

International Journal on Artificial Intelligence Tools
© World Scientific Publishing Company

EXAMINING THE USE OF CONCEPTUAL GRAPHS IN ADAPTIVE WEB-BASED SYSTEMS THAT AID TERMINOLOGY LEARNING

KALINA BONTCHEVA*

*Department of Computer Science, University of Sheffield
Regent Court, 211 Portobello Street, Sheffield S1 4DP, UK
kalina@dcs.shef.ac.uk*

VANIA DIMITROVA

*School of Computing, University of Leeds
Woodhouse Lane, Leeds, LS2 9JT, UK
vania@comp.leeds.ac.uk*

Received (22 September 2003)

Revised (12 January 2004)

Accepted (20 January 2004)

This paper discussed the use of Conceptual Graphs (CGs) for implementing tasks employed in web-based educational systems that aid terminology learning. Specifically, we focus on two critical issues in intelligent tutoring - student diagnosis and generation of adaptive explanations. Both tasks are demonstrated in terminological domains where learners have to familiarize themselves with concepts in a specific subject area (e.g. computing, finance, chemistry). Based on CG reasoning, robust and computationally tractable algorithms for student modelling and adaptive explanation generation are defined. Two intelligent systems are presented - STyLE-OLM and HYLITE+. STyLE-OLM is an interactive learner modelling system that extracts extended models of the learners' cognition. HYLITE+ is a natural language generation system that generates adaptive Web pages based on a learner model(LM). The two systems are complementary and have been implemented separately. However, considered together they cover most of the key tasks in adaptive web-based educational hypermedia that aid learning technical terminology. Based on evaluative studies of STyLE-OLM and HYLITE+, the use of CGs for interactive open student modelling and adaptive concept explanations is examined. The applicability of CGs in adaptive web-based systems that aid learning technical terminology is discussed.

Keywords: Conceptual Graphs; Student Modelling, Adaptive Web Explanations, Learning Technical Terminology

*Author names are ordered alphabetically.

1. Introduction

Understanding technical terminology becomes increasingly important nowadays with the expansive technological advancements in all sectors. People need to be aware of the meaning of technical terms to perform routine everyday tasks, e.g. to buy a computer, to open a bank account, to invest money on the stock market, to find the best mortgage. Many of these tasks are frequently carried out over the Internet and it is there that people would seek advice, assistance, explanation, guidance, tutoring, etc. Despite its importance, learning terminology has not been investigated sufficiently in web-based intelligent computer assisted instruction.

Most of the existing computer-assisted learning systems concerned with terminology tend to focus on the user's free exploration of the subject area and function as terminological dictionaries and domain encyclopaedias.^{1,2,3,4} Typically they provide little teaching assistance or guidance. Moreover, due to the system's lack of modelling of users' conceptual understanding and explanation tailoring, it may well be the case that the information provided by the system contains both references to terms the user is not familiar with and redundant explanations of terms the user already knows.

Several computer-assisted language learning projects consider terminology as an instructional domain^{5,6} and explore primarily multimedia technology to simulate realistic life situations. These systems, like the terminological resources mentioned above, suffer from lack of adaptability. A major obstacle to having adaptable terminology tutors is that there is little attention paid to both modelling the learners' cognition in this domain and providing adaptive features in the explanations/feedback generated by the system. These tasks require the application of Artificial Intelligence techniques. More specifically, user modelling techniques should be used to extract advanced student models and sophisticated language processing methods employed to generate adaptive hypermedia.

Adaptive terminology tutors on the web are a kind of web-based intelligent tutoring systems.⁷ Typically in these systems the domain expertise is presented in some knowledge representation formalism, a student model represents the system's view about the student's domain competence, and a pedagogical model incorporates the system's knowledge of how to teach.⁸ Recent developments in the field have shown the importance of robust and reusable mechanisms to accomplish adaptive behavior, and this has led to the adoption of some more systematic, ontology-based approaches.⁹ The definition and use of domain ontology is a critical part of concept-oriented web-based educational systems, an example of which are terminology learning systems. The development of well-specified, tractable algorithms that deal with intelligent tutoring tasks (e.g. student diagnosing, content planning, presentation planning, feedback generation) based on established knowledge representation formalisms would significantly increase software inter-operability and reuse.

In this paper we discuss the use of Conceptual Graphs (CGs)¹⁰ for implement-

ing tasks employed in web-based educational systems that aid learning technical terminology. Specifically, we focus on two critical issues to ensure adaptability in intelligent tutoring - student diagnosis and generation of adaptive explanations. We have developed robust and computationally tractable algorithms based on CG reasoning to implement student modelling and adaptive explanation generation. Both tasks have been demonstrated in terminological domains where a learner/user has to familiarize themselves with concepts in a specific subject area (e.g. computing, finance, chemistry). Student diagnosis is illustrated in STyLE-OLM which implements a novel student modelling approach that enables a student to inspect and discuss the content of the student model. As a result, an advanced and reliable student model is extracted, which can be used by an intelligent tutoring system to adapt to the needs of each individual student. The use of such an advanced student/user model for producing adaptive explanations is demonstrated in HYLITE+ that uses sophisticated natural language techniques to generate adaptive web-based hypermedia. Both systems use the same formalism for representing the domain ontology - Conceptual Graphs. Subsequently, the student/user model is based on CGs as well as algorithms for student modelling and text generation. The two systems are complementary and have been implemented separately. However, considered together they cover most of the key tasks in adaptive web-based educational hypermedia⁷ that aid learning technical terminology. Based on formative evaluation of the two systems, we assess the suitability of CGs for interactive student modelling and adaptive concept explanations. This has enabled us to examine the use of CGs in adaptive web-based systems that aid learning technical terminology.

Next in the paper, Conceptual Graphs are introduced and the reasons for their employment in the the two systems are discussed (Section 2). STyLE-OLM and HYLITE+ are then outlined in Section 3. The following two sections present the use of Conceptual Graphs for interactive student modelling in STyLE-OLM (Section 4) and for adaptive web explanations in HYLITE+ (Section 5). Based on the formative evaluation of the two systems, presented in Section 6, we discuss the use of CGs in intelligent tutoring systems for terminology learning. We also compare the approaches presented here with relevant work on using CGs for intelligent tutoring, user modelling, and text generation (Section 7). In conclusion, we discuss the use of CGs in adaptive web-based educational systems and point out future research directions.

2. Terminological knowledge and Conceptual Graphs

Terminology is defined as “a structured set of concepts and their designations in a particular subject field”,¹¹ and is directly linked to conceptualization. Terminological knowledge is represented as domain ontology that includes concept type taxonomy and a set of axioms that define concept properties, roles, and relations with other concepts.¹ A domain ontology usually includes a concept type taxonomy explicating hierarchical relations between the main classes of concepts, concept

properties and roles as well as internal relations that link concepts to each other.¹²

Conceptual graphs,¹⁰ with their formal structures and operations, appear to be a suitable formalism for constructing domain ontologies. CGs allow concept descriptions using necessary and sufficient conditions as well as structuring the concepts in a hierarchy. Furthermore, CGs provide syntactic clarity together with powerful expressiveness about quantifier processing, which concerns the individual-generic distinction supported by the flexibility of references.¹³ The application of inference rules supports extracting knowledge which is not directly encoded in the domain ontology.

2.1. *Conceptual Graphs: a brief introduction*

Conceptual graphs were designed as a “system of logic that can express the propositional content of sentences in natural language in as simple and direct a manner as possible”.¹⁴ In various projects, CGs have served as an intermediate language that simplifies the mapping of computer-oriented formalisms to and from natural languages. Their graphical representation provides a human readable and understandable representation of formal meaning of sentences. In this section, following Sowa,^{10,15} we will give a summary of the main features of conceptual graphs focussing on those aspects related to the use of CGs in STyLE-OLM and HYLITE+.

A conceptual graph is a finite, bipartite, directed graph that has two types of nodes called *concepts* and *conceptual relations* connected by directed arcs. There are two commonly used forms for presenting CGs: a graphical form (where concepts are represented as rectangles and relations as ellipses) and a linear form (which is used in this paper, see for example Figure 4). Throughout the paper the relations are abbreviated following CG conventions,¹⁰ e.g. CHAR (characteristic), CONT (contain), ISA (is a, for hierarchical links), etc.

Concepts represent natural language “content words” - nouns, verbs, adjectives, and adverbs.¹⁴ Each concept is represented by its concept type, which determines the typical characteristics of the concept, and its referent, which differentiates a particular instantiation of the concept. A concept c is denoted by $[type_c : referent_c]$. Some concepts do not identify a particular individual and are called generic concepts. The referent part for these concepts is omitted, e.g. [OBJECT-ORIENTED LANGUAGE], [ENCAPSULATION], [WASTE WATER], [DISPERSION]. Individual concepts refer to particular individuals, e.g. [OBJECT-ORIENTED LANGUAGE: C++], [OBJECT-ORIENTED LANGUAGE: Java].

Concept types are denotations for sets of individuals and present *typicality*. Concept types are organized in a *hierarchy* according to their level of generality. The hierarchy is a lattice where each concept type is linked with its supertypes. In some CG queries below we use a special concept [T: ?] to denote that the concept is searched for and can be of any type. The hierarchy permits information inheritance: concepts which belong to the same type inherit all characteristics of this type as well as from its supertypes. In contrast with the standard semantic networks, conceptual

graphs provide a clear differentiation between the hierarchical ISA relations between types and the ISA links between a type and its instances and the relations between instances.¹⁶

Conceptual relations indicate that there are certain dependencies between concept meanings. In natural language, these are thematic roles or conjunctions.¹⁴ Each conceptual relation has a type (e.g. CHAR) and a valence that corresponds to the number of arcs. Most relations are dyadic (one incoming and one outgoing arc), as are the ones used in STyLE-OLM and HYLITE+.

Conceptual graphs that represent real or possible situations in the external world are called canonical graphs. An intelligent system's experience about the world is encoded in a collection of canonical CGs. By using formation rules - *restrict*, *join*, *simplify*, and *copy* - new canonical graphs can be derived from already existing ones.¹⁰ The rule *restrict* derives a new graph by replacing a concept c_1 in an existing graph with another concept c_2 where c_2 is either an instance of c_1 (if c_1 is generic) or has a type which is a subtype of the type of c_1 . The rule *join* derives a graph combining two graphs which have a common concept. It may well happen that after joining two graphs some conceptual relations are duplicated in the resulting graph. In this case, the rule *simplify* is applied to delete duplicated relations and include only one relation from a set of identical relations. The rule *copy* derives an identical graph from an existing graph and is often employed to check if two graphs are identical.

The canonical formation rules are *specialization rules*. If a canonical graph G is derived from a graph G_1 by applying formation rules, then G is called a *specialization* of G_1 , and G_1 is called a *generalization* of G . If G is as specialization of G_1 , then there should be a subgraph of G that represents the original G_1 to which joins and restrictions have been applied. The algorithm for obtaining such a subgraph is known as *projection*. Other useful algorithms are those for finding *common generalization* or *common specialization* of two graphs. In the following sections, we describe how projection, specialization, generalization, common specialization, and common generalization have been used in STyLE-OLM and HYLITE+.

2.2. Why STyLE-OLM and HYLITE+ have employed Conceptual Graphs

Due to their suitability for representing terminology, Conceptual Graphs have been employed in both systems discussed here. STyLE-OLM exploits CGs for planning diagnostic dialogue, reasoning about the student's domain beliefs, and assigning a level of correctness to the student's knowledge. HYLITE+ employs CGs for modelling users and defining the content of adaptive web explanations.

Certain characteristics of CGs are essential for interactive open learner modelling and for adaptive web explanations. STyLE-OLM adopted CGs because they are:

- rigorous, permitting reasoning fruitful for interaction planning and student

diagnosis;

- relatively understandable having a clear graphical form useful for learner model inspection and discussion.

HYLITE+ adopted CGs because they:

- support the representation of knowledge at different levels of granularity through use of nested graphs;
- are easily linked to the lexicon entries.

3. Brief description of STyLE-OLM and HYLITE+

3.1. *STyLE-OLM: interactive learner modelling*

STyLE-OLM^{17,18} is the Open Learner Modelling component in STyLE (Scientific Terminology Learning Environment) developed within the *Larflast* project,^a aimed at assisting learners from Bulgaria, Romania and Ukraine in acquiring Finance terminology in English. The users are university students who attend a Financial English course. The role of STyLE-OLM in STyLE is twofold. On the one hand, STyLE-OLM is part the user modelling component in STyLE dealing with the dynamics and the consistency of the learner model. On the other hand, due to its potential to encourage reflection and improve learning,¹⁷ STyLE-OLM is included in the learning activities offered by the instructional planner in STyLE.

STyLE-OLM is a learner modelling tool where a learner model is constructed with the active participation of the learner being allowed to inspect and discuss the content of the model the computer builds of her. The system is based on a framework for interactive open learner modelling¹⁸ and includes:

- a discourse model that manages the interaction between the computer and the user;
- a modal logic mechanism that maintains a jointly constructed user model;
- a graphical communication medium that provides both the computer and the learner with a symmetrical power in maintaining the learner model.

The architecture of STyLE-OLM does not depend on the subject area. Consequently, two instantiations of the system have been developed - in Computing and Finance. The former was used for testing purposes and for the training stage in the evaluative study (see Section 6). The latter was integrated in the STyLE terminology learning environment. The examples below are chosen from the STyLE-OLM instantiation in the Computing domain.

^a*Larflast* was an EU funded project (November 1998 - October 2001) that involved the University of Leeds (UK), the University of Sofia (Bulgaria), the Bulgarian Academy of Sciences, UMIST (Manchester, UK), the University of Montpellier (France), the Romanian Academy of Sciences (Romania), the University of Simferopol (Ukraine), and Virtex Ltd. Details about the project can be found at <http://www-it.fmi.uni-sofia.bg/larflast/>.

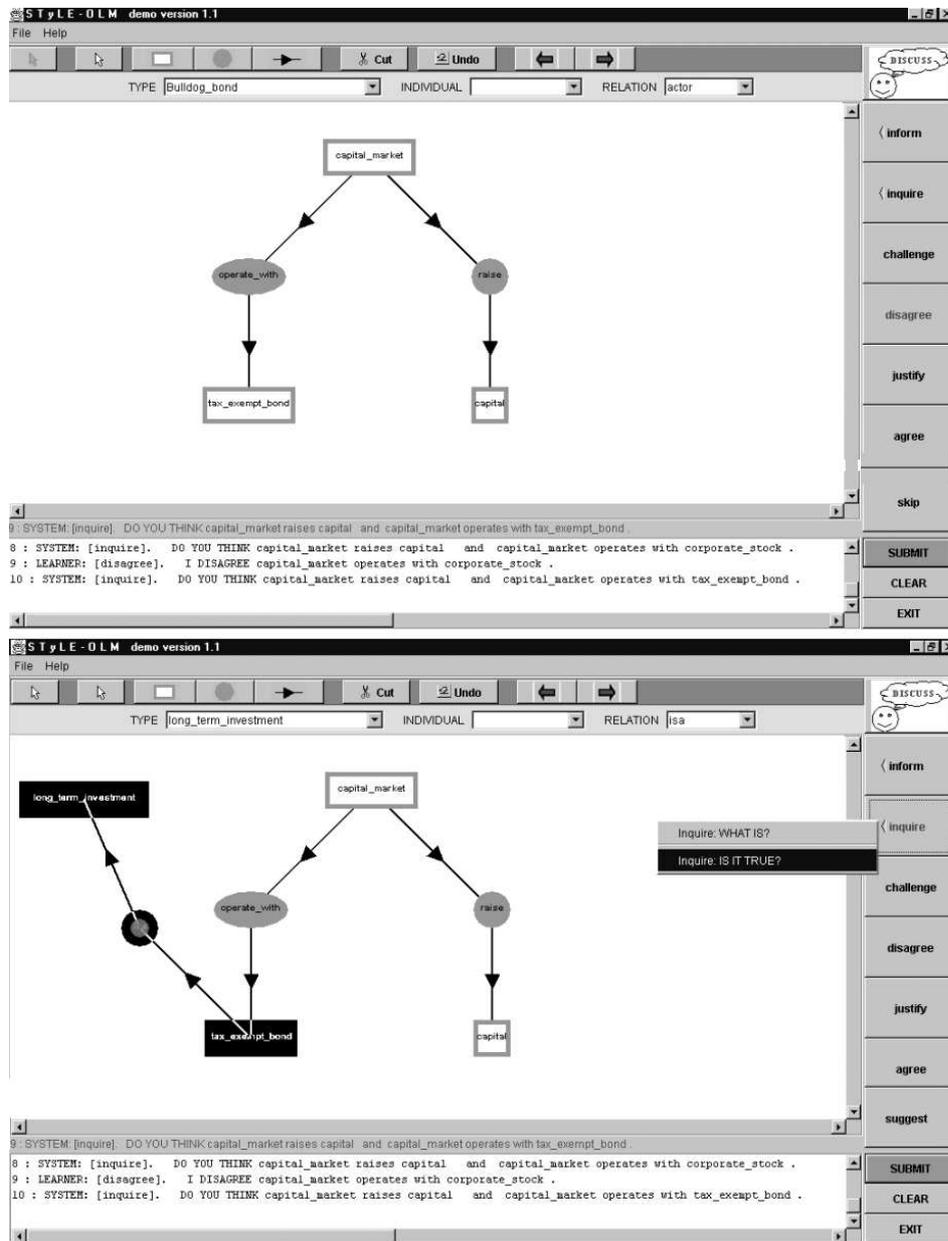


Fig. 1. STyLE-OLM in DISCUSS mode - the system and the learner discuss the learner's domain beliefs. The top screen shows the system asking a question. In order to answer to the system's question, the learner needs to clarify a domain aspect and composes an inquiry, which is presented in the bottom screen.

The interaction in STyLE-OLM includes two modes: DISCUSS, where the learner discusses aspects of her domain knowledge and influences the content of the LM (see Figure 1), and BROWSE, where the learner inspects the current state of her LM (see Figure 2).

The communication medium employs graphically rendered CGs to present discussed domain propositions (in DISCUSS mode) and beliefs from the learner model (in BROWSE mode). Throughout the discussions, the system makes plausible inferences about what is further believed by the learner on the basis of what is explicitly asserted, and from this a dialogue strategy is determined. The dialogue is guided by general explanations of misconceptions, for example, that an entity may be believed to be a member of a class if it has some features of that class. The inspection of the LM provides learner control over the diagnostic process and can trigger further discussions. A resultant LM that incorporates the views of both the computer and the learner about the learner's knowledge is extracted.

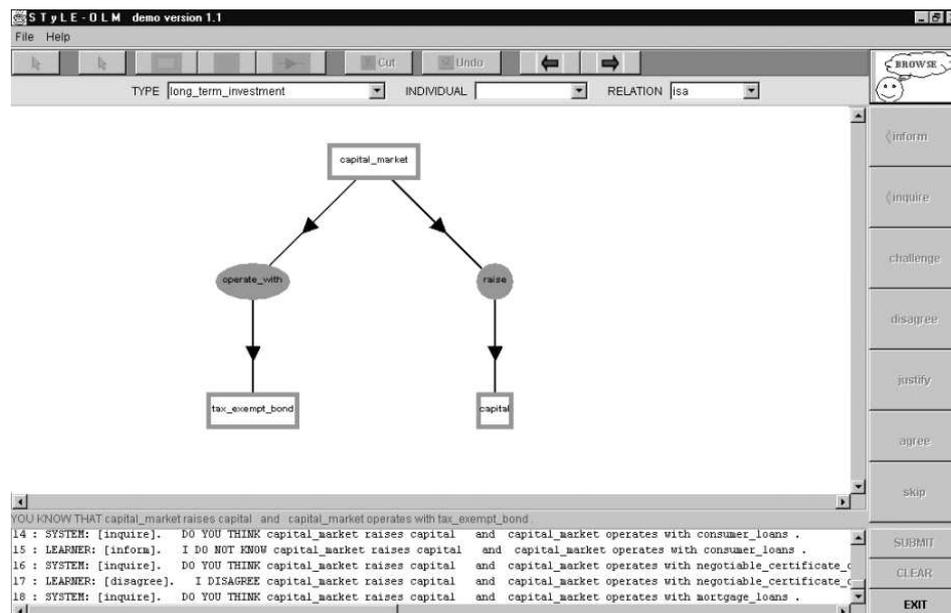


Fig. 2. STyLE-OLM in BROWSE mode - the learner is enabled to browse through the beliefs in the learner model, extracted from the interaction in DISCUSS mode.

3.2. HYLITE+: adaptive web explanations

HYLITE+^{19,20} is a dynamic hypertext system that generates encyclopaedia-style explanations of terms. The approach itself is domain-independent and was tested in two different domains: chemistry and computers. Similar to other dynamic hy-

pertext systems,^{21,22,23} HYLITE+ uses Natural Language Generation (NLG) techniques to create dynamically the hypertext nodes and links.

The user interacts with the system in an ordinary Web browser (e.g., Netscape, Internet Explorer) by specifying a term she wants to look up. Further information about subtypes, parts, and other related concepts is obtained by following hypertext links or specifying another query.

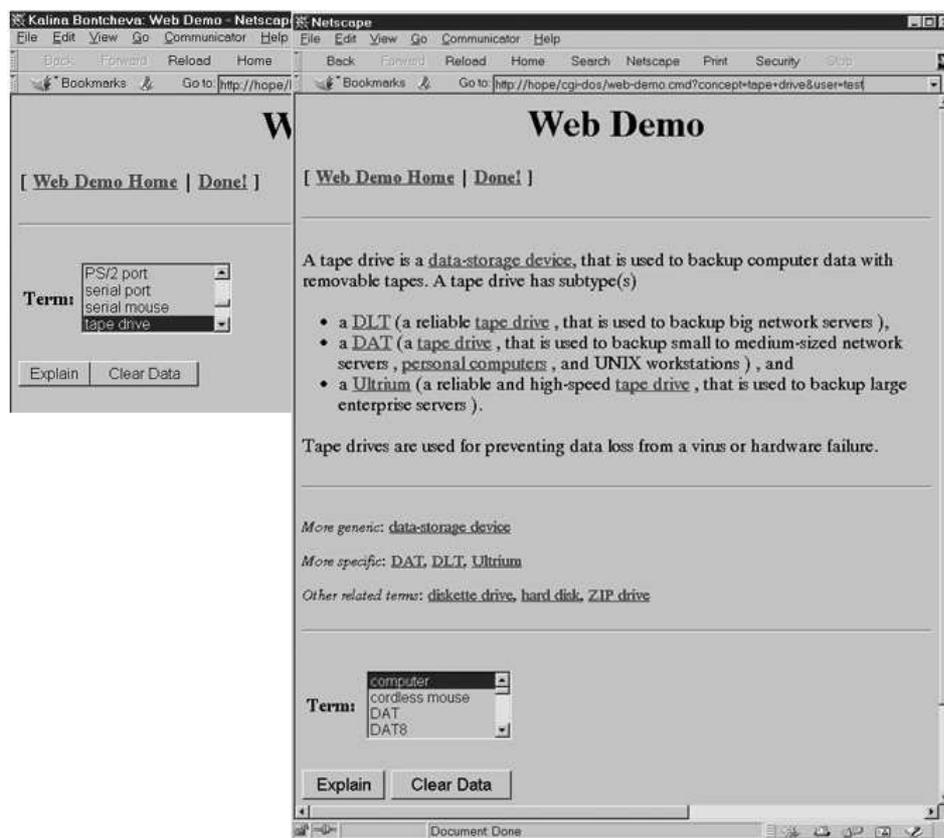


Fig. 3. The system home page (left) and a generated explanation by the adaptive system (right)

The system consists of several modules organized in two main stages: (i) content organization, which includes modules that select the semantic content of the text and organize it in a coherent way; and (ii) *surface realization* modules that generate sentences from the semantic representation (i.e., the conceptual graphs).²⁴ The adaptivity is implemented on the basis of a learner and a discourse models. The learner model is updated dynamically, based on the learner's interaction with the system. When a learner registers with the system for the first time, their model is initialized from a set of stereotypes. The system determines which stereotypes

apply on the basis of information provided by the learner. If no such information is provided, the system assumes a novice learner.

Unlike previous adaptive NLG systems which have their own, application-specific user models, our dynamic hypertext system has re-used a generic agent modelling framework (**ViewGen**) instead.²⁰ Apart from avoiding the development costs of a new model, this also enabled a more modular and extendable system architecture. As argued by Brusilovsky,⁷ such modularity and re-use are much desired in adaptive hypertext systems and one way of achieving that is by using generic user/learner shells, such as BGP-MS²⁵ and **ViewGen**²⁶.

The learner model is used to determine which concepts are unknown, so clarifying information can be provided if appropriate (e.g., parenthetical definitions for parts/subtypes). An example of a hypertext explanation generated by the adaptive system appears in Figure 3. In this example, the user model did not contain these concepts as known, so all three types of tape drive are explained briefly in parenthesis. More detailed information about each one of them is available by following the hypertext links provided.

The model is also used to detect misconceptions, which might come either from a user stereotype or individual's beliefs (see Section 5).

4. Interactive Learner Modelling

4.1. *The learner model and the ontology*

STyLE-OLM imports a domain ontology built with CGs, which includes a taxonomy of concept types and a set of graphs that represent propositions with relations between concept types and individuals. The LM is yielded by an extended overlay method and incorporates:

- *beliefs* that represent domain propositions, and are open for inspection and discussion with the learner;
- *misconceptions* that provide explanations for the learner's errors at a conceptual level, and are used for dialogue planning.

Beliefs are represented with CGs, and can be *correct* (student beliefs that are supported by the expert), *erroneous* (student beliefs that are not supported by the expert), and *incomplete* (missing beliefs that the expert believes but the learner does not possess them yet). Misconceptions in StyLE-OLM apply to error patterns, such as *misclassification* (a concept has been wrongly assigned to a class) or *misattribution* (an attribute has been wrongly assigned to a concept). These are defined as bug rules. For instance, misclassification rules can be applied when an erroneous $isa(i, t)$ is discovered. One type of misclassification rule describes that an individual i has been wrongly considered as a member of a concept type t - $isa(i, t)$ - because i has features $[A]$ that are part of the definitional features for t , e.g. Visual Basic $[i]$ is an Object-Oriented Language $[t]$ because it contains objects $[A]$.

Based on the bug rules, CG schemata are defined to determine conditions that STyLE-OLM has to examine in subsequent dialogue in order to confirm a misconception, e.g. the misclassification error above will trigger the CG schema:

```
Error : isa(i,t).
Type : Misclassification_1.
Conditions_to_Check = []
G_i = find_graph(i)
G_t = definition(t)
A = generalisation_subst(G_i, i, G_t, t)
A_i = restrict(A, theta, i)
A_t = restrict(A, theta, t)
Conditions_to_Check = add([A_i, A_t], Conditions_to_Check)
Return : Conditions_to_Check
```

This schema will be applied when the learner makes a wrong statement, for example.

L. I think that Visual Basic is an object-oriented language.

Following the schema, STyLE-OLM first searches both the CGs from the domain ontology and the CGs that represent the learner's beliefs for a proposition about t (object-oriented language). Then, a *generalization* is found which is a common feature of t and i (Visual Basic) - they both contain objects. Restricting the generalization to i and t , the schema will return that a possible explanation for the learner's erroneous belief may be that

the learner believes that:

'Visual Basic contains objects' and 'Object-oriented languages contain objects'.

In the following dialogue, STyLE-OLM will check these beliefs by asking appropriate questions, i.e. the next move will be:

STyLE-OLM. Do you think that Visual Basic contains objects?

If the learner does not confirm this belief:

L. I don't think so.

the system will search for another misclassification rule to explain the erroneous belief 'Visual Basic is an object-oriented language'. When such a rule is applied, the system discovers that the learner might believe that Visual Basic is an object-oriented language because both it and Visual C++, which is an object-

oriented language, allow programming in a visual environment. The system will then check the conditions for this misclassification:

STyLE-OLM. Do you think that Visual C++ is an object-oriented language?

4.2. *Maintaining diagnostic dialogue*

The dialogue in STyLE-OLM is organized as a series of dialogue games, which represent dialogue episodes and follow diagnostic tactics. There are three types of game.

Explain learner errors. The system initiates such games to look for reasons that might have caused a learner's erroneous beliefs. CG schemata based on misconception rules are used for planning dialogue content, as shown above.

Explore domain knowledge. These games aim to gather as much as possible information about the learner's domain beliefs. They are initiated either by the learner when she changes the dialogue focus making a claim or asking a question or by the system when it probes for a learner's beliefs in order to decide how to react to her errors. To plan interaction content, the system extracts from the ontology CGs that contain focus concepts. It generates more CGs by applying specialization rules (replacing concept types with descendants from the hierarchy). For example, following that

isa(Java, Object-oriented language),

the system infers that

'Java contains objects and classes and has characteristics inheritance and encapsulation'.

This proposition will be included in a dialogue game that explores the learner's knowledge of object-oriented languages. Note that the system can either ask a question or make a statement with this proposition depending on the current state of the LM.

Negotiate learner model. These games aim at clarifying agents' positions when discrepancies in their views are discovered. Such games may be initiated by both parties when asking the other side for justifications, e.g. the student may challenge the validity of the computer's beliefs about the student's knowledge, while the computer may ask for clarification when a learner's statement is not confirmed by the domain ontology.

A graph G constructed by the learner is confirmed if G is a *specialization* of a graph from the domain ontology or if G can be *projected* into a graph from the domain ontology. The example in Figure 4 shows two graphs G_3 and G_4 that are confirmed from the domain ontology because G_3 is a *specialization* of G_1 and G_4 is a *projection* of G_2 , where G_1 and G_2 are from the domain ontology. However, the graphs G_5 and G_6 are not confirmed by the domain ontology.

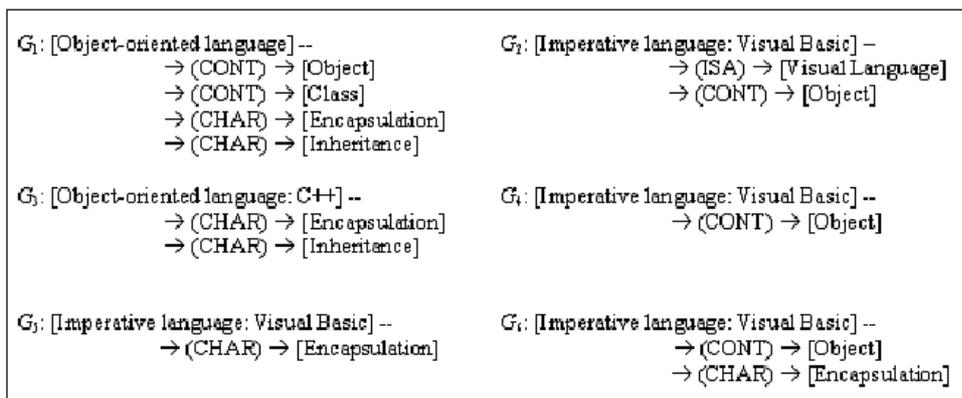


Fig. 4. Example for CG confirmation in STyLE-OLM

Dialogue coherence in STyLE-OLM is maintained through a focus mechanism. Each dialogue game has a focus space - list of related concepts that can be discussed in the game to assure the dialogue is coherent. Related concepts are defined as either directly linked in the taxonomy through `isa` links (e.g. object-oriented language, Java, C++) or belonging to a common CG (e.g. object-oriented language, inheritance, encapsulation).

The relevance of a learner's utterance to the dialogue game focus is verified by comparing the concepts in her proposition with the game focus space. If there is no intersection, STyLE-OLM concludes that the learner has suggested a change in focus and an explanatory dialogue game is initiated (if appropriate) to discuss domain propositions related to the new focus. When the learner stays on focus STyLE-OLM follows the tactic of the current dialogue game but rearranges the propositions to be discussed, so that the ones that are most relevant to the student's move will be discussed first.

4.3. Answering questions

STyLE-OLM enables a learner to ask questions about domain facts. Two question types are catered for:

- *what-questions* (the user asks for a domain concept with certain properties)
- *is-true-questions* (the user checks the validity of a proposition).

Giving learners opportunities to clarify aspects of the domain promotes concept learning and meta-cognition. For example, when a learner's erroneous claim:

L. I think that C is an object-oriented language.

is challenged by the system:

STyLE-OLM. Why do you think that C is an object-oriented language?

the validity of the learner's beliefs is questioned. She searches for possible clarifications, such as the definition of the concept type:

L. What are the characteristics of an object-oriented language?

This question is encoded in a CG query form as:

```
(cg) [OBJECT-ORIENTED LANGUAGE] ->  
      (CHAR) -> [ T:? ]
```

The answer is a projection of the query upon the graph with the definition of the concept type object-oriented language (graph G_1 above).

STyLE-OLM. Object-oriented languages have characteristics inheritance and encapsulation?

Alternatively, the learner may compose a proposition and check its validity, for example:

L. Is it true that C has objects?

To answer this question STyLE-OLM employs a mechanism for confirming a graph, similar to the one used for focus maintenance, as described above.

4.4. *Updating the learner model*

Throughout the dialogue the beliefs of the system and the learner about the learner's knowledge are accumulated in separate belief stores. When the interaction finishes, the belief stores are combined and a resultant learner model is extracted (by using some modal logic plausible reasoning).²⁷

Each belief in the resultant model, represented with a CG, is assigned a degree of correctness:

- *correct* (if confirmed by the domain ontology)
- *erroneous* (if not confirmed by the ontology)
- *incomplete* (the CG is in the ontology but is not believed by the learner).

An evaluative study of STyLE-OLM with postgraduate students at Leeds University (see Section 5) showed the potential of the approach for improving the quality of the learner model and providing the means for reflection.¹⁷ Some participants in the study stressed the need of computer explanations of domain terms generated in a natural language and tailored to the learners' knowledge. This vital component of intelligent tutoring systems that aid learning technical terminology is illustrated in HYLITE+.

5. Generating adaptive Web explanations

5.1. *The learner model and the ontology*

The learner model and the ontology are the basis of the generated adaptive hypertext explanations. HYLITE+ uses the **ViewGen** user modelling framework²⁶ to access and update the user beliefs as the interaction progresses. The propositions believed by each user are encoded as CGs and the CG reasoning mechanisms are used to detect whether or not the learner already holds a given belief.

ViewGen distinguishes the beliefs of each learner by putting them in separate environments, including one for the system itself which contains the domain ontology. The motivation for holding the domain facts as system beliefs, instead of being outside the learner modelling component, is that the system can treat them in the same manner and also deal with incomplete system beliefs.

The separation between system and user belief environments also enables the modelling of discrepancies between propositions believed by the system and those believed by the learner, i.e., support for *erroneous* and *incomplete* beliefs as in STyLE-OLM. One example of such discrepancies is taxonomical knowledge, where the system has detailed domain taxonomy while the learner has only partial and/or incorrect representation. Such discrepancies might come from common misconceptions encoded as stereotypes (e.g., common mistake about photographic emulsion) or be specific to the given learner, in which case they are stored exclusively in her belief space. Misconceptions from stereotypes are ascribed dynamically to all learners who conform to that stereotype.

In this way, differences between the domain ontologies of different agents are easily encoded. **ViewGen**'s operations take such differences into account and can reason correctly about the learner's beliefs when these are affected by different ontologies. For example, when determining the beliefs of a chemistry expert about photo emulsion, **ViewGen** also considers propositions inherited from gels and sols. However, by default for laymen, **ViewGen** takes into account the default stereotype and includes beliefs inherited from emulsion instead.

5.2. *Generating the explanations*

When the user requests information about a topic (e.g., emulsion), HYLITE+ needs to select a set of relevant propositions, from which the generation algorithms produces the hypertext page in the following manner:

- Obtain relevant propositions from the domain ontology given the user's question about a concept.
- Determine which CG propositions are known already by the learner (see below) and mark them as such. In case of contradictory beliefs between system and learner, both system and learner beliefs are added to the propositions to be conveyed, e.g. $bel(user, isa(photo_emulsion, emulsion))$ and

bel(system, isa(photo_emulsion, gel)).

- Inspect each proposition for unknown related concepts and mark them as such.

Unknown concepts in definitions, as well as unknown properties, parts, and super-/sub-concepts, trigger the inclusion of additional information (usually their definition), e.g.:

A personal computer is a type of microcomputer^b (a small computer that uses a microprocessor as its central processing unit).

The learner model is also used to detect partially known facts (equivalent to *incomplete* beliefs in STyLE-OLM) which are then verbalized by the generator using contextual phrases such as *besides* and *as well as*. For example, if the learner has already seen an explanation of the concept TAPE DRIVE and follows the link to STORAGE DEVICE, then the description of the types of storage devices is generated using such a phrase:

As well as tape drives, other types of data-storage devices are ...

Incorrect beliefs, detected in the learner model, are explained by providing them in parallel with the correct fact. In the photographic emulsion example shown above, the system generates the following sentence as part of the explanation^c:

A common error is to believe that photographic emulsion is emulsion, whereas, in fact, it is a gel.

5.3. Consulting the learner model during generation

Given the set of relevant propositions selected for generation, the generator first establishes whether any of them are already believed by the learner. **ViewGen** is queried about each proposition and all already believed CGs are annotated as such. After all believed CGs are identified, the system also checks for any partially known propositions. This is necessary because the generator frequently uses the CG projection operation to present only part of a given conceptual graph (to make the explanation shorter). For example, one of the graphs extracted for an explanation of the DISPERSION concept is:^d

```
(cg2) [DISPERSION] <- (CHAR) <- [WASTE_WATER]. :fs(derived_from: cg1)
```

^bThe underlined words are hypertext links to a Web page providing their definitions and other information. These pages are generated only if requested.

^cThe “a common error is...” expression is used to convey the uncertainty of this assumption which has been made on the basis of a stereotype. The generator can also deal with incorrect beliefs of individual users.

^dfs() is used to encode features which can be associated with concepts, relations and graphs. In this example the feature specifies which graph was used in the derivation of the graph it is associated with.

This is in fact a subgraph of the following:

```
(cg1) [WASTE_WATER] -  
      (CHAR) -> [CONCENTRATION]  
      (CHAR) -> [RESISTANCE]  
      (CHAR) -> [DISPERSION].
```

Therefore, if the learner asks later about WASTE_WATER, the generator extracts the whole graph cg1. Naturally ViewGen then returns that this fact is not already believed by the learner since the learner belief environment only contains graph cg2.

Such partially known facts are identified by the generator in the following way:

- Obtain from ViewGen all propositions already believed by the learner.
- Check every obtained graph for a `derived_from` feature. If found, check whether the graph from which it was derived is among the graphs selected for explanation. Then annotate each such parent graph (e.g., cg1) as partially known (see example below).
- Use subsumption to determine whether some of the graphs returned as already believed by the learner subsume parts of a graph selected for explanation. If so, annotate the appropriate part as already known and the whole graph as partially known.

In our example, at this step graph cg2 will be identified as already believed by the learner and also derived from cg1. So cg1 is annotated as partially known and the relevant parts of it are annotated as already believed^e. The result is:

```
[WASTE_WATER] -  
  (CHAR) -> [CONCENTRATION]  
  (CHAR) -> [RESISTANCE]  
  (CHAR):fs(um_state: known) ->[DISPERSION].  
      : fs(um_state: partially_known)
```

In other words, a new feature `um_state` is added to the whole graph and also all relations which also occur in the already explained graph. This information is then used by the generator to choose a phrasing which conveys that dispersion has been mentioned earlier and makes the generated hypertext more coherent. Finally, ViewGen is also consulted about concepts that appear in the selected graphs. If no learner beliefs can be obtained about these concepts, then the generator annotates them with a `(um_state: unknown)` feature, which is used to determine what additional information needs to be provided for this concept, e.g., its definition in parenthesis.²⁴

^eThese added features do not alter the conceptual graph in any way and can be ignored by other applications which want to use the CGs for other purposes. Also the features are removed by the application as soon as the text is generated. In this way the graph can be annotated differently in later explanations if the user beliefs have changed.

6. Evaluation

Empirical evaluations of the two systems have been conducted to estimate pitfalls and outline potentials of the frameworks so that they can be employed in intelligent learning environments. In general, the focus of this paper is on CGs and their appropriateness. Due to space limitations many details about the methodologies and results, unrelated to CGs, are omitted. Full descriptions of the evaluations of the two systems are available elsewhere.^{17,19}

Adaptive systems are typically evaluated following a modular approach, i.e. components are assessed separately in several stages before the impact of the integrated system is measured as a whole.²⁸ There are two phases in the evaluation of intelligent tutoring systems - formative and summative evaluation.²⁹ The former focuses on the system performance, while the latter assesses the usefulness of the overall approach, which is typically stated in the system objectives. Summative evaluation is appropriate once the main development is completed and a fully fledged tutoring system exists. In order to thoroughly assess the educational impact of STyLE-OLM and HYLITE+, both systems need to be integrated within a learning environment. Nonetheless, even at this component stage we have been able to investigate some issues of the computational and educational benefits of our approaches.

The evaluative studies of STyLE-OLM and HYLITE+ addressed formative evaluation and verified the functionality of the main system components. In addition, some aspects of summative evaluation have been addressed to assess whether STyLE-OLM and HYLITE+ meet their objectives. Specifically, the STyLE-OLM evaluation focused on the quality of the obtained learner model, which is a critical factor to assess the effectiveness of a student modelling tool,²⁹ and the means provided for meta-cognitive activities, promoted by open learner modelling.³⁰ The HYLITE+ evaluation focused both on the quality of the generated text and on the effectiveness of adaptation. The latter is the key factor which is typically quantified by measuring human performance on a set of tasks, given different versions of the system.^{31,32}

6.1. *The evaluation of STyLE-OLM*

An evaluative study was conducted primarily to examine the behavior of STyLE-OLM, specifically the use of the communication medium, the dialogue management, and the learner model maintenance. The study also allowed us to examine the benefits of the system in terms of improving the quality of the learner model and providing means for reflective learning. The study was done in the Finance domain and was part of the evaluation of the overall STyLE environment, developed within the *Larflast* project. The domain ontology imported in STyLE-OLM included a type hierarchy with ninety six concept types, and twenty five CGs. The ontology was developed by the Bulgarian team and used in the *Larflast* project.⁸

The study involved nine postgraduate students at the Computer Based Learning Unit, Leeds University. Most participants were novices in Finance, only one of

them appeared to have a substantial understanding of the domain from his past experience. All participants participated on a voluntary basis. In addition, an expert - a teacher in the Finance domain - helped with the evaluation of the dialogue maintenance and the validation of the learner models.

Each learner attended an individual session. A week before the session, each learner was given an introductory text from the topic Financial markets to familiarize with. Prior to their sessions, the learners were asked to answer several drill questions about the terms they studied. This took about twenty minutes. The learners' performance was assessed and initial user models obtained (by hand). The interactions with STyLE-OLM were initiated using these models. Training with the communication medium was provided which was generally half an hour and allowed the students to interact with the instantiation of STyLE-OLM in the Computing domain.

In an experimental session with STyLE-OLM, ran in the Finance domain, the participants were asked to help the computer system to obtain a better model of their domain conceptualizations, which would facilitate adaptability in pedagogical situations like generating explanations, providing feedback, selecting instructional materials. The learners were encouraged to inspect their learner models, discuss their domain knowledge, and influence the content of the learner model. On average, these interactions lasted about half an hour and were terminated by the learners. The sessions with STyLE-OLM were observed and monitored by the experimenter. Log files were recorded to examine the system's behavior.

At the end of each session, the participants were given a questionnaire, with the dialogue transcripts and the resultant learner models enclosed, to elicit their subjective view about the system's performance. Some learners were briefly interviewed to explore interesting issues arising in the interactions. A questionnaire with enclosed dialogue transcripts, initial and obtained learner models from STyLE-OLM sessions was given to a teacher who was experienced in teaching Financial English to non-English students (all participants but one in the study were non-native speakers). The teacher analyzed the suitability of the dialogues generated by STyLE-OLM as well as a possible learning effect.

The evaluative study showed that the sessions with STyLE-OLM led to a *better quality learner models*. It was found that the resultant student models:

- contained more learners' beliefs, due to the fact that learners were actively engaged in the construction of the student model, which resulted in the expansion of the model.
- included less invalid beliefs, since the model was constantly validated by the users.
- removed inconsistencies in the learner model because when such inconsistencies were discovered STyLE-OLM initiated a clarification dialogue that enabled inconsistent information to be removed and more facts related to the uncertain aspects of the student's knowledge added.

- contained some explanations of learners' misconceptions, which refer to student reasoners included in advanced student model.³³

The above show that STyLE-OLM deals with the dynamics of the user model, a problem common for most traditional user modelling methods.

In general, the CG mechanisms used to maintain the student model (which are described in Section 4) were appropriate. However, at times the resultant student models contained conceptually conflicting domain facts somehow hidden and not captured by the CG inference engine employed. For instance, the STyLE-OLM inference mechanism did not recognize that two graphs represented facts that were paraphrases of one another. The following extract illustrates such contradictions.^f

Teacher-expert: *I do not understand how the learner does not know 212 and 213, and yet knows 211 which is a paraphrase of 212 and 213. Actually, 211 combines the statements made in 212 and 213. Please look at that in the initial learner model.*

In order to capture different perspectives in propositions presented with CGs, more advanced CGs reasoning is needed. There are some recent attempts to tackle this problem³⁴ but they assume that the graphs are constructed by experts (i.e. they are correct) while in tutoring systems some graphs constructed by the student may indeed be erroneous.

Throughout the study a number of explanatory dialogue games aimed at discovering possible learners' misconceptions were initiated following the CG schemas presented in Section 4, while only a few misconceptions were actually registered in the resultant learner models. The analysis of the interactions revealed that the learners frequently discovered their errors and made claims to change the student model. As a result, registered potential misconceptions were not confirmed, i.e. not included in the resultant model. We also discovered that learner's misconceptions were not confirmed because learners changed the flow of the dialogue by initiating new conversational topics. We consider, however, that enabling the learners to take the initiative in maintaining the interaction is an important feature that promotes engagement in reflective activities (see below).

Opening the learner model in STyLE-OLM helped to overcome some deficiencies in dialogue management - at every time the learners could look at their models and initiate dialogue to correct beliefs that were assigned wrongly. The users found the dialogue moves useful and did not experience major problems to participate in the interactions. The interactions with STyLE-OLM were classified by most of the users as coherent discussions about their domain knowledge. The focus main-

^fNumbers here indicate IDs of conceptual graphs as follows: 211 - 'Primary market operates with (trades with) new issues of security and supports new investments', 212- 'Primary market is the process by which a corporate stock is issued for the first time', 213- 'Primary market supports new investment through selling newly issued stocks'.

tenance (based on CG inference) was relatively robust and allowed discussing connected terms. The analysis of the dialogues showed that the scope of articulated domain propositions was extended in a coherent manner so that learners could build a consistent picture connecting related domain facts. It was also found that the students were provided with various alternatives to explore relevant domain facts. The learners were referred to term definitions, situations with domain terms, hierarchical relations, and exemplars of generic terms. However, in some cases the interactions were slightly boring because the system generated specializations of the same graph exploring predominantly exemplars of generic terms or hierarchical relationships due to the limited number of conceptual graphs in the knowledge base. This problem points at the importance of dealing with incomplete domain expertise, mentioned later in the discussion in Sections 7 and 8.

The analysis of the interactions with STyLE-OLM showed that the learners were engaged in *activities that promote reflective thinking*.¹⁷ The process that the learners went through when interacting with STyLE-OLM was recorded in interaction transcripts that combined the dialogue history and the experimenter's observations. These were analyzed to identify reflective engagement based on Educational Psychology literature. The following reflective activities[§] were observed:

- *Activity 1* - The students rendered statements about their domain beliefs, thus they externalized their knowledge and experience as well as recalled and reconsidered domain aspects.³⁵ Typical examples of such activities involved dialogue episodes containing inform, agree, challenge, or disagree moves from the learners.
- *Activity 2* - The students went back to claims about their beliefs and (sometimes) changed these claims, thus they recalled and reconsidered domain aspects³⁵ and validated their domain beliefs³⁶. These activities included (a) dialogue episodes (e.g. the system challenged a learner's claim and the learner made a statement to alter the challenged claim); (b) the learners made statements to change their previous claims after going through the dialogue history; (c) being provoked by a system's move (e.g. inquire or challenge) learners switched to BROWSE mode to inspect their models.
- *Activity 3* - The students investigated arguments to support their beliefs, thus they searched for the grounding of their beliefs.³⁶ In this group of reflective activities we considered the learners' justifications of their claims (including justify or inform moves) in response to system's challenges. Another evidence for the students' searching for the grounding of their beliefs were episodes when a system's challenge was followed by a student's inquiry to clarify domain aspects and then a student's claim that supports/withdraws her beliefs.

[§]Activity is defined as a fragment of the interaction that can be considered to have led to the fulfilment of a particular goal.

We found that interactions with STyLE-OLM were beneficial both for more knowledgeable and less knowledgeable learners. While the former were engaged in reflective interactions about the domain, the later were provoked by the system to inspect their models and challenge the robustness of their domain knowledge. Metacognitive activities encourage deep conceptual understanding and lead to meaningful learning.^{35,36}

6.2. *The Evaluation of HYLITE+*

First, we carried out performance-based evaluation of the the adaptive version of HYLITE+ by comparing it to the non-adaptive baseline one (a version that does not include a user model). This revealed that generation of additional information in adaptive explanations only adds between 5% and 20% to the overall execution time – e.g., 5-10% for additional information which is less than 33% of the baseline explanation; 20% for additional information which is 50-60% of the baseline one. With both systems the overall generation time for half a page of hypertext is less than 1.5 seconds (on a PIII 800MHz PC). These results show that the user modelling framework and the associated adaptivity algorithms do not result in a substantial performance overhead.

Next, the acceptability and utility of the adaptive features were evaluated by users who interacted with two versions of our system: a baseline one and the adaptive one. The two versions were, in fact, the same system with the user model and adaptivity features switched off to form the non-adaptive baseline. In this way, we ensured that the same information was available in both systems. Also, this approach minimizes the potential influence of different systems' response times on the experiment results, because both versions generate hypertext on the fly and have similar performance.^h

The initial system page, which was identical in the two versions, contained an alphabetical list of topics, relevant to the experimental tasks (see Figure 3 left). The users requested an explanation on a given topic by selecting it and clicking **Explain**. After reading the generated page, the participants could obtain further information by following links, navigating back and forward with the browser buttons, or by selecting a new term in the topic list (see Figure 3 right).

The explanations generated by the adaptive version were enriched with parenthetical definitions of unknown terms, examples, and links to other more general, specific, and related terms (see Figure 3 right).

Hypermedia applications are often evaluated with respect to: *interface look and feel*, *representation of the information structure*, and *application-specific information*.³⁷ The information structure is concerned with the hypertext network (nodes and links) and navigation aids (e.g., site maps, links to related material,

^hThe ILEX system, for example, used pre-generated static pages as a baseline and the study reported that the difference in the two systems' response times might have influenced some of the results.³¹

index). The application-specific information concerns the hypermedia content – text, images, etc. For our system there is no need to evaluate the interface, since HYLITE+ generates simple HTML and uses Web browsers as rendering tools. Therefore, the evaluation efforts were concentrated on the information content and navigational structure of the generated hypertext. In addition, observation, questionnaires and informal interviews were used to elicit problems and assess the acceptability of the implemented adaptivity techniques.

Information content was measured on the basis of:

- average *time to complete* each task;
- average number of *pages visited* per task;
- average number of *distinct pages* visited per task;
- percent of *correctly answered questions* per task;
- questionnaire results about *content* and *comprehension* of the generated pages;
- *user preference* for any of the systems.

The **navigational structure** was measured by the following metrics:

- average *time per page visited*;
- average *number of pages visited*;
- *total number of pages visited*;
- number of *links followed*;
- usage of the browser **Back** button;
- usage of the *system's topic list* to find information;
- observation and subjective *opinion on orientation*;
- subjective *opinion on navigation* and ease of finding information.

The experiment has a *repeated measures, task-based* design (also called within-subjects design), i.e., the same users interacted with the two versions of the system, in order to complete a given set of tasks. Prior to the experiment, the participants were asked to provide some *background information* (e.g., computing experience, familiarity with Web browsers, and electronic encyclopaedia) and fill in a *multiple choice pre-test*, that diagnosed their domain knowledge.

The design of the tasks follows the design used in the evaluation of two other adaptive hypermedia applications.^{32,37} Each of the participants was first given a set of three tasks – each set contained one browsing, one problem-solving, and one information location task. The order was not randomized, because the browsing task was also intended as a task that would allow users to familiarize themselves with the system and the available information; it was not used for deriving the quantitative measures discussed above.

Qualitative data and feedback were obtained using a *questionnaire* and *semi-structured interviews*, where the subjects could discuss their experience with the two systems. There were two main types of questions and statements: those related to the usability of the adaptive and baseline systems, e.g., statements like “I found

the adaptive system difficult to use”; and those related to hypertext and navigation, e.g., links, text length, structure.

The tasks were completed by eight participants – three male and five female. They were assigned randomly to the system they used first: four started with the adaptive and four - with the non-adaptive system. After completing the first three tasks, the users swapped systems for the other three tasks. None of the users had any prior experience with the system, but they all had extensive experience with Web browsing and hypertext. All users had previously used on-line dictionaries and encyclopaedia, which are the types of systems most similar to ours. The participants also had similar computing background, i.e., they all had computer science degrees and were either postgraduate students or research staff at the Computer Science department.

With respect to knowledge of the computer hardware concepts, used in the evaluation tasks, the pre-test differentiated two groups of participants: *complete novices*, who scored between 2 and 4 out of the maximum 15 points on the test; and users with *basic knowledge*, who scored 7 to 8. The two types of users were split equally between the two groups: adaptive-first and non-adaptive first.

Due to the small sample size (8 subjects) and the differences in users’ prior domain knowledge and browsing styles, the results obtained could not be used to derive a statistically reliable comparison between the measures obtained for the adaptive and the non-adaptive versions, but the quantitative results and user feedback are sufficiently encouraging to suggest that HYLITE+ adaptivity is of benefit to the user.¹⁹

In brief, these experiments showed that the participants found the adaptive system easy to use and the generated hypertext intuitive to navigate. The generation techniques were sufficiently fast, so users did not have to wait for the system response. All users felt comfortable with both the adaptive and the non-adaptive versions. None of them stated that either system was unusable. In addition, the majority of the users preferred the adaptive system, where they also felt it was easier to perform the tasks.

6.3. Summary

The evaluation of STyLE-OLM and HYLITE+ confirmed the suitability of conceptual graphs for interactive student modelling and adaptive explanations in terminological domains. The dialogue in STyLE-OLM was handled in a coherent manner, more aspects of the students’ domain knowledge were revealed, some reasons for erroneous beliefs were discovered. Along this line, the users working with HYLITE+ appreciated the relevance of the domain content of the explanations produced, and found the adaptive features adequate and beneficial.

The main advantage of CGs was shown to be their intuitive graphical representation which could be understood by the learners after some training and enabled them to interact visually with the model built by the system (see Section 3.1). The

CG reasoning operations were also used heavily in both systems. From the point of view of generating adaptive explanations, the projection operation was particularly useful, because it extracts sub-graphs thus enabling shorter sentences. The STyLE-OLM evaluation also revealed that more advanced CG reasoning is needed in some cases in order to address the problem of different perspectives.

It must be stressed that in both systems CGs were used in combination with another logical framework that handled the maintenance of the learner models and included some commonsense reasoning over the agents' beliefs. We found that CGs were best suited for representing propositional knowledge and an additional reasoning level was added in both systems to reason about agents' beliefs (which commonly included nested beliefs and non-monotonic reasoners).

The learner models we have described here incorporate learner's beliefs and misconceptions that provide possible explanations of patterns of learner errors. Conceptual graphs on their own are not sufficiently flexible to allow such misconceptions to be represented, therefore we use CGs only as means for representing and communicating about domain knowledge and employ a meta-level commonsense reasoning to infer about learner beliefs.

7. Related Work

In this section we revisit the algorithms employed in STyLE-OLM and HYLITE+ to compare them with similar approaches developed in other projects. Due to space limitations, the review is focused predominantly on CG-based systems, although selected related work in intelligent tutoring systems, language generation, and adaptive hypermedia is also covered.

Several algorithms in STyLE-OLM and HYLITE+ compare two graphs. This task is known as matching conceptual graphs and finding the similarity between them. The predominantly used CG matching method, also employed by us, is to consider a common generalization of the two graphs.³⁸ Depending on the aims of the matching algorithm, different forms of common generalization have been exploited. For example, a minimal common generalization (the most specific of the possible generalizations) has been used in an analogy tool³⁹ and in a case-based tutoring system;⁴⁰ a most interesting common generalization (the common generalization that maximizes a particular interest function) has been used in a matching algorithm proposed by Poole and Cambell.⁴¹ In order to implement practical mechanisms for graph matching, some systems combine the common generalization with other techniques. For instance, the task of finding the similarity between different views is tackled by Delugach⁴² through joining graphs around counterparts to overcome problems with the insufficient specification level of the common generalization. Dieng and Hug⁴³ examine the same problem by exploring a generalization and conceptualization. The latter considers relations between instantiations of the graph concepts.

Managing the diagnostic dialogue in STyLE-OLM entails deciding about the

relevance of the learner's contribution, presented with a CG, to a set of focus concepts. We have considered a comparison between both sets - the graph concepts and the focus concepts - based on concept correspondence, which is a simplified approach. Possible improvements can take into account the fact that the task of finding the relevance between concept sets is a descendant of the broader task of defining the semantic distance between two concepts.^{43,44}

In order to answer questions posed by the learner, a projection operation has been used both in STyLE-OLM and HYLITE+. This operation is commonly used for query search in a CG Knowledge Base (KB).¹⁵ Natural language generation systems utilizing CGs have also used projection in a manner similar to HYLITE+.^{45,46}

Several tasks in STyLE-OLM and HYLITE+ include confirming the truth value of a CG on the basis of the facts represented in a CG ontology. This task shares some similarity with two other tasks used in CG projects: inference with conceptual graphs and integrating new knowledge into an existing CG ontology. Sowa⁴⁸ suggests the use of specialization, for inferring new facts from ontology with canonical conceptual graphs. The same method has been used in the algorithms presented above. A possible improvement of the efficiency of the CG reasoning is discussed by Clark and Porter⁴⁹ who suggest the use of access-paths to specify predefined chains of subgoals to each inference goal and to control the incompleteness. Integrating new knowledge to an existing ontology is tackled in projects that build shared ontology incorporating different experts' views.^{43,50} These projects define specialization and projection in a slightly different way from the conventional CG definitions and rely on experts' reliability. While useful in managing ontologies when experts are involved, these approaches should be applied with caution in tutoring systems where learner's expertise might not always be reliable.

Modelling beliefs and misconceptions plays an essential role in adaptive tutoring. Various definitions for learner's misconceptions have been explored. Our definition is similar to that of Sison and Shimura who describe misconceptions as: " incorrect or inconsistent facts, procedures, concepts, principles, schemata or strategies that result in behavioral errors ".⁵¹ Detecting misconceptions has been studied in some early intelligent tutoring systems (e.g. ^{52,53}) as well as in work on student modelling (e.g. ^{54,51}). It typically concerns the automatic bug (malrules) library generation. There has also been natural language generation research on explaining, and possibly correcting, misconceptions once they have been discovered.⁵⁵ The systems presented here contribute to research on modelling beliefs and misconceptions by showing how this can be done utilizing CGs.

Recently, some research has been carried out to build user models employing CGs. Nenkova and Angelova⁵⁶ present a user model based on CG actors that is used for building an adaptive language generation system. Baldwin et al.⁵⁷ propose a user modelling mechanism that can build user profiles by observing users' interaction with a computer system. While these projects consider modelling a user's cognitive state merely on a belief level, the learner models we have described in this paper incorporate learner's beliefs and misconceptions that provide possible

explanations of learner's erroneous beliefs. Following this, the use of CGs planning dialogues aimed at discovering learner's misconceptions and at generating adaptive explanations to correct these misconceptions is original. We have shown that CGs can be used to represent a deeper cognitive level (not merely beliefs as in the above projects).

Despite their popularity, CGs have been scarcely employed in intelligent tutoring systems. Zock⁵⁸ demonstrates how CGs can be used in a tutoring system for learning a foreign language to represent semantics of natural language sentences. Fung and Adam⁵⁹ and then Fung and Kemp⁴⁰ deploy CGs to measure similarity between problem examples in a case-based physics problem solver. Recently, Strupchanska et al.⁶⁰ use CGs in a 'self-tutoring system' that generates questions to the user from given text lessons, analyzes the learner answers and generates adaptive feedback. All sentences in the lessons have to use Controlled English, which is the case with terminological texts. This approach is the closest to those of STyLE-OLM and HYLITE+ in terms of dealing with technical terminology, and demonstrates other intelligent tutoring components - automatic generation of courseware and personalized feedback - not discussed in this paper. The methods discussed in this paper utilize CGs to maintain two important tasks - student modelling and adaptive explanations - and demonstrate a contribution to the few adaptive tutoring systems that employ CGs.

In the context of similar open student modelling approaches, STyLE-OLM demonstrates a novel approach to tackle the interaction with the student when the student model is open for influence from the user. The issue of interaction has not been explicitly addressed in the existing computational architectures that involve users in diagnosis. In TAGUS,⁶¹ UM,⁶² PACMOD,⁶³ and VisMod⁶⁴ the system's claims are rendered in viewers that externalize the student model and the learners are provided with some command options to form their statements about the content of the model. The learners can ask for explanations and justifications of the computer's opinions. The Mr Collins system⁶⁵ provides more enhanced interaction means in a menu-based environment for negotiating the learner model. The negotiation is initiated when conflicts between the system and learner's views about the learner model arise. None of these diagnosers utilizes discourse models and the diagnosee (the learner) is the only one who has any modelling of the interaction; the learner's participation is strictly delimited by the restricted set of command options or menu choices that the diagnostic system provides.

In the context of previous work on text generation, the approach developed in HYLITE+ is novel because it re-uses a generic user modelling framework (i.e., general purpose UM shell) - **ViewGen**, which can be shared with other parts of the system, e.g., dialogue planning and understanding. In comparison, other NLG systems which employ user models in order to generate tailored output^{21,22,23,66} tend to use custom-implementations of overlay user models where facts from the generator's knowledge base can be marked as known by the user. Differences and conflicts in user and system belief are often not represented with a few exceptions.⁵⁵

Typically the representational power and structure of the user model are chosen to be sufficient for the task at hand. The advantage of the HYLITE+ approach is that it allows sharing of the user model with other components of the tutoring system. In addition, because the system uses an independent user modelling framework, it can benefit from new advances in user modelling such as distributed access and privacy.⁶⁷ Another advantage of using a user modelling framework is that it can support both simpler, overlay-style models and more complex ones, including conflicting attitudes, user goals, and plans. This flexibility makes it easier to re-target HYLITE+ to new applications, without need for further implementation or fine-tuning of the user modelling mechanism. As demonstrated by the performance evaluation results, the use of **ViewGen** for the generation of adaptive hypertext did not result in a substantial processing overhead, due to the ‘inference on demand’ approach to building and maintaining of the belief environments.

Unlike previous NLG-based approaches, other adaptive hypermedia systems have used successfully such generic user modelling frameworks. The most similar system is KN-AHS⁶⁸ which uses the general purpose user modelling shell system BGP-MS.²⁵ In terms of representational power, BGP-MS is very similar to **ViewGen**: both frameworks model domain knowledge (BGP-MS uses the KL-ONE formalism), support stereotypes, individual user models, and are capable of representing conflicts. The main difference between HYLITE+ and KN-AHS is that HYLITE+ employs language generation techniques to produce automatically the hypertext content and links, while KN-AHS uses human-written texts. Since HYLITE+ uses a well-defined protocol to communicate with **ViewGen**, this effectively de-couples the generation system from the concrete user modelling framework and makes it possible to replace **ViewGen** with other UM shells (e.g., BGP-MS,²⁵ UM toolkit⁶²) as long as they offer similar functionality. In other words, the novel aspects of HYLITE+ are not in the user modelling approach that it employs, but in the way user modelling is used to generate adaptive hypertext explanations, given a CG knowledge base.

8. Conclusion

This paper discussed the use of Conceptual Graphs (CGs)¹⁰ for implementing tasks employed in web-based educational systems that aid terminology learning. Specifically, we focus on two critical issues in intelligent tutoring - student diagnosis and generation of adaptive explanations. Based on CG reasoning, robust and computationally tractable algorithms for student modelling and adaptive explanation generation were developed. Both tasks have been demonstrated in terminological domains where learners have to familiarize themselves with concepts in a specific subject area (e.g. computing, finance, chemistry). Student diagnosis is illustrated in STyLE-OLM which implements a novel student modelling approach that enables a student to inspect and discuss the content of the student model. As a result, an advanced and reliable student model is extracted, which enables the intelligent

tutoring system to adapt to the needs of each individual student. The use of such an advanced student model for producing adaptive explanations is demonstrated in HYLITE+. StyLE-OLM and HYLITE+ are complementary – when considered together they cover the two most critical issues that in web-based intelligent educational systems, aimed at terminology learning.

The paper also presented the results from the formative evaluation of STyLE-OLM and HYLITE+, which enabled us to measure the suitability of CGs for open student modelling and generation of adaptive explanations. The main advantage of CGs was shown to be their intuitive graphical representation which could be understood by the learners after some training and enabled them to interact visually with the model built by the system. The CG reasoning operations were also used heavily in both systems.

Both approaches presented here suppose that domain ontology provides all the necessary factual knowledge. However, this may not always be the case in learning systems. Often, tutoring knowledge is incomplete (lacking a particular piece of information), vague (there are concepts that are difficult to define), ambiguous (it may be hard to distinguish between several possibilities for a learner's reasoning). Further investigation is needed to address these aspects in CG-based tutoring systems, which may consider similar studies in ontology.⁶⁹

Another strand of ongoing work is concerned with studying the applicability of dynamically generated adaptive explanations in the medical domain, as part of the e-science project MIAKT. Although this work is still at a relatively early stage, our experience so far shows that language generation techniques developed originally to aid terminology learning can be applied with little modification to this new domain.

Acknowledgements

Work on HYLITE+ is currently being supported by the UK Engineering and Physical Sciences Research Council as part of the MIAKT project (grant GR/R85150/01), which involves the University of Southampton, University of Sheffield, the Open University, University of Oxford, and King's College London. Previous support included a University of Sheffield PhD fellowship and an Overseas Research Students Award. The first author also wishes to thank her PhD supervisor Yorick Wilks.

The development of STyLE-OLM was supported by the EU funded Larflast project and an Overseas Research Students Award. Work on interactive open learner modelling was conducted by the second author during her PhD studies at the University of Leeds – the supervision of John Self and Paul Brna is greatly acknowledged.

Both authors wish to thank the anonymous referees whose comments and suggestions helped us improve this paper. Last but not least, we thank the human evaluators who participated in the evaluative studies of the two systems.

References

1. J. Sager, *A practical course in terminology processing*. John Benjamin, Amsterdam/Philadelphia (1990).

2. J. Smith, *Welcome to everyone's ABC of accounting language*. URL: <http://www.soc.staffs.ac.uk/cmjrjs/agloss.htm>.
3. Wordnet. *Wordnet the English Language*. URL: <http://www.cogsci.princeton.edu/wn/>
4. Stanford. *Stanford KSL Network Services Webster Gateway*. URL: <http://WWW-KSL-SVC.stanford.edu:5915/WEBSTER/>
5. D. Berleant, L. Shi, X. Wei, K. Viswanathan, C. Chai, N. Majid, Y. Qu and P. Sunkara, *LEARN: software for foreign language vocabulary acquisition from English unrestricted text*, Computer Assisted Language Learning. **10 (2)** (1998).
6. T. Chanier, *Learning a second language for specific purposes within a hypermedia framework*, Computer Assisted Language Learning. **9 (1)** (1998) 3–43.
7. P. Brusilovsky, *Adaptive hypermedia*, User Modelling and User-Adapted Interaction, **11(1–2)** (2001) 87–110.
8. G. Angelova, A. Nenkova, S. Boycheva and T. Nikolov, *CGs as a knowledge representation core in a complex language learning environment*, Proceedings of International Conference on Conceptual Structures. Shaker Verlag (2000).
9. L. Aroyo and D. Dicheva, *Concepts and Ontologies in Web-Based Educational Systems*, Proceedings of Workshop held in conjunction with ICCE2002, Auckland, NZ (2002).
10. J. Sowa, *Conceptual Structures: Information Processing in Mind and Machine*, Addison-Wesley, MA (1984).
11. C. Galinski, and G. Budin, *Terminology*, in Survey of the State of the Art in Human Language Technology, eds. R. A. Cole et al. <http://www.cse.ogi.edu/CSLU/HLTsurvey/HLTsurvey.html>
12. N. Fridman and C. Hafner, *The state of the art in ontology design: a survey and comparative review*, AI Magazine, **18(3)** (1997) 53–74.
13. B. Biebow and G. Chaty, *A comparison between conceptual graphs and KL-ONE*, Conceptual graphs for knowledge representation: Proceedings of the First International Conference on Conceptual Structures, eds. G. W. Mineau, B. Moulin and J. Sowa, Springer, Berlin (1993) 75–89.
14. J. Sowa, *Towards the expressive power of natural language*, Principles of semantic networks explorations in the representation of knowledge, ed. J. Sowa, Morgan Kaufmann, San Mateo, California (1991) 157–190 .
15. J. Sowa, *Conceptual graphs summary*, Conceptual Structures: Current Research and Practise, eds. T. Nagle, J. Nagle, L. Gerholz and P. Eklund, Ellis Horwood (1992) 3–51.
16. G. Mann, *Conceptual graphs for natural language representation*, Technical report 9311. School of Computer Science and Engineering, University of New South Wells (1993).
17. V. Dimitrova, *STyLE-OLM Interactive Open Learner Modelling*. International Journal of Artificial Intelligence in Education, IOS Press, **13** (2002).
18. V. Dimitrova, *Interactive Open Learner Modelling*, PhD thesis, Computer Based Learning Unit, Leeds University (2001).
19. K. Bontcheva, *Adaptivity, Adaptability, and Reading Behaviour: Some Results from the Evaluation of a Dynamic Hypertext System*, Proceedings of the Second International Conference on Adaptive Hypertext (AH2002), Springer, Berlin (2002).
20. K. Bontcheva, *Tailoring the Content of Dynamically Generated Explanations*, User Modelling 2001: 8th International Conference, eds. M. Bauer, P.J. Gmytrasiewicz and J. Vassileva, Lecture Notes in Artificial Intelligence 2109, Springer, Berlin (2001).
21. M. Milosavljevic, A. Tulloch and R. Dale, *Text generation in a dynamic hypertext*

- environment*, Proceedings of the 19th Australian Computer Science Conference, Melbourne (1996).
22. A. Knott, C. Mellish, J. Oberlander and M. O'Donnell, *Sources of flexibility in dynamic hypertext generation*, Proceedings of the 8th International Workshop on Natural Language Generation (INLG'96) (1996) 151–160.
 23. L. Ardissono and A. Goy, *Dynamic generation of adaptive web catalogs*. Adaptive Hypermedia and Adaptive Web-Based Systems, eds. P. Brusilovsky, O. Stock and C. Strapparava, number 1892 in Lecture Notes in Computer Science, Springer, Berlin (2000) 5–16.
 24. K. Bontcheva and Y. Wilks, *Dealing with dependencies between content planning and surface realisation in a pipeline generation architecture*, Proceedings of the International Joint Conference in Artificial Intelligence (IJCAI'2001), Seattle, USA (2001).
 25. A. Kobsa and W. Pohl, *The user modelling shell system BGP-MS*. User Modelling and User-Adapted Interaction, **4(2)** (1995) 59–106.
 26. A. Ballim and Y. Wilks, *Artificial Believers*. Lawrence Erlbaum Associates, Hillsdale, NJ (1991).
 27. V. Dimitrova, J. A. Self and P. Brna, *Maintaining a jointly constructed student model*. Artificial intelligence: Methodology, systems, and applications, eds. S.A. Cerri and D. Dochev, Springer, Berlin (2000) 221–231.
 28. S. Weibelzahl and G. Weber, *Evaluating the Inference Mechanism of Adaptive Learning Systems*. User Modelling: Proceedings of the 9th international conference, eds. P. Brusilovsky, A. Corbett and F. de Rosis, Springer, Berlin (2003) 154–162.
 29. M. Mark and J. Greer, *Evaluation methodologies for intelligent tutoring system*. Journal of Artificial Intelligence in Education, bibbf 4(2/3) (1993) 129–154.
 30. S. Bull, P. Brna and V. Dimitrova, *Learner modelling for reflection*: Proceedings of workshop held in conjunction with AIED03, Sydney, Australia (2003).
 31. R. Cox, M. O'Donnell and J. Oberlander, *Dynamic versus static hypermedia in museum education: an evaluation of ILEX, the intelligent labelling explorer*, Artificial Intelligence in Education: Open Learning Environment: New Computational Technologies to Support Learning, Exploration and Collaboration, eds. S.P. Lajoie and M. Vivet, IOS Press, Amsterdam (1999) 181–188.
 32. K. Höök, *Evaluating the utility and usability of an adaptive hypermedia system*. Knowledge-Based Systems, **10** (1998) 311–319.
 33. V. Dimitrova, J.A. Self and P. Brna, *Applying Interactive Open Learner Models to Learning Technical Terminology*. Proceedings of User Modelling. Springer, Berlin, (2001).
 34. M. Ribiere and R. Dieng-Kuntz, *A viewpoint model for cooperative building of an ontology*, Conceptual structures: interactions and interfaces, eds. U. Priss, D. Corbett and G. Angelova, Springer, Berlin (2002) 220–234.
 35. D. Boud, R. Keogh and D. Walker, *What is reflection in learning*, Reflection: turning experience into learning, eds. D. Boud, R. Keogh and D. Walker, London, Kogan Page (1996) 7–17.
 36. J. Dewey, *How we think - a restatement of the relation of reflective thinking to the educational process*. D.C. Heath and Company (1960).
 37. G. B. Wills, I. Heath, R. Crowder and W. Hall, *User evaluation of an industrial hypermedia application*. Technical report, M99/2, University of Southampton (1999).
 38. T. Nagle, J. Esch and G. Mineau, *A notation for conceptual structure graph matchers*, Conceptual structures: Current researches and practice, eds. T. Nagle, G. Nagle, L. Gerholz and P. Eklund, Ellis Horwood (1992) 585–593.

39. D. Leishman, *An analogical tool based on evaluations of partial correspondences over conceptual graphs*, *Conceptual structures: Current researches and practice*, eds. T. Nagle, G. Nagle, L. Gerholz and P. Eklund, Ellis Horwood (1992) 349–360.
40. P-W. Fung and R. Kemp, *On developing case-based tutoring systems with conceptual graphs*, *Conceptual structures: Standards and practices*, eds. W. Tepfenhart and W. Cyre, Springer, Berlin (1999) 214–229.
41. J. Poole and J.A. Campbell, *A novel algorithm for matching conceptual and related graphs*, *Conceptual structures: Applications, implementation and theory*, eds. G. Ellis, R. Levinson and W. Rich, Springer, Berlin (1995) 293–307.
42. H. Delugach, *Analysing multiple views of software requirements*, *Conceptual structures: Current researches and practice*, eds. T. Nagle, G. Nagle, L. Gerholz and P. Eklund, Ellis Horwood (1992) 391–410.
43. R. Dieng and S. Hug, *MULTIKAT: a Tool for comparing knowledge of multiple experts*, *Conceptual structures: theory, tools, and applications*, eds. M-L. Mugnier and M. Chein, Springer, Berlin (1998) 139–153.
44. N. Foo, B.J. Garner, A. Rao and E. Tsui, *Semantic distance in conceptual graphs*, *Conceptual structures: Current researches and practice*, eds. T. Nagle, G. Nagle, L. Gerholz and P. Eklund, Ellis Horwood (1992) 149–154.
45. G. Angelova and K. Bontcheva, *DB-MAT: Knowledge acquisition, processing and NL generation*, *Conceptual Structures: Knowledge Representation as Interlingua*, eds. P. Eklund, G. Ellis and G. Mann, number 1115, *Lecture Notes in Artificial Intelligence*, Springer, Berlin (1996) 115–129.
46. N. Nicolov, C. Mellish and G. Ritchie, *Sentence generation from conceptual graphs*, *Proceedings of the 3rd International Conference on Conceptual Structures (ICCS'95)*, number 954, *Lecture Notes in Artificial Intelligence*, Springer, Berlin (1995).
47. O. Guinaldo and O. Haemmerle, *Knowledge querying in the conceptual graph model: The RAP module*, *Conceptual structures: theory, tools, and applications*, eds. M-L. Mugnier and M. Chein, Springer, Berlin (1998) 287–294.
48. J. Sowa, *Conceptual graphs: Draft proposed American national standard*, *Conceptual structures: Standards and practices*, eds. W. Tepfenhart and W. Cyre, Springer, Berlin (1999) 1–65.
49. P. Clark, and B. Porter, *Using access paths to guide inference with conceptual graphs*, *Conceptual structures: fulfilling Peirce's dream*, eds. H. Delugach, M. Keeler, D. Lukose, L. Searle and J. Sowa, Springer, Berlin (1997) 168–182.
50. M. Ribiere, *Using viewpoints and CGs for the representation and management of corporate memory in concurrent engineering*, *Conceptual structures: theory, tools, and applications*, eds. M-L Mugnier and M. Chein, Springer, Berlin (1998) 94–110.
51. R. Sison and M. Shimura, *Student modeling and machine learning*. *International Journal of Artificial Intelligence in Education*. **9** (1998) 128–158.
52. R. Burton and J. Brown, *Diagnostic models for procedural bugs in basic mathematical skills*. *Cognitive Science*. **2** (1978) 155–191.
53. W.L. Johnson and E. Soloway, *Intention-based diagnosis of program errors*, *Proceedings of the 2nd national conference on Artificial Intelligence* (1984).
54. Y. Kono, M. Ikeda and R. Mizoguchi, *THEMIS: A nonmonotonic inductive student modelling system*. *Journal of Artificial Intelligence in Education*, **5(3)** (1994) 371–413.
55. K. F. McCoy, *Generating context-sensitive responses to object-related misconceptions*. *Artificial Intelligence*, **41(2)** (1989) 157–195.
56. A. Nenkova and G. Angelova, *User modelling as an application of actors*, *Conceptual structures: theory, tools, and applications*, eds. M-L. Mugnier and M. Chein, Springer, Berlin (1998) 287–294.

- tual structures: Standards and practices: 7th International conference on conceptual structures, eds. W. Tepfenhart and W. Cyre, Lecture Notes in Artificial Intelligence 1640, Springer, Berlin (1999) 83–89.
57. J. F. Baldwin, T. P. Martin and A. Tzanavari, *User modeling using conceptual graphs for intelligent agents*, Conceptual structures: logical, linguistic, and computational issues: 8th international conference on conceptual structures, eds. B. Ganter and G.W. Mineau, Springer, Berlin (2000).
 58. M. Zock, *SWIM or Sink: The Problem of Communicating Thought*, Intelligent Tutoring Systems for Foreign Language Learning, eds. M. Swartz and M. Yazdani, Springer-Verlag (1992) 235–248.
 59. P-W. Fung and A. Adam, *Conceptual graphs for similarity measurement in a case-based physics problem solver*, Proceeding of AIED'95, ed. J. Greer (1995) 170–177.
 60. A. Strupchanska, M. Yankova and S. Boytcheva, *Conceptual Graphs Self-Tutoring System*, Proceedings of ICCS-2003, 11th International Conference of Conceptual Structures, Lecture Notes in Artificial Intelligence, Springer, Berlin (2003).
 61. A. Paiva and J.A. Self, *TAGUS - a user and learner modelling workbench*. User Modeling and User-Adapted Interaction, **4** (1995) 197–226.
 62. J. Kay, *The UM toolkit for cooperative user modelling*. User Modelling and User-Adapted Interaction, **4(3)** (1995) 149–196.
 63. R. Morales, *Exploring participative learner modelling and its effects on learner behaviour*. PhD Thesis, University of Edinburgh (2000).
 64. J-D Zapata-Rivera and J. Greer, *Inspecting and visualising distributed Bayesian student models*, Proceedings of the 5th International conference on Intelligent Tutoring Systems, eds. G. Gauthier, C. Frasson and K. VanLehn, Springer, Berlin (2000) 544–553.
 65. S. Bull, *Collaborative student modelling in foreign language learning*. PhD Thesis, University of Edinburgh (1997).
 66. J. D. Moore, *Participating in Explanatory Dialogues*. MIT Press, Cambridge, MA, (1995).
 67. J. Fink, A. Kobsa and J. Schreck, *Personalized hypermedia information through adaptive and adaptable system features: User modeling, privacy and security issues*, Intelligence in Services and Networks: Technology for Cooperative Competition, eds. A. Mullery, M. Besson, M. Campolargo, R. Gobbi and R. Reed, Springer, Berlin (1997) 459–467.
 68. A. Kobsa, A. Nill and J. Fink, *Hypertext and hypermedia clients of the user modelling system BGP-MS*, Intelligent Multimedia Information Retrieval, ed. M. Maybury, MIT Press (1997).
 69. F. Da Silva, W. Vasconcelos, D. Robertson, V. Brillhante, A. de Melo, M. Finger and J. Augusti, *On the insufficiency of ontologies: problems in knowledge sharing and alternative solutions*. Knowledge-Based Systems. **15** (2002) 147–167.
 70. E. Wenger, *Artificial Intelligence and Tutoring Systems*. Morgan Kaufman (1987).