have to provide domain knowledge to be used as a source for diagnosing student's conceptual knowledge and for instructional planning. All these entails that a foreign language terminology learning environment should adopt advanced language analysis methods that focus not only on the form but also on the meaning of the student's input.

Terms constitute a relatively stable and clearly determined kernel of lexical units in any Language for Special Purposes (LSP). It is well-known that many basic terms have stable meaning without ambiguity in the considered domain,

[1] STyLE (Scientific Terminology Learning Environment) is under development in the Copernicus'98 JRP LARFLAST (LeARning Foreign LAnguage Scientific Terminology), November 1999 - October 2001. Partners: CBLU, Leeds, UK; UMIST, Manchester, UK; LIRMM, Montpellier, France; Academy of Sciences, Romania; Simferopol University, Ukraine; Sofia University and Virtech Ltd., Bulgaria.

with established typical collocations of usage in that LSP. This linguistic stability implies the numerous attempts for acquisition of formal models either as sophisticated terminological lexicons and term banks (see e.g. [1]) or as ontologies of domain knowledge ([2]). Terms and relations between them fix (in a natural way) the choice and granularity of the formal concepts, so the knowledge-based system offers to its user lexical and conceptual units which correspond to the user's intuitive fragmentation of the domain. Projects like [2] show that it is possible to build CALL-oriented domain ontologies in complex domains (law); the "core" ontology encodes the educational content communicated to the learner in order to support understanding and obtaining insight in solving cases in administrative law. Thus, the task of terminology learning exploits a relatively structured (although extremely difficult to acquire) conceptual model. Once acquired, the ontology—in addition to the educational content—might be exploited in two essential ways: *(i)* the correctness of the student's answer can be evaluated within the Knowledge Base (KB) and *(ii)* the planning of moves in the system-user interaction can be guided by this information.

Learning terminology in a foreign language is a stream in second language learning. CALL-applications for other (foreign) LSP address many potential users, by default adults with some professional demands [3]. Hence, sophisticated adaptive systems with learner modelling and proper diagnostics are highly desirable achievements. Language learning presupposes that students type free Natural Language (NL) statements since it is unnatural to acquire a new language by only selecting menu options. CALL systems that provide such learning environments require Natural Language Processing (NLP) techniques to check the correctness of learner's utterances. Due to the very complicated nature of the task, however, it is difficult to find successful examples of intelligent CALL prototypes for second language in general and for terminology in particular. In the state-of-the-art collection of NLP in CALL papers [4], a general conclusion that "so few of these systems have passed the concept demonstration phase" has been made. As application of NLP techniques, the systems described 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 the system BRIDGE/MILD [5], [6], [7] which matches the learner's utterance (a lexical conceptual structure) against the prestored expected lexical conceptual structures. This matching is implemented by an algorithm defining the intuitive notion of a *correct match*; the simple examples for semantic correctness in [7] show that testing semantics is far beyond the foreseen progress expected in near future. To conclude, it seems clear that every project for language learning (including ours) needs to be restricted by a balanced choice of what the system gives to and expects from the learner, what is the main focus, which AI techniques are available to provide reaction in real time etc.

This paper presents results obtained so far in a project where the main focus is improving the understanding/writing competence of the learner (adult, non-native English speaker) in the domain of finances. The implementation, the Web-based tool STyLE, follows some principles and ideas presented in [8].

Section 2 sketches the project as a whole. Section 3 presents in more detail the diagnostic module, the learner model, the design of drills' annotation, the mechanism for checking the correctness of the learner's utterances in free NL, and the pedagogical agent planning local reactions in certain learning situations. Section 4 contains an example. Section 5 concludes the paper by discussing further work.

2 LARFLAST Project Paradigm

The project aims at the development of a Web-based learning environment where the student accomplishes three basic tasks (reading teaching materials, performing test drills and discussing her own learner model with the system). The project is oriented to learners who need to improve their English language competence as well as their expertise in correct usage of English financial terms. Thus in general we attempt at finding some balance in the achievement of the goals: *(i)* to cover enough domain knowledge and relevant English terms; *(ii)* to test students' language and conceptual knowledge, and *(iii)* to find easy ways of student-system communication and discussion of learner misconceptions by diagrammatic representations, which are considered a powerful expressive language (the chosen technique is Open Learner Model (OLM), see e.g. [9]). This ambitiously formulated knowledge-based paradigm implies the necessity:

- to support an intuitive conceptual representation (providing simple graphical visualisation of domain knowledge and learner model facts to the learner),
- to integrate formal techniques for NL understanding, allowing for analysis of the users' answers to drills where the student is given the opportunity to type in free NL text (the system Parasite, developed at UMIST by Allan Ramsay—see e.g. [10]—is already integrated in STyLE).

Knowledge base. The central knowledge resource in Larflast is a manually acquired Knowledge Base (KB) of conceptual graphs. Domain knowledge in finances is kept in four formats [11]: *(i)* graphical, used by the knowledge engineer during the knowledge-acquisition phase and by OLM for communication of diagrammatic representations to the learner; *(ii)* first order logic, applied when important domain facts are translated as meaning postulates to be used for proving the correctness of learner's utterances by Parasite; *(iii)* CGIF, used for generation of Web-pages explaining the educational content of domain knowledge in immersive context; and *(iv)* Prolog representation used for further KB processing by generalisation, specialisation, natural join, projection etc. Formats *(ii)*, *(iii)* and *(iv)* are automatically generated by the primary representation *(i)*. Specific problems and solutions relevant to acquisition of domain knowledge are considered in [12].

Current Implementation. Fig. 1 shows already elaborated components of STyLE, integrated under a Web-server, and internal software communications.

The system Parasite provides checking of the morphological, syntactic and semantic correctness of the learner's utterances in especially designed drills, while the prover STyLE-Parasite checks the correctness against the available domain knowledge. OLM is developed by the Leeds team [9]. Other STyLE components, which provide learning materials with interface oriented to the student/teacher and which are developed by the other project partners, are not shown in Fig. 1.

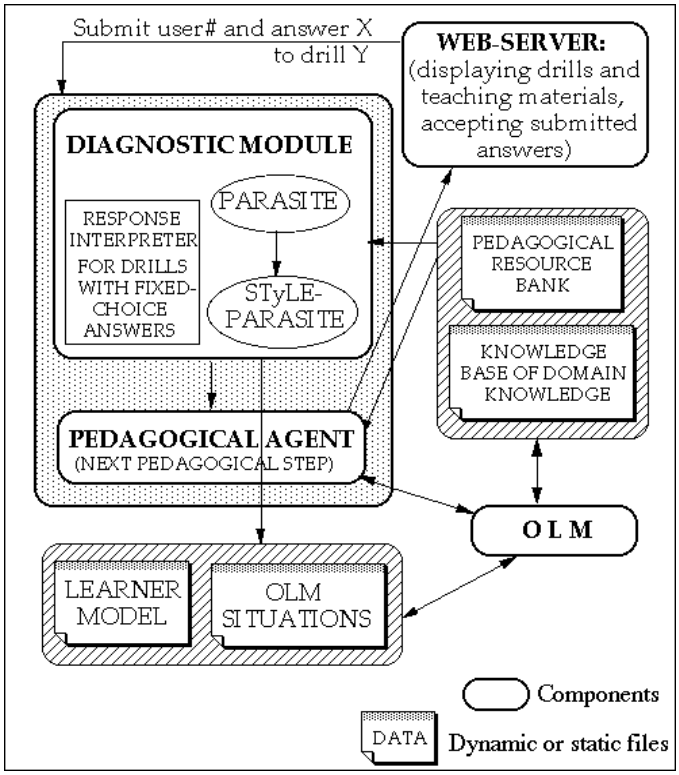

Fig. 1. Architecture of current STyLE components, discussed in this paprer

3 Pedagogical Resources and their Maintenance by STyLE Components

Having the main the responsibility for the integration of the STyLE components, in this paper we focus on the following issues:

