Supporting teachers as content authors in intelligent educational systems

Peter Brusilovsky*

School of Information Sciences
University of Pittsburgh
135 North Bellefield Avenue
Pittsburgh, PA 15260, USA
E-mail: peterb@pitt.edu
*Corresponding author

Judith Knapp

European Academy Bozen/Bolzano (EURAC)
Institute of Specialised Communication and Multilingualism
Viale Druso 1, 39100 Bozen/Bolzano, Italy
E-mail: jknapp@eurac.edu

Johann Gamper

Free University of Bozen-Bolzano
Dominikanerplatz 3, 39100 Bozen, Italy
E-mail: gamper@inf.unibz.it

Abstract: Intelligent Educational Systems (IESs) need large amounts of educational content that is typically not provided by the creators of these systems. In this paper we discuss a new approach for authoring practical IESs where core authoring is done by professional design teams, while the educational content is mainly developed by teachers who use the system in their classes. The major bottleneck of this approach is the lack of intelligent authoring support tools that allow regular teachers to author intelligent content that an IES needs in order to perform its functions. As a contribution to solving this problem, we present our recent work on authoring support for an adaptive vocabulary acquisition system, ELDIT. The paper describes the ELDIT system, the needs and challenges of language content authoring by teachers, and the two authoring support components that we have developed for two essential kinds of language learning content: illustrative examples and educational texts.

Keywords: intelligent tutoring system; Intelligent Educational System (IES); content authoring; authoring support; language learning.

Reference to this paper should be made as follows: Brusilovsky, P., Knapp, J. and Gamper, J. (2006) 'Supporting teachers as content authors in intelligent educational systems', *Int. J. Knowledge and Learning*, Vol. 2, Nos. 3/4, pp.191–215.

Biographical notes: Peter Brusilovsky received his PhD degree in Computer Science from the Moscow State University in 1987. Since that time, he participated in the development of many intelligent educational systems including ELM-ART, a winner of 1998 European Academic Software Award. Dr. Brusilovsky is currently an Associate Professor of Information Science and Intelligent Systems at the University of Pittsburgh. In the past, he was also holding visiting faculty appointments at the Moscow State University (Russia), Sussex University (UK), Tokyo Denki University (Japan), University of Trier (Germany), Free University of Bolzano (Italy) and Carnegie Mellon University (USA). He is a recipient of Alexander von Humboldt Fellowship and NSF CAREER award, a board member of several journals, and a co-editor of *Technology, Instruction, Cognition and Learning*. He is also the current President of User Modeling Inc., a professional community of user modelling researchers.

Judith Knapp was born in Brunico (Italy). From 1991 to 1998, she studied Mathematics at the Graz University of Technology. She received her Masters degree in November 1998. In October 2004, she received her PhD from the University of Hannover. Her main work was the development and implementation of a new approach to Computer Assisted Language Learning (CALL), focusing on providing linguists with new tools to author an innovative dictionary-based language learning system (ELDIT). Currently, she is involved in several educational projects, which are all strongly focussed on technology-based language learning.

Johann Gamper received his MSc in Computer Science from the Technical University of Vienna in 1989 and a PhD in Computer Science from the RWTH Aachen in 1996. From 1996–2000, he was Senior Researcher at the European Academy Bolzano-Bozen. From November 2000 to September 2003, Gamper was project leader for the establishment of a new Faculty of Computer Science at the Free University of Bozen-Bolzano, where he is Assistant Professor since October 2003. The main research interests are in temporal databases, e-learning, and e-government. He is the author of over 60 publications in national and international journals and conference proceedings.

1 Introduction

1.1 The content problem in Intelligent Educational Systems (IESs)

More and more web-based Intelligent Educational Systems (IESs) (Brusilovsky and Peylo, 2003) are reaching the point when they can be used in the context of real education, the area that is now almost exclusively served by traditional non-intelligent Web-Based Educational (WBE) systems (Brusilovsky and Miller, 2001). Thanks to years of research, the problem of how to represent the domain model, procedural expertise, knowledge about the student, as well as interface development can now be solved in a number of domains by relatively small research teams. The choice of the web as an implementation platform can help a small team to solve problems of delivery, installation, and maintenance and make their intelligent systems available to hundreds and thousands of students. Yet, there is ‘the last barrier’. The traditional static, non-intelligent WBE systems and courses have something that almost no intelligent system developed by a small research team can offer – large amounts of diverse educational material. A high-quality traditional WBE course may have thousands of presentation pages, and hundreds of other fragments of learning material – examples, explanations, animations, and objective questions created by a team of instructional developers. In comparison, the number of presentation items in even the most intelligent WBE systems is well under 100 and the number of other fragments of learning material, such as problems or questions, is no more than a few dozens. These numbers are certainly sufficient for a serious classroom study of the system, but are still quite far from the practical needs of web-based education, *i.e.*, the ability to support reasonable portions of practical courses as they are taught to large numbers of students, semester by semester.

The origin of this bottleneck lies within the established approach to the design of intelligent educational systems. With this approach, a system is created by a team of expert developers and shipped to their users (teachers and students) as a whole. Within this approach, little can be done to magnify or adapt the volume of available educational content. One reasonable solution is to develop some authoring tools to help the designers create educational content more quickly (Murray, 1999). Another interesting approach is to use the authored fragments of educational content for multiple purposes, *i.e.*, an authored problem can be presented to the student as both an example to learn from and as a problem to be solved (Brusilovsky, 1992b; Ritter *et al.* 1998b). These two approaches can certainly help to increase the amount of available educational material, but they can not provide the required dramatic increase. By virtue of its purpose, a system design team is mainly concerned and challenged with the core functionality of an intelligent system. The developers have little time and motivation to create hundreds of content items. In addition, being far from real classrooms, there is less of a chance that they will be able to create really versatile and comprehensive content that will satisfy different teachers who are the prospective users of the system.

We think that the solution to this problem is quite clear: moving an Intelligent Educational System (IES) from the research lab to regular classrooms must be supported by a change in the design paradigm. A significant increase in the role of teachers as principal users of intelligent educational systems must be supported by an increase in their participation in the authoring process. We argue that this new paradigm should transform teachers into more active players in the authoring process by clearly separating the authoring process into two parts: core IES authoring and educational content authoring. Core authoring should comprise the development of the core functionality of an IES: knowledge representation, algorithms, interfaces, and core educational content.[1] This part is not different from traditional authoring and should remain in the hands of a professional development team (which may include champion teachers). At the same time, the core of an IES should be designed in a way to allow the majority of educational content (such as explanations, examples and problems) to be authored by teachers working independently of the development team (and possibly long after the system was originally developed).

The idea to involve teachers as educational content authors comes naturally to many developers of practical IESs that have been used in dozens of classrooms. It is not surprising that the first implementation of this idea by Ritter *et al.* (1998a) was done in the context of the PAT Algebra Tutor, the first IES that made a leap from the lab to hundreds of classrooms (Koedinger *et al.*, 1995). Later, this idea was also explored in the context of AnimalWatch (Arroyo *et al.*, 2001), another practical algebra tutoring system. This solution appears to be a silver bullet. Not only does it solve the lack of content bottleneck, but it also offers multiple additional benefits. The possibility of contributing their favourite content transforms teachers from passive users of the new technology into active coauthors. It turns an IES from being a teacher's potential competitor into a powerful tool in the teacher's hands. It gives the teachers a feeling of controlling the new technology and is more likely to make them active proponents of it. While the above may be true for almost any advanced educational technology, in the context of knowledge-rich IESs, it is also important that the experience of authoring content enables the teachers to better understand the knowledge embodied in the system and how the system operates (Ritter *et al.*, 1998a). Being an author is important for many teachers. A strong feature of traditional non-intelligent WBE systems is that they allow teachers to author, with merely a moderate effort, their own educational content. An IES that allows teachers to contribute their content will have a much better chance to compete with the dominant non-intelligent systems.

1.2 Supporting teachers as authors of 'intelligent content'

The goal of this paper is to investigate further the issue of teachers developing educational content for IESs. Claiming that it is not a trick used by a few researchers (Arroyo *et al.*, 2001; Ritter *et al.*, 1998a), but an essential component of the new design paradigm, we want to explore this approach systematically, draw connections to similar works, and finally move beyond the existing narrow area of authoring algebra word problems.

The teacher's involvement in the process of IES authoring is both a recognised need and a recognised research stream in the IES community. However, the original goal was to involve the teachers in the core design process. This direction of work brought little success. After a number of attempts to turn teachers into key developers of IESs (Lewis *et al.*, 1987; Murray, 1999; Murray and Woolf, 1992), no one kept the illusion that a regular teacher could design an IES, even with the help of advanced authoring tools. As pointed out by Murray in his comprehensive overview of ITS authoring tools: "The average teacher should not be expected to design ITSs any more than the average teacher should be expected to author a textbook in their field" (Murray, 1999). The current design paradigm limits teacher participation in IES authoring to the model of 'best teachers working as members of a design team'.

The new design paradigm offers teachers a different place in the process of IES authoring. It leaves the core authoring in the hands of well-prepared design teams and gives teachers a chance to extend the system and fine tune it to their local needs by contributing educational content. This division of labour is quite natural. In a similar way, instructors are rarely able to create a textbook for their courses, but many augment existing textbooks with their own examples, problems, questions, and even additional explanations of complicated concepts.

There is evidence that, with proper support, teachers can contribute intelligent content to an IES or fine tune an existing IES to the needs of their courses. The possibility of employing teachers as authors of intelligent content has been explored since late 1980 in a narrow stream of work on the crossroads of CAI and ITS. Such systems as ECAL (Elsom-Cook and O'Malley, 1990) and REDEEM (Ainsworth and Grimshaw, 2004) attempted to provide an authoring system that would allow teachers to create 'more intelligent' courses by adding content with a relatively small amount of knowledge of the IES system, behind pages in a domain-independent runtime system. By reducing the demand placed upon teachers, the developers of REDEEM were able to demonstrate not only that teachers can use the authoring tool, but also that they can actually develop effective courses. The ability to fine tune an IES that has been created by a professional team to the needs of a specific class is a kind of 'minimalist' teacher involvement that has also been explored in several other projects. In particular, several curriculum sequencing and adaptive hypermedia systems give the teacher the option to specify an educational goal for a class or an individual student as a set, sequence, