An Approach to Representation and Extraction of Terminological Knowledge in ICALL

Darina Dicheva[1] and Vania Dimitrova[2]

[1] Faculty of Mathematics and Informatics, University of Sofia, Bulgaria
[2] Computer Based Learning Unit, University of Leeds, UK

This paper addresses an innovative approach to computer assisted learning of foreign language terminology which involves supporting not only foreign language learning focused on specific terminology but also the enhancement of conceptual knowledge in the subject area. ITELS - an intelligent tutoring system aimed at helping Bulgarians to learn English terminology in a particular subject area exemplifies the main ideas of this approach. The paper focuses on the issues of representation and extraction of terminological knowledge, which are of crucial importance for the system's overall performance. The most significant aspect of the proposed approach lies in separating language knowledge from subject area knowledge. The paper suggests a way of building a terminological knowledge base and of using it for intelligent language instruction.

Keywords: Terminology Knowledge Processing, Conceptual Graphs, Computer Assisted Language Learning, Intelligent Tutoring Systems.

1. Introduction

Foreign language instruction for university students in Bulgaria usually comprises two modules. The first one covers a basic set of words and general grammar rules of the target language. The second one introduces the students to the particular terminology of their subject area. The success of this instruction depends on the students' knowledge of the semantics of the domain terms. Classroom experience shows that students often have problems with the subject area concepts which makes learning the foreign language terminology difficult.

This paper addresses an innovative approach to computer assisted learning of foreign language terminology which involves supporting not only foreign language learning focused on specific terminology but also the enhancement of conceptual knowledge in the subject area. ITELS — an Intelligent Tutoring System (ITS) aimed at helping Bulgarians to learn English terminology in a particular subject area exemplifies the main ideas of our approach. The paper focuses on the issues of representation and extraction of terminological knowledge, which are of crucial importance for the system's overall behavior depending on the efficiency of the main pedagogical activities performed by the system: answering questions, suggesting help, generating exercises, and diagnosing student knowledge.

ITELS (Intelligent TErminology Learning System) is designed with the following main features in mind:

- *Intelligence.* It is a knowledge-based system which provides adaptive language instruction with an emphasis on vocabulary acquisition. The system includes the typical ITS modules: an expert module, a student modeling module, and a pedagogical module (Wenger 1987). In addition, it includes an authoring module allowing the human teacher to supply learning materials for use by the system's pedagogical module. The main design principles and the system architecture are discussed elsewhere (Dimitrova & Dicheva 1998).
- *Usability.* In addition to technical university students, the system can also be used by translators of technical texts — to help them

in translation and improve their understanding of the subject area concepts.

- *Reusability.* The system has two distinct layers — a language layer and a conceptual layer. Representation of the knowledge in the conceptual layer is based on Conceptual Graphs (CGs) (Sowa 1984) and is language-independent. Thus the system can be adapted and extended to other languages by only replacing the language layer. In another way the approach is extended through the development of tools for building a CGs knowledge base allowing the adjustment of the system to support learning in different terminological areas.

The paper is organized in the following way. First, we briefly describe some related work by discussing knowledge representation in language learning systems and outlining some terminological knowledge systems. Then we describe the expert knowledge in ITELS which is represented in two ways: implicitly and explicitly. Both parts of the expert model — the linguistic knowledge and the conceptual knowledge — are described in detail. Section four addresses the representation of the student's terminological knowledge. How does the system extract terminological knowledge in order to provide feedback, to generate new learning materials, and to diagnose student's knowledge is the focus of the discussion in section five. Next, we briefly describe the current state of the system's implementation. We conclude by summarising the approach presented and give some directions for future work.

2. Background

A large number of Computer Assisted Language Learning (CALL) systems have already been developed and successfully used to assist mainly second language learning in its initial phase. For a detailed survey see, for example, (Cameron et al. 1986) and the recent Higgin's review on using computers in language education (Higgin 1995). Most authoring language-based systems are organised as a series of frames including material (text and audio-visual) to be displayed to the student together with a list of anticipated answers and associated actions. The set of responses that can be matched is predetermined and, consequently, answers from students who have difficulties or misunderstanding not anticipated by the courseware author cannot be accommodated. The recent trend in the development of CALL systems is the use of multimedia technology (Yazdani 1993).

The difference between Intelligent CALL (ICALL) and CALL has usually been their explicit expertise, i.e. their domain knowledge, typically embodied in their parsers and grammars, and not student modelling capability or their tutorial expertise (du Boulay 1992). Most of the ICALL systems do focus on form because that is what parsers — a core Natural Language Processing (NLP) technique — do well (Dorr et al. 1995). The work in ICALL is concentrated in two main areas, both concerned with the student's ability to write syntactically correct sentences in a second language: correction of syntactic errors in individual sentences freely chosen by the student or in individual sentences chosen by the system (c.f. Bowerman 1990, du Boulay 1992).

Despite significant progress made in NLP technology, recent ICALL systems face problems in language tutoring caused by the limitations of the technology and the complexity of its application (Holland et al. 1995). The fundamental limitation of most ITSs in language learning lies in their lack of semantic analysis. There are few attempts for incorporating some very restricted semantic knowledge allowing the system to check not only the form but also the content of the student's responses to very simple questions (Weischedel 1978, Culley 1992, Frederiksen et al. 1992, Handke 1992, Dorr et al. 1995, Decoo et al. 1996, Ingraham et al. 1996).

Most of the ICALL systems focus on learning foreign language syntax, but there are a few focusing on the second language vocabulary learning. These systems are typically built as learning environments supplying various tools for representing and extracting word meaning, see for example (Swartz 1992, Chanier 1996, Berleant et al. 1997).

In order to avoid ITSs constrained capacity to produce effective lessons, few ICALL systems provide authoring tools allowing the teachers to create their own lessons and to make suggestions about sequencing those lessons (Levin &

Evans 1995, Kreyer & Criswell 1995, Kaplan & Holland 1995). All these, however, are oriented only to support learning the second language's syntax.

Some of the problems that the students face when learning foreign language terminology are similar to those experienced by the translators in translating specialised texts in unfamiliar domains: quite often they don't know the concepts behind the terms. Recently a knowledge-based approach to understanding and translating technical texts has been suggested (Meyer et al. 1992, Holmes-Higgin & Ahmad 1992, vHahn & Angelova 1994) where the translators are supported by a user-friendly environment providing linguistic and domain knowledge explanations. The knowledge-based machine aided translation systems, however, are not aimed at assisting foreign language terminology instruction: they act more like terminological dictionaries and domain encyclopaedias. They support the user's exploration of the subject domain but fail to offer teaching assistance and guidance. For example, the explanations they provide may well contain further unfamiliar terms.

A subject area language which is a language for special purposes is built out of subject area terms. Representation of the terminological knowledge is a central issue in the terminological knowledge-based systems. They inherit their general characteristics from KL-ONE (Woods & Schmolze 1992, Buchheit et al. 1993). These systems suggest semantic models which allow the information to be retrieved from the terminological KB by employing a deductive process, thus solving basic problems of terminological knowledge processing.

What is ITELS? It is designed as an ICALL system for vocabulary learning with a focus on scientific, technical vocabulary (Dimitrova & Dicheva 1998). The system provides tools allowing the courseware author to construct instructional materials and suggest a suitable sequence for presenting these materials to the student. However, being *intelligent*, the system can change the preset sequence in order to adapt the instruction to the individual needs of the particular learner. When necessary, it automatically generates learning materials by using the system's knowledge base. In addition to the *system-controlled* mode of use, ITELS can be used as a learning environment which offers various learning activities to be explored by the student.

Following the dual nature of the terms (they are language constructions which name concepts with special meaning in the subject area) the expert model and the student model in ITELS include both types of knowledge: linguistic knowledge and conceptual knowledge (Dicheva & Dimitrova 1996). The linguistic expert knowledge (a lexicon and a set of morphological rules) represents the lexical competence the student is intended to acquire (Dimitrova & Dicheva 1997). The conceptual knowledge represents term semantics, i.e. subject area concepts and relations between them. In the following sections we discuss the knowledge representation in ITELS in detail.

3. The Expert Knowledge

The expert knowledge is presented in the system in two forms — implicit and explicit.

3.1. Implicit knowledge

As in traditional CAL systems, the implicit teacher's expertise is concentrated in learning blocks (training and information blocks) designed by the courseware author. These blocks build the system Data Base (DB). In each learning block the teacher not only embeds her/his subject matter and pedagogical knowledge related to teaching a particular term, but also specifies the terms which are relevant to the goal term (by including them in the same block). This information is used by the system for instructional planning.

Each training block is built around a term from the terminological area under consideration and contains either a question or an exercise together with its level of complexity. Four basic kinds of questions are included in the system:

- *multiple choice* questions (including *Yes/No* questions);
- *multiple answer* questions (with more than one correct answer);
- *fill-in-the gap* questions (the answer is to be filled in a blank position in the text);

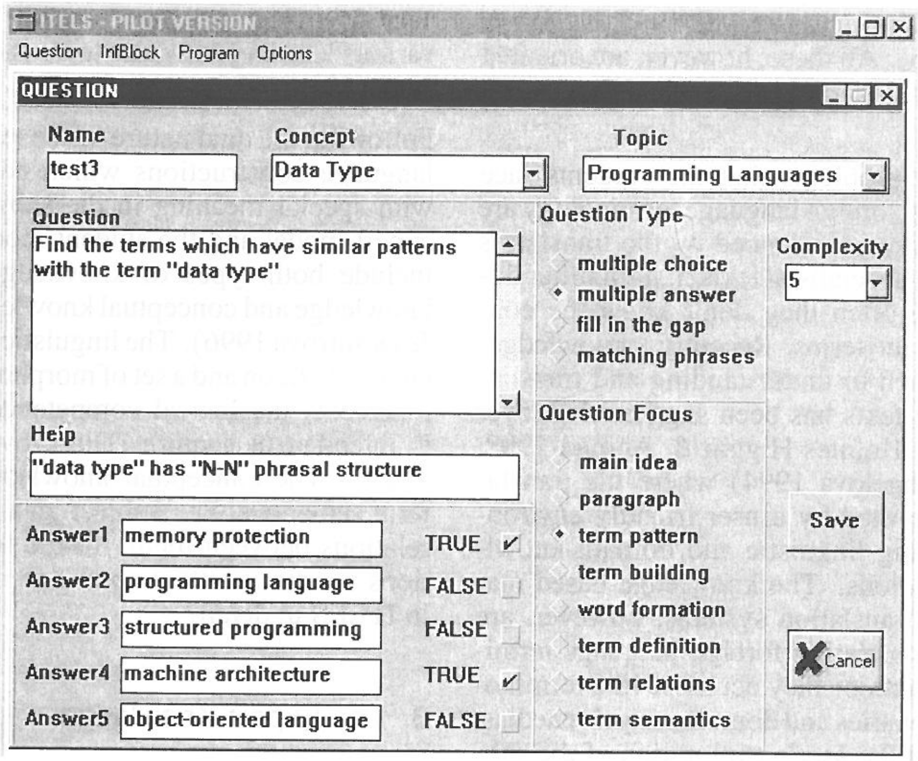

Fig. 1. The interface of the authoring module.

- *matching phrases* questions.

Each question focuses on a particular aspect of terminology learning: understanding the reading text, term's phrasal structure, or term semantics. The interface of the authoring module, representing an example of constructing a multiple answer question, is shown in Figure 1.

The language-oriented exercises include sentence composition or translation of very simple sentences, changing the word order to get a correct sentence, and changing the order of sentences to get a meaningful text.

The information blocks represent implicitly the teacher's knowledge about the most appropriate help that the student could get when having difficulties with particular terms. Besides the text they could also include visual information. Information blocks are used by ITELS' pedagogical module in addition to the help generated automatically by the system.

3.2. Explicit knowledge

The explicit knowledge representation is the main feature of an ITS. The expert module containing the explicit knowledge provides the domain intelligence for the system (Anderson 1988). For any given domain it is necessary to decide not only what knowledge to include in the expert model but also how to encode it. The expert module represents the domain-specific expert knowledge and the inferring process involved in solving problems in the domain. In language domains expert knowledge includes a lexicon and some type of grammar for the target language, and the expert inference engine is a parser to process language inputs (Swartz & Yazdani 1992).

The new idea in ITELS is related to representing and using semantic knowledge as well. Thus, in addition to the linguistic knowledge (see Section 3.2.1.), its expert model is extended to incorporate semantic knowledge and the corresponding inference engine (see Section 3.2.2.).

3.2.1. Linguistic Knowledge

The core part of the linguistic knowledge in an intelligent tutoring system for vocabulary learning is the lexicon. The ITELS lexicon contains entries for ordinary words and for terms. Ordinary words are represented by their grammatical category (as parts of speech), translations in Bulgarian, and information about inflections.

This information is used when the student requires the translation of a particular word. Since ordinary words are not the subject of instruction in ITELS, it is neither feasible nor necessary to support any deeper information such as explanation, multilingual information, synonyms, antonyms, etc.

The entry for each term includes its definition, phrasal structure (pattern), translation, and a link to the knowledge base. The latter allows an efficient search of the KB, for example, when generating an explanation. In order to avoid term ambiguity, for terms with several meanings one entry per meaning is encoded. Figure 2 summarises the main structure of lexical entries for terms. Other information about terms (e.g. inflections) is kept in the ordinary entries.

Psycholinguists consider acquiring word formation skills (pattern recognition and production of word forms) as one of the main aspects of vocabulary acquisition (Ellis 1995). Understanding the meaning of affixes and the way they are used to build words is very useful in tackling new lexical items.

Entry_ID:	$T\#m_1$
Entry:	'output'
Pattern:	n
Sense_Label:	'data'
Translation:	'izhodni danni'
Definition:	'The result of data processing activity when it is presented external to the system.'
Concept_ID:	$C\#k_1$

Entry_ID:	$T\#m_2$
Entry:	'output'
Pattern:	n
Sense_Label:	'signal'
Translation:	'izhodni signal'
Definition:	'A signal that is obtained from an electrical circuit, such as a logic circuit'
Concept_ID:	$C\#k_2$

Entry_ID:	$T\#m_3$
Entry:	'output'
Pattern:	n
Sense_Label:	'process'
Translation:	'izwegdane na danni'
Definition:	'To produce a result.'
Concept_ID:	$C\#k_3$

Fig. 2. An example of lexical entries. The term *'output'* has three meanings *'data'*, *'signal'*, and *'process'* (Dictionary of Computing 1990).

In ITELS the word formation knowledge is represented in a set of *affix rules*. Similarly to (Byrd 1983) each affix rule is presented by *affix*, its *kind* (prefix/suffix), *type* (the affix's purpose), *meaning* (the affix's meaning), *condition* (the sub-categorisation and selection constraints on the base), and *result* (the categorial characteristics of the result after applying the affix). Figure 3 shows some examples of affix rules.

(*un, prefix, 'negative/positive', 'not, not good enough'*, { }, { })

(*mini, prefix, 'size', 'small'*, { }, { })

(*ation, suffix, 'noun-forming', 'in the manner of', verb, noun*)

(*er, suffix, 'noun-forming', 'a person who / a thing which', verb, noun*)

(*all, suffix, 'adjective-forming', 'having the quality of', noun, adjective*)

Fig. 3. Some affix rules.

The expert module includes a morphological engine developed to process the word formation knowledge. It works in two modes:

1. Analysis

   Input: a word.

   Output: the base and the affixes from which the word is derived or NIL.

   Examples:

   computer → compute, -er

   computational → compute, -ation, -al

   compute → NIL.

   Description: The engine seeks the affix of the input word by pattern matching and compares the result with the word category. Then removes the affix and finds in the lexicon the base corresponding to the affix condition. The algorithm is repeated recursively.

2. Synthesis

   Input: a base and an affix.

   Output: the word from the lexicon that is derivable by applying the affix to the base or NIL.

Examples:

compute , -er → *computer*

compute , -or → *NIL.*

Description: If the base satisfies the rule condition for applying the affix, the engine adds the affix to the base and checks whether the generated word is in the lexicon.

3.2.2. Conceptual Knowledge

The main role of the domain in a terminological area is the structuring of knowledge by determining the relations between the terms, which represent lexical items and related concepts as units of abstraction of subjects and facts from reality. The conceptual knowledge is mainly declarative and usually represented by frames, semantic networks or other formalisms relevant to the declarative knowledge representation.

The terminological knowledge in ITELS comprises the subject area terms and the relations between them. It is classified into topics. Currently, the area of Computer Science is chosen as a test-bed and the following topics are considered: *"Data Organisation"*, *"Programming Language"*, *"Computer Organisation and Architecture"*, *"Operating Systems"*, *"Software tools"*. The conceptual knowledge in each topic is organised into two parts representing correspondingly the type hierarchy (taxonomy) and the relationships between concepts.

The **taxonomy** of the concept types represents their level of generality. The hierarchy permits information inheritance. A part of the hierarchy of the topic *"Programming Language"* is shown in Figure 4. The type is a denotation for a set of individuals. For example, *Object – Oriented Language* = {*SmallTalk*, *Simula*, *C++*,...}.

Each concept from the subject area is determined by its *concept type* (from the type hierarchy) and its *referent* which represents specific individuals. We denote concepts by *pairs* [*Concept Type: Referent*]. Some of the concepts do not identify a particular individual, so they are *generic concepts*. The referent part for these concepts is omitted. *Individual concepts* refer to particular individuals. For example, the concepts [*Object Program*], [*Source Program*], and [*Object Language*] shown in Figure 5 are generic concepts whereas the concept [*Action: 'Translate'*] is an individual concept.

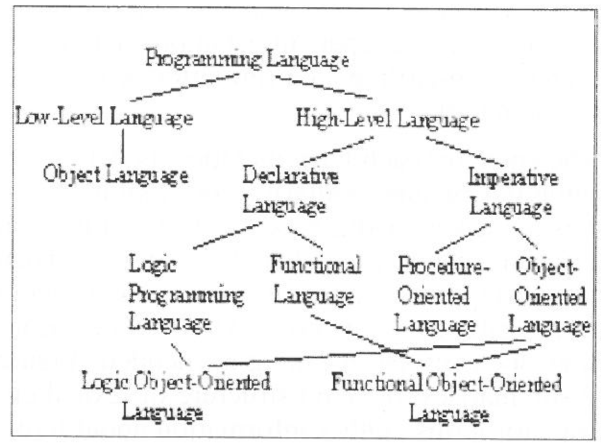

Fig. 4. A part of the *"Programming Language"* KB: Concept hierarchy.

The relationships between concepts indicate that their meanings are connected in some way. A set of basic **conceptual relations**, commonly used in terminological areas, is presented in the system. This includes:

- case relations: agent (AGNT), patient (PTNT), recipient (RCPT), instrument (INST), and result (RSLT);
- attribute relations: general (ATTR), characteristic (CHRC), and part (PART);
- metarelations: kind (KIND), subtype (SUBT), description (DSCR), statement (STMT), and representation (REPR);
- *false friends* relation (FLSF): a relation between two concepts which are often confused, e.g. *"data type"* and *"data structure"*.

The relations between concepts are represented in ITELS by conceptual graphs. The conceptual graph representation has been chosen since it is a formalism with direct mapping to natural language, allowing a convenient extraction of sentence meaning (Sowa 1992). A conceptual graph connects concepts with conceptual relations. An example of a conceptual graph that represents the sentences *"An object program is the translation of a source program into an object language"* is shown in Figure 5.

New conceptual graphs can be derived from the CGs in the knowledge base by applying the operations *restrict, join and simplify.*

```
[ACTION: 'Translate']–
    (RSLT) -> [Object Program]
    (PTNT) -> [Source Program]
    (INST) -> [Object Language]
```

Fig. 5. An example of a CG that represents the sentence *"An object program is the translation of a source program into an object language"*.

A conceptual graph $G_r = restrict(G)$ is derived from G either by replacing a concept from a given type T with a concept from a type which is a subtype of T or by replacing a generic concept with an individual concept from the same type. An example of the *restrict* operation is shown in Figure 6. The CG which represents the sentence *"Smalltalk has the attributes of polymorphism, inheritance, data abstraction, and encapsulation"* is derived from the CG representing the sentence *"An object-oriented language has the attributes of polymorphism, inheritance, data abstraction, and encapsulation"*.

```
G: [Object-Oriented Language]–
    (ATTR) -> [Polymorphism]
    (ATTR) -> [Inheritance]
    (ATTR) -> [Data Abstraction]
    (ATTR) -> [Encapsulation]
```

```
Gr: [Object-Oriented Language: 'Smalltalk']–
    (ATTR) -> [Polymorphism]
    (ATTR) -> [Inheritance]
    (ATTR) -> [Data Abstraction]
    (ATTR) -> [Encapsulation]
```

Fig. 6. An example of the *restrict* operation: $G_r = restrict(G)$.

The operation *join* combines two graphs containing identical concepts (with the same concept type and referent) by overlying one of the graphs on the top of the other so that the two identical concepts merge into a single concept. In the resulting graph this single merged concept is linked to all the conceptual relations that have been previously linked to each of the original concepts. If one of the concepts is generic and the other one is individual, they could be joined after restricting. Figure 7 shows an example of the join operation. The new graph $G_j = join(G_1, G_2)$ represents the semantics of the sentence *"Smalltalk has the attributes of polymorphism, inheritance, data abstraction, and encapsulation. It is the first object-oriented language developed at the Rank Xerox research centre."*

When two graphs get joined some relations in the resulting graph may become redundant. Any of the duplicated relations can be deleted by the operation *simplify*.

```
G1: [Object-Oriented Language: 'Smalltalk']–
    (ATTR) -> [Polymorphism]
    (ATTR) -> [Inheritance]
    (ATTR) -> [Data Abstraction]
    (ATTR) -> [Encapsulation]
```

```
G2: [Object-Oriented Language: 'Smalltalk']–
    (CHAR) -> [First]
    (Developed) ->
        [Research Centre: 'Rank Xerox']
```

```
Gj: [Object-Oriented Language: 'Smalltalk']–
    (ATTR) -> [Polymorphism]
    (ATTR) -> [Inheritance]
    (ATTR) -> [Data Abstraction]
    (ATTR) -> [Encapsulation]
    (CHAR) -> [First]
    (Developed) ->
        [Research Centre: 'Rank Xerox']
```

Fig. 7. An example of the *join* operation: $G_j = join(G_1, G_2)$.

For any two CGs G and G_1 such as G_1 is derived from G by applying a sequence of conceptual operations, there exists a *projection mapping* $\pi : G \to G_1$ with the following properties:

1. For each concept $c \in G$ the concept $\pi c \in G_1$ has a type which is common or a subtype of the type of c. If c is an individual concept, then both c and πc have also the same referents.
2. For each conceptual relation $r \in G$ linked to c its image $\pi r \in G_1$ is linked to πc (Sowa 1984).

The *projection mapping* allows extraction of information from a CG related to specific properties of the concepts (e.g. attributes, characteristics, etc.).

New types of concepts and relations can be defined in terms of simpler ones. A new type is defined by specifying its supertype (*genus*) and a defining graph (*differentia*) that allows the new type to be distinguished from the genus. Figure 8 shows the definition of the new type

Functional Language. New conceptual relations can also be defined by constructing conceptual graphs using basic relations.

```
type: Functional Language (x)
      genus: Declarative Language
      differentia:
      [Declarative Language *x]-
      (INST) -> [Function { *}]
```

Fig. 8. A definition of the new concept type *Functional Language*.

4. Student's Knowledge

The current state of the student's terminology knowledge is represented in the student model. Because of the complexity of the knowledge represented in ITELS we use a combination of approaches for student diagnosing, including variants of the overlay and issue tracing approaches (Polson & Richardson 1988). The student model represents both the grammatical and the terminological knowledge of the student.

As the grammar acquisition is beyond ITELS primary goals, the part of this knowledge that is represented is very simple: it only indicates whether the student correctly applies a few grammar rules related to number, indefinite and definite articles, and case. The grammatical part of the student model consists of a number of counter pairs for these basic grammar rules. The first counter of each pair represents the faults and the second one accounts for the correct use of the corresponding rule.

The second part of the student model represents the level of mastery of the foreign language terminology by the student. It is a kind of an overlay model. The terminological knowledge consists of linguistic and conceptual knowledge. For each term it includes:

- spelling of the term;
- translation of the term in English;
- phraseology (ordering words in the term);
- place in the taxonomy of the terms;
- definition of the term.

The level of the student's understanding of each term is represented by a set of indicators corresponding to the above characteristics attached to the term's entry in the KB. The scale used consists of four states: *completely known, probably known, probably unknown, and completely unknown.*

Each conceptual relationship is covered by similar measures related to the level of its acquiring.

5. Extracting Terminological Knowledge

The efficient performance of an ITS depends heavily on the ways of extracting knowledge from the system's KB. This section discusses the extraction of terminological knowledge, necessary for completing the main pedagogical activities in ITELS, namely:

- answering questions;
- suggesting feedback;
- generating exercises;
- diagnosing student's current knowledge.

5.1. Answering questions

ITELS uses a mixed-initiative approach which allows questioning by the system and questioning by the student. Allowing the student to ask the system for explanations enhances the system's diagnostic abilities. Indeed, the student's questions provide complimentary information about her/his state of understanding which in some cases appears to be the only source of diagnostic information.

Five types of questions are supported in the system. The student could ask about:

- the grammatical category of the anticipated answer or of any term from the suggested list of possible answers (in the currently used learning block);
- the term definition;
- the translation of a term (i.e. the anticipated answer) after supplying it in the native language;
- terms *similar* to the anticipated answer;
- further information about the anticipated answer.

When the student asks about the grammatical category or about an explanation of a term (a term definition), the system extracts this information from the lexicon (see Section 3.2.1.).

Sometimes it is useful to allow the student to supply the anticipated answer in the native language and to ask the system about its translation in the target language. For example, the student may know the meaning of the expected answer, i.e. s/he may know it in the native language, but does not know its translation in the target language. In this case, there is no gap in the conceptual knowledge of the student but in her/his linguistic knowledge. So, instead of giving the student a series of useless and frustrating questions for teaching her/him the term's meaning the system could just supply the term's translation in the target language. This information is also extracted from the lexicon.

The student could ask about some terms similar to the anticipated answer. Two concepts are considered to be similar if their types belong to the same concept hierarchy in the knowledge base and the difference between their levels is less than two or if there is any kind of relationship between them (see Section 3.2.2.). There are two kinds of similarity:

1. *Hierarchical similarity.* Three kinds of links are considered:

   • supertype, e.g. for terms of the type *Logic Object-Oriented Language* similar terms are all terms of the type *Logic Programming Language* and of the type *Object-Oriented Language* (see Figure 4);

   • subtype, e.g. for terms of the type *High-Level Language* the similar terms are of the types *Declarative Language* and *Imperative Language*;

   • terms with common parents, e.g. for *Functional Object-Oriented Language* such a similar term is *Logic Object-Oriented Language*.

2. *Relational similarity.* The terms are in the same CG or a graph that could be obtained from the CG knowledge base by using the operations *restrict, join*, and *simplify*. For example, the terms [*Object Program*], [*Source Program*], and [*Object Language*] are similar to the term [*Action: Translate*] (See Figure 5).

When the student asks about the terms that are hierarchically similar to the anticipated answer the system finds these terms in the KB and generates explanations using pre-stored templates. For example, if the anticipated answer is [*High-Level Language*] the system will generate the following sentences:

> *The correct term is a subtype of Programming Language.*
>
> *The correct term is a supertype of Declarative Language and of Functional Language.*

Similarly, when the student asks about terms relationally similar to the anticipated answer the system will find these terms in the KB and will generate the corresponding explanations. For example, for [*Object-Oriented Language: Smalltalk*] the following explanation will be generated from the conceptual graph G shown in Figure 7:

> *The correct term has attributes Polymorphism, Inheritance, Data Abstraction, and Encapsulation.*
>
> *It is the first Object-Oriented Language. It is developed in the Rank Xerox Research Centre.*

The suggested explanations can be optionally tuned to the current state of the student's knowledge (as represented in the student model). The system could suggest either a *general* explanation or a *tuned* explanation. In the first case all *similar* terms from the KB will be used, while in the second — only the terms already known by the student.

When the student asks about further information related to the anticipated answer the system displays a relevant information block (defined by the teacher) from the DB (see Section 3.1.). The underlying idea is to give the teacher the opportunity to present her/his own teaching expertise. Such enrichment of the system's explanatory capability improves the system's performance.

5.2. Suggesting feedback

Besides the student's requests for information the system itself could decide that the student needs some help. The system could suggest information about the grammatical category of the anticipated answer, the term's definition, or some information supplied by the teacher. In

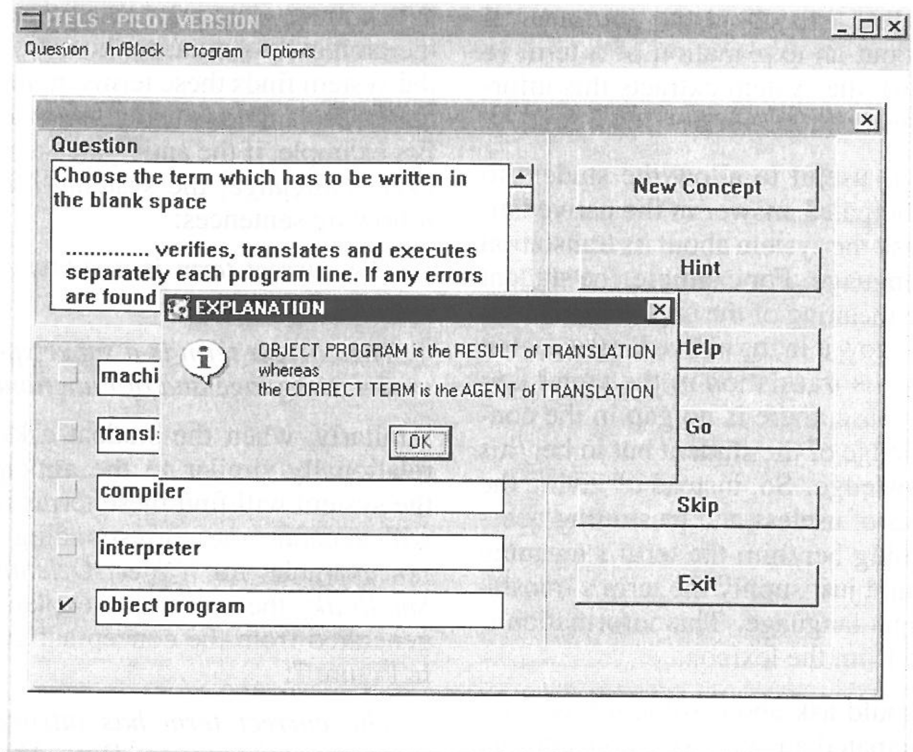

Fig. 9. System's feedback.

addition, it could give a hint to the student in case s/he is near to the correct answer. For this purpose the system first finds out whether the two terms — the correct one and the one suggested by the student — are similar either through hierarchical similarity or relational similarity (see Section 5.1.).

For example, if the expected correct answer is [*Compiler*] and the student suggests the term [*Interpreter*], the system will generate the following explanation:

> *The correct term and the term Interpreter have a common parent — the term Translator.*

Figure 9 shows the explanation suggested by the system when the student has suggested the term [*Object Program*] instead of the correct term [*Compiler*].

The system explains the difference between a concept type and its *genus* by using the *differentia* from the type definition (see Section 3.2.2.). For example, the difference between the terms [*Declarative Language*] and [*Functional Language*] will be explained by the following sentence generated in accordance with the definition shown in Figure 8:

> *The functional language is a Declarative Language which operates with functions.*

The system could also suggest information about all *similar* (*hierarchically or relationally*) terms to the correct answer. The explanation can again either be a *general* or a *tuned* explanation.

5.3. Generating exercises

The pedagogical module controls the overall performance of the system. It determines both the term to be exercised next and the teaching/learning activity to be undertaken. The latter includes presenting a *training block* to the student. When the system cannot find an appropriate *training block* in the DB (see Section 3.1.) it automatically generates exercises. It is able to generate two kinds — word formation exercises and the ones on term semantics.

In order to generate word formation exercises the system uses the *synthesis* mode of the morphological engine which works on the affix rules and the terms entries in the lexicon (see Section 3.2.1.). Three kinds of word formation exercises can be generated:

1. Given an affix (e.g. *-er*) and a number of bases (e.g. *operate, compute, disk, compile, print*) the student is asked to indicate which of the bases can take the affix.
2. Given a base (e.g. *compute*) and several affixes (e.g. *un-, -ation, -er, -ment, -ness*) the student is asked to find out which of the affixes can be taken by this base.
3. Given a list of affixes of similar function (e.g. the noun-forming suffixes *-er, -or, -ness, -ance*) and a list of bases the student is asked to match bases against affixes.

As to the exercises on term understanding the system extracts relevant conceptual knowledge from the CGs KB. It generates sentences either directly from existing CGs or from CGs obtained by using *joint, restrict*, or *simplify* operations and uses some templates to construct *multiple choice* and *fill-in-the-gap* exercises. For example, Figure 10 displays an exercise generated from the CG shown in Figure 5.

term: *object program*
kind: *fill-in-the-gap*
type: *term understanding*
content: Fill in the blank the correct term:
"An .... is the translation of a source program into an object language."

Fig. 10. An exercise generated from the CG shown in Figure 5.

5.4. Extraction of terminological knowledge for student diagnosis

The system has to *know* the level of the student's understanding of the currently considered term. In order to diagnose the current state of the student's knowledge the system compares the student's knowledge against the expert's knowledge, more precisely — the student's answer against the correct answer. To do so it extracts terminological knowledge from the KB. When the student's and the expert's answers are different the system determines whether they are *near* or *far*. Two terms are *near* if they are *similar hierarchically* or *relationally*. Two terms are *far* if they are not *near*.

The system undertakes different pedagogical actions depending on whether the terms are *near* or *far*. For example, if the student suggests the term [*Polymorphism*] whereas the correct term is [*Inheritance*], after projection mapping on the CG G_1 shown in Figure 7 the system concludes that the student probably knows that both terms are attributes of [*Object Oriented Language*] but does not understand the difference between them. The student can be given the definitions of both terms extracted from the lexicon. If the student suggests an answer which is *far* from the correct one, the system concludes that s/he does not know either of the terms and probably is confused with other terms similar to them. For example, from the confusion of the term [*Object Language*] with the term [*Object-Oriented Language*] (which are far) the system will conclude that the student has problems with some of the terms similar to the above terms: [*Low-Level Language*], [*Imperative Language*], [*Procedure-Oriented Language*], [*Logic Object-Oriented Language*], and [*Functional Object-Oriented Language*] (see Figure 4), [*Object Program*], [*Source Program*], and [*Action: Translate*] (see Figure 5), [*Polymorphism*], [*Inheritance*], [*Data Abstraction*], and [*Encapsulation*] (see the CG G from Figure 6). These terms will be included in the list of the terms to be exercised next.

6. Implementation

The pilot version of the system is implemented in C++ and runs in a Windows environment. It has a simplified distributed architecture with a blackboard model for implementing the communication between modules. The system guides the instruction by choosing the term to be focused upon next, selecting relevant exercises from the predefined data base, and providing appropriate feedback. The student uses buttons to ask for help or to select terms to be exercised.

The current version of the system incorporates an Authoring module which allows the teacher to create learning materials — texts, exercises, and help materials. A Conceptual Graph editor has been developed to allow the knowledge engineer to build a subject area knowledge base. The system KB currently includes computer science terms from the topic "*Programming Languages*".

In the pilot version some simplifications of the system modules have been made. The engine that generates sentences from CGs uses simplified algorithms based on those described in (Sowa 1984). Since the basic idea was to tackle intelligent tutoring problems related to teaching terminology, we tried to avoid NLP problems as much as possible. For example, teaching grammar is beyond the primary goals of the system, the lexicon has simplified structure and the module for generating natural language sentences includes only a restricted number of language generation rules concerning articles, numbers, and some agreements. It could well happen that explanation generated by the system contains repetitions and not very well formulated sentences. Some system components such as the morphological engine and the module for generating exercises are still under development.

The pilot version of ITELS will be the subject of an experiment in the regular English course for Computer Science students at Shumen University in the Autumn 1998. This will allow for a proper evaluation of the system.

7. Conclusion

ITELS is an intelligent system for foreign language terminology learning, which comprises reusable components. The system is aimed at assisting both:

- foreign language learning focused on specific terminology;
- enhancing the knowledge of the particular subject area concepts.

Efficient system performance depends heavily on the ways of extracting terminological knowledge. The paper focuses on issues related to the representation of terminological knowledge as well as the extraction of terminological knowledge for answering students' questions, suggesting feedback, generating exercises, and diagnosing student knowledge.

The most significant aspect of the proposed approach lies in separating the language knowledge from the subject area knowledge. The added value of this separation is the reusability of the system: it can be easily adapted to support teaching of different terminological areas. The language knowledge is represented by a lexicon and morphological rules. The conceptual knowledge represents the semantics of the subject area terms and is based on conceptual graphs. The paper suggests a way of building a terminological knowledge base and of using it for intelligent language instruction.

Our future plans include improving the system modules related to natural language processing as well as elaborating ITELS CGs model in order to implement a modifiable student model. The latter implies providing graphical tools for observing and changing the student model. This will be a development of the suggested (Cumming & Self 1991) idea for system-student cooperation in constructing the student model.

Acknowledgements The research reported here was done when the second author was a PhD student at Shumen University. We would like to thank first Jordan Dimitrov — for his valuable help and continuing assistance during the work. Special thanks to the students Julian Yorgov and Atanas Zankov for taking part in the implementation of the pilot version. Thanks also to Paul Brna — for proof reading this paper and for all his support. We are particularly grateful to the anonymous referees whose comments and recommendations helped us to improve the quality of the paper.

This work was partly funded by NSF Project No. MY-MM 6/96 and NSF Project No. I-502/95.

References

J. ANDERSON, The Expert Module, In *Foundation of Intelligent Tutoring Systems* (M. Polson, J. Richardson Eds.), (1988) pp. 21–53, Lawrence Erlbaum.

D. BERLEANT AND L. SHI ET AL, LEARN: software for foreign language vocabulary acquisition from English unrestricted text, *Computer Assisted Language Learning*, **10** (2) (1997).

C. BOWERMAN, ICALL an underview of the state of the art in CALL, *Computer Assisted Language Learning*, **3** (1990), 45–54.

M. BUCHEIT, F. DONINI AND A. SCAERF, Decidable reasoning in terminological knowledge representation systems, *Journal of Artificial Intelligence Research*, **1** (1993), 109–138.

R. BYRD, Word Formation in Natural Language Processing Systems, Presented at the *Proceedings of the Eigth International Joint Conference on Artificial Intelligence*, (1983), William Kaufmann Inc., Los Altos.

K. CAMERON, W. DODD AND S. RANTZ, Eds., *Computers and Modern Language Studies*, Ellis Horwood Limited, 1986.

T. CHANIER, Learning a second language for specific purposes within a hypermedia framework, *Computer Assisted Language Learning*, **9** (1) (1996), 3–43.

G. CULLEY, From Syntax to Semantics in Foreign Language CAI, In *Computer-Assisted Instruction and Intelligent Tutoring Systems*, (J. Larkin, R. Chabay, Eds.), (1992) pp. 47–72, Lawrence Erlbaum.

G. CUMMING AND J. SELF, Learner modelling in collaborative intelligent educational systems, In *Teaching Knowledge and Intelligent Tutoring* (P. Goodyear, Ed.), (1991) Ablex Publishing Corporation Nowood, New Jersey.

W. DECOO, E. HEUGHEBAERT, N. SCHONENBERG AND E. VAN ELSEN, The standard vocabulary programs of Didascalia: in search of external versatility and didactic optimisation, *Computer Assisted Language Learning*, **9** (1996), 319–338.

D. DICHEVA AND V. DIMITROVA, Representation and extraction of terminological knowledge in ITELS — an intelligent system for foreign language terminology learning, Presented at the *Proceedings of the Joint Conference on Knowledge-Based Software Engineering*, (1996), Sozopol, Bulgaria.

V. DIMITROVA AND D. DICHEVA, "Who is Who": the roles in an intelligent system for foreign language terminology learning, *British Journal of Educational Technology*, **1** (1998), 47–57.

V. DIMITROVA AND D. DICHEVA, Learning terminology in a foreign language, Presented at the *Proceedings of the International Conference Recent Advances in Natural Language Processing*, (1997) Tzigov Chark, Bulgaria.

DICTIONARY OF COMPUTING, Oxford University Press, 1990.

B. DORR , J. HENDLER, S. BLANKSTEEN AND B. MIGDALOFF, Use of Lexical Conceptual Structure for Intelligent Tutoring, In *Intelligent Language Tutors: Theory Shaping Technology* (M. Holland, J. Kaplan, M. Sams, Eds.), (1995) pp. 289–310, Lawrence Erlbaum Associates, New Jersey.

B. DU BOULAY, A Tutorial Survey of Intelligent Computer Assisted Language Learning Systems. Internal Technical Report, School of Cognitive and Computer Sciences, University of Sussex, November 1992.

N. ELLIS. The psychology of foreign language vocabulary acquisition: implications for CALL. *Computer Assisted Language Learning*, **8** (2–3), (1995).

C. FREDERIKSEN, J. DONIN, M. DECARY AND M. HOOVER, Semantic discourse processing and tutoring systems for second language learning, In *Intelligent Tutoring Systems for Foreign Language Learning* (M. Swartz, M. Yazdani, Eds.), (1992) pp. 103–121, Springer–Verlag.

J. HANDKE, WISDOM: a multiple purpose language tutoring system based on AI techniques, In *Intelligent Tutoring Systems for Foreign Language Learning* (M. Swartz, M. Yazdani, Eds.), (1992) pp. 293–306, Springer–Verlag.

J. HIGGINS, *Computers and English Language Learning*, Intellect, Oxford, 1995.

A. HOLMES–HIGGIN AND K. AHMAD, The Machine Assisted Terminology Elicitation Environment: Text and Data Processing and Management in Prolog, Technical report in Knowledge Processing #8, University of Surrey, 1992.

M. HOLLAND, J. KAPLAN AND M. SAMS, Eds., *Intelligent Language Tutors: Theory Shaping Technology*, Lawrence Erlbaum Associates, New Jersey, 1995.

F. INGRAHAM, T. CHANIER AND C. EMERY, CAMILLIE an European project to develop language training for different purposes in various languages on a common hypermedia framework, *Computers and Education*, **23** (1–2), (1994), 107–115.

J. KAPLAN AND M. HOLLAND, Application of learning principles to thedesign of a second language tutor, In *Intelligent LanguageTutors: Theory Shaping Technology* (M. Holland, J. Kaplan, M. Sams, Eds.), (1995) pp. 273–288, Lawrence Erlbaum Associates, New Jersey.

S. KREYER AND E. CRISWELL, Instructor as author in an adaptive, multimedia, foreign language tutor, In *Intelligent Language Tutors: Theory Shaping Technology* (M. Holland, J. Kaplan, M. Sams, Eds.), (1995) pp. 289–310, Lawrence Erlbaum Associates, New Jersey.

L. LEVIN AND D. EVANS, ALICE-chan: a case study in ICALL theory and practice. In Intelligent Language Tutors: Theory Shaping Technology (M. Holland, J. Kaplan, M. Sams, Eds.), (1995) pp. 77–98, Lawrence Erlbaum Associates, New Jersey.

F. MEYER, D. SKUCE, L. BOWKER AND K. ESK Towards a new generation of terminological resources: an experiment in building a terminological knowledge base, Presented at the *Proceedings of COLING'92 — International Conference on Computational Linguistics*, (1992) Nantes.

M. POLSON, AND J. RICHARDSON, Eds., Foundation of Intelligent Tutoring Systems, Lawrence Erlbaum 1988.

J. SOWA, Conceptual Graphs Summary, In *Conceptual Structures: Current Researches and Practice* (T. Nagle, G. Nagle, L. Gerholz, P. Eklund, Eds.), (1992), pp. 3–52. Ellis Horwood.

J. SOWA, *Conceptual Structures: Information Processing in Mind and Machine*, Addison–Wesley, MA, 1984.

M. SWARTZ, Issues for tutoring knowledge in foreign language intelligent tutoring systems, In *Intelligent Tutoring Systems for Foreign Language Learning* (M. Swartz, M. Yazdani, Eds.), (1992) pp. 219–234, Springer–Verlag.

M. SWARTZ AND M. YAZDANI, Eds., *Intelligent Tutoring Systems for Foreign Language Learning*, Springer–Verlag, 1992.

W. V. HAHN AND G. ANGELOVA, Providing factual information in MAT, Presented at the Proceedings of the *International Conference Machine Translation: Ten Years On*, (1994) Cranifeld.

R. WEISCHEDEL, W. VOGE AND M. JAMES, An Artificial Intelligence Approach to Language Instruction. *Artificial Intelligence*, **10** (1978), 225–240.

E. WENGER, *Artificial Intelligence and Tutoring Systems*, Morgan Kaufmann, Los Altos, CA, 1987.

W. WOODS AND J. SCHMOLSE,] The KL-ONE family, In *Semantic Networks in Artificial Intelligence* (L. Lehmann, Ed.), (1992) pp. 133–178, Pergamon Press.

M. YASDANI, *Multilingual Multimedia: Bridging the Language Barrier with Intelligent Systems*, Intellect, Oxford, 1993.

Received: September, 1997
Accepted: February, 1998

Contact address:
Darina Dicheva
Faculty of Mathematics and Informatics
University of Sofia
Bulgaria
e-mail: darinad@fmi.uni-sofia.bg

Vania Dimitrova
Computer Based Learning Unit
University of Leeds, UK
e-mail: V.Dimitrova@cbl.leeds.ac.uk

DARINA DICHEVA is an associate professor at Faculty of Mathematics and Informatics, University of Sofia. Her research interests include application of Artificial Intelligence methods and tools to education, especially intelligent tutoring systems and interactive learning environments; and flexible and distance learning through telematics. URL: http://www-it.fmi.uni-sofia.bg/ darinad/

VANIA DIMITROVA is a PhD student at Computer Based Learning Unit, University of Leeds. Her work is related to developing modifiable learner models in intelligent tutoring systems for learning technical/scientific terminology in a foreign language. URL: http://www.cbl.leeds.ac.uk/ vania/