- elaboration of the pedagogical resource bank: *(i)* drills and their annotation: predefined drill goals, correct answers, etc.; *(ii)* pedagogical knowledge: weights of domain concepts/relations with respect to teaching, records for possible learners' errors with close-semantic and close-language friends, etc.
- design and maintenance of the learner model;

– development of the prover STyLE-Parasite, which—in drills with free NL input from the learner—takes a logical form from the Parasite's output and proves its domain correctness within the context of the KB facts, by matching the given answer against the expected answer(s);
– design and development of a pedagogical module (pedagogical agent), which plans local reactions in certain learning situations.

At present, the communication between the learner and STyLE is maintained by two main modules—Diagnostic Module (DM), which is responsible for the Learner Model (LM), and Pedagogical Agent (PA). DM assures analysis of the learner's performance and fills in the LM; PA plans what is to be done next and refers, when necessary, to *(i)* OLM, where the learner discusses about her conceptual knowledge, and/or to *(ii)* the generator of dynamic web-pages, where the learner reads relevant texts with immersive context (this generator is not shown in Fig. 1).

Currently STyLE offers test unit, covering about 80 basic English terms in finances. Each test unit consists of an explanatory text about important concepts and a set of drills of both types—with fixed choice answers and with free-text answers (see [13]). After the learner completes a drill with fixed choice answers, the results are submitted to DM where the response interpreter analyses the answer and computes the learner's score. After the learner completes a drill with free text entry, the answer is submitted to Parasite for linguistic analysis and is passed to STyLE-Parasite for proving the domain correctness of the utterance. All information about learner's performance is passed over to the PA, it plans what is to be done next depending on the learning situation and calls OLM if OLM situations have arose after the completion of the previous drill(as in Fig.1).

3.1 Pedagogical resource bank of drills and pedagogical knowledge

Following some established practice in web-design of drills (see e.g. the Half-baked educational software [14]), STyLE user is presented with seven types of drills: *Multiple choice, Gap fill, Crossword, Jumbled-sentence, Matching exercise, Ordering exercise* (usually with fixed-choice answers) and *Text-entry* (with free text answers).

An annotation (internal description) is associated with every drill. It

– shows the way the answer is to be checked—how to match the given answer to a preliminary stored set of correct responses. One drill entry can be associated with more than one possible correct answers with different score;
– contains information about how different drills and their items relate to domain concepts, encoded as KB items. A number—weight between 0 and 10—is associated with each concept and it shows the domain importance of this concept in respect to teaching;
– contains explicitly stated goals of drills, showing which facts and relations concerning given concept are being tested. More then one goal can be associated with one and the same drill if it tests different perspectives of relations.

Possible goals are: *test definition of concept X, test relations between X and Y, test similarity between given concepts, test difference between given concepts.* All goals have weighs, which is important for the planning of what is to be shown next.

Fig. 4 shows fragments of drill annotation. The predicate test_aspect/6 records in its 3rd-6th arguments correspondingly the correct answers (strings true/false in this case), the tested relation, the tested KB concept and the concept weight.

3.2 Diagnostic module

DM checks the correctness of the learner's answer, generates feedback with learner's score to the learner, fills in the LM and organises data for further reflective dialogue with the learner in OLM. As shown in Fig. 1, there are two main modules analysing learner's performance in drills:

Response Interpreter for drills with fixed-choice answer. It matches the learner's and expected answers and marks all the cases where they coincide and where they differ by asserting the fact that the user knows, respectively does not know the concept and attributes set in the goal of the drill's item. The score is calculated according to the number of correctly answered items in the drill. While matching the answers, the interpreter analyses all the history in the LM facts and if a contradiction arises, it records an "OLM-situation".

STyLE-Parasite Interpreter for drills with free-text answer. To provide advanced NL understanding in cases when the learner is given the opportunity to type in freely, Larflast integrates the system Parasite.

Parasite either recognises the learner's utterance as a correct one or returns information about linguistic inconsistency of learner's utterances: morphological, syntax and semantic errors (in the later case no logical form can be computed). Answers with correct linguistic semantics are subjects to further considerations of their domain relevance, proved by STyLE-Parasite. At present STyLE-Parasite distinguishes the following cases of wrong conceptualisations: *(i)* over-generalisation, *(ii)* over-specification, *(iii)* usage of concept the definition instead of its name, *(iv)* predicates—i.e. domain facts—included in the answer expectation but missing in the student's answer, *(v)* parts of the student's response that lead to contradictions with the answer expectations. Relevant information about all cases is asserted in the LM.

Domain correctness is proved in several basic steps as follows:

- Preparation of expected answers: preliminary generation and storage as files of the syntax trees and logical forms of all correct expected answers. Human experts choose the "essential minimum" in each answer;
- Preparatory analysis in run-time: application of Parasite to each learner's answer for obtaining the syntax tree and logical form of the learner's utterance;

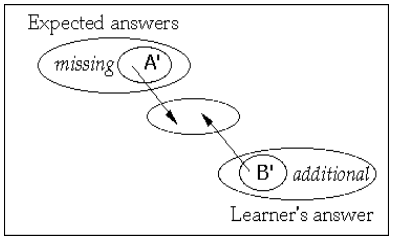

Fig. 2. Space of search in STyLE-Parasite

- Comparison of the inference sets by STyLE-Parasite: the two inference sets (the expected one, A, and the received one, B) are compared as sets of predicates (see Fig. 2). All predicates from $B \backslash A$ are recorded in a file *additional*. Then B is reduced to $B' = B \backslash additional$. All predicates from $A \backslash B'$ are stored as file *missing* and A is reduced to $A' = A \backslash missing$. STyLE-Parasite compares A' and B'. This procedure does not change the number of the occurrences of each of the remaining predicates (with different values as arguments) in each of the resulting sets. If the number of predicates in B' is greater than those in A', then after binding all of the predicates' variables in A', redundant predicates are removed from B' (i.e. those predicates from B' which have some unbound variables) and appended to the file *additional*. If the number of the predicates in A' is greater than the number of those in B', the binding of all variables is impossible and this leads to contradiction.
- Search within the space of possible bindings of the free variables in A' and B'. STyLE-Parasite applies heuristics for binding of the variables in A' and B' predicates: the predicates with more free variables and least binding candidates have priority for binding. Contradiction causes backtracking.

There might be several kinds of mistakes in the received answer, so learner's utterances are to be investigated with respect to all possible error types applying the above-described steps. STyLE-Parasite inference is complete, since it finds all existing ways to bind the variables. But it is not necessary to find all bindings, because the conclusion "correct learner utterances" is indicated after the first correct binding and the proving halts.

3.3 Learner Model

LM keeps track of learner's performance during all sessions. After analysing user's answer to each drill, DM asserts to the user's LM information about her knowledge: e.g.

```
know(UserName, Concept, [List Relations], DrillName, Number).
```

Currently four types of diagnostics about the learner's knowledge are asserted: $know$, not_know, $self\_not\_know$ and $know\_wrongly$. The Number argument shows for which time the concept is tested. This allows us to keep track of the stability of user's knowledge because we can detect cases of gaps and changing performance.

3.4 Link to Open Learner Model

Analysing learner's answers, DM discovers situations where either the learner or the system need further dialog, providing elaboration of learner's conceptual knowledge. This entails a link to the OLM component. The following situations are diagnosed at present:

- *contradiction*—there are LM-facts *know* and *not_know* about the same concept. This mean that the user's knowledge is not stable or that she does not know some of the more complicated attributes of the concept;
- *confuse close semantic concepts*—LM shows that the learner confuses concepts marked as very closely semantically related (for example money market and financial market). We remind that information about semantic closeness in teaching is explicitly encoded by the domain/teaching expert in the pedagogical resource, to point domain concepts and relations usually confused by novices in the domain;
- *confuse close language concepts*—LM shows that the learner confuses concepts that sound related, because of the words constituting the term. These types of confusion are typical for non-native speakers, who are mislead due to phonological or linguistic similarity [3].

While in the first situation a dialogue in OLM aims at solving the inconsistency in the learner's knowledge, in the next two a further interaction learner-OLM articulates aspects of learner's domain knowledge and assigns possible reasons for the learner's errors. OLM situations are shown at Fig. 6.

3.5 Pedagogical Agent

The main role of PA at present is to plan future learner's moves between lessons and drills. Since considerations concern presentational as well as educational issues, according to the terminology in [15] we would classify our planner as performing some aspects of instructional as well as content planning. There are two main movement strategies—local and global. The local strategy plans moves between drills, testing different characteristics of one and same 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 gives again previously completed drills if the student has performed poorly. The global strategy plans movements between drills, testing different concepts, according to their place in the ontology. PA chooses next learner's movement depending on: *(i)* Predefined drill's goals, *(ii)* KB items, *(iii)* Concept weights and *(iv)* Learner's Score.

If the score is under 50% of the maximal one, PA shows link to readings. If after the learner's answer OLM situation arises, PA shows link to OLM for further discussions. PA always offers the default moves to correct answers and next drill or unit.

4 Example

Figure 3 shows a fixed-choice drill and the corresponding student's answers in the STyLE environment. The internal drill identifier is 'drill_2'

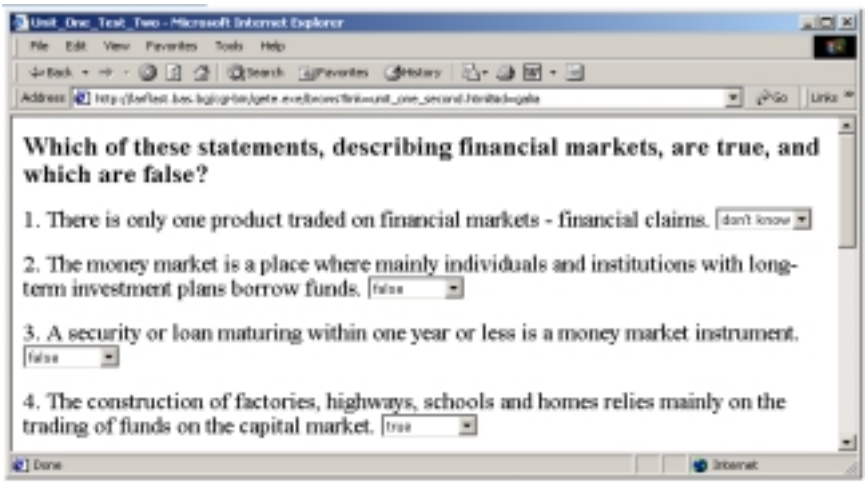

Fig. 3. Student performance in STyLE

Learner's answers are matched with the drill annotation shown in Figure 4. The learner "student007" answers "I don't know" to the first item of the drill, which is recorded as a fact in the LM (see Figure 5). This data is used by PA for selection of suitable study materials.

```
test_aspect(drill_2, item1,[false], [object], financial_market,10).
test_aspect(drill_2, item2,[false], [attribute], money_market, 5).
test_aspect(drill_2, item3,[true], [instrument], money_market, 7).
test_aspect(drill_2, item4,[true], [attribute], capital_market, 5).
```

Fig. 4. Fragments of internal drill annotation

Drill entries two and three are interesting because they test different type of information about the same concept, the learner knows one of them and does not know the other one. Those facts are put in the LM and OLM situation of the first type (contradiction) is registered. As a suggestion for next move PA generates to the learner's web-interface a web-page containing, together with the default hyperlinks, a hyperlink to OLM and another hyperlink to readings, relevant to financial market topics.

```
self_not_know(student007, financial_market, [object], drill_2, 1).
know(student007, money_market, [attribute], drill_2, 1).
not_know(student007, money_market, [instrument], drill_2, 2).
know(student007, capital_market, [attribute], drill_2, 1).
```

Fig. 5. Status of LM obtained after the work of Response interpreter

5 Conclusion and further work

This paper considers the present components developed within the ongoing Larflast project. Current results allow to evaluate important aspects of the final product: *(i)* we believe that it is possible to achieve a relatively simple but complete ontology of 100-200 terms in the financial domain; *(ii)* on-line integration of Parasite can be done in a Web-environment by attentive design of appropriate drills; *(iii)* planning helps essentially in guiding the learner within a rich environment where the learner is offered many choices, including free Web surfing, and seems to be an obligatory control component.

The future work includes integration of the whole system STyLE and relevant user study and evaluation.

References

1. Galinski C. and Budin G. (1995) Terminology. In Cole R. A. et al., Eds., Survey of the State of the Art in Human Language Technologies, http://www.cse.ogi.edu/CSLU/HLTsurvey/HLTsurvey.html
2. Breuker, J., Muntjewerff A. and B. Bredeweg. (1999) Ontollogical Modelling for Designing Educational Software. In Proc. AI-ED Workshop on Ontologies for Intelligent Educational Systems, Le Mans, France, July 18–19, France.
3. Vitanova, I. Learning Foreign Language Terminology: the User Perspective. Larflast report 8.1, August 1999, delivered to the EC.
4. Holland, V. M., Kaplan, J. and M. Sams (eds.) Intelligent Language Tutors: Theory Shaping Technology. Lawrence Erlbaum Associates, UK, 1995.
5. Sams, M. (1995) Advanced Technologies for Language Learning: the BRIDGE Project within the ARI Language Tutor Program. In [4], pp. 17–21.
6. Weinberg, A., Garman, J., Martin, J. and P. Merlo.(1995) A Principle-Based Parser for foreign Language Tutoring in German and Arabic. In [4], pp. 23–44.
7. Dorr, B., Hendler, J., Blanksteen, S., and B. Migdaloff. (1995) On Beyond Syntax: Use of Lexical Conceptual Structure for Intelligent Tutoring. In [4], pp. 289–310.
8. Alpert, S.R., Singley, M.K. and Fairweather, P.G. (1999). Deploying intelligent tutors on the web: an architecture and an example, IJAIED, 10, 183–197.
9. Dimitrova, V., J.A. Self and P. Brna. The interactive maintains of Open Learner Models. In Lajoie S.P. and Vivet M. (eds.), Proc. 9th Conf. AIED, Frontiers of AI and Applications, Vol. 50, IOS Press, pp. 405–412, 1999.
10. Ramsay, A. Meaning as Constraints on Information States, in Rupp, Rosner , Johnson (eds.) Constraints, Language and Computation, 1994, Academic Press, London: 249–276. Parasite home page at: http://ubatuba.ccl.umist.ac.uk
11. Dobrev, P., and Kr. Toutanova. CGWorld - A Web-Based Workbench for Conceptual Graphs Management and Applications. To appear in ICCS-2000, Darmstaht, Germany, August 2000.
12. Angelova, G., A. Nenkova, Sv. Boycheva, and T. Nikolov. CGs as a Knowledge Representation Core in a Complex Language Learning Environment. To appear in ICCS-2000, Darmstaht, Germany, August 2000.
13. Larflast project site, http://www.larflast.bas.bg/site
14. Hot Potatoes at: http://www.halfbakedsoftware.com/licence/
15. Vassileva, J. and B. Wasson. (1996) Instructional Planning Approaches: from Tutoring towards Free Learning. In Proc. EuroAIED'96, Lisbon, Portugal, 1996, pp. 1-8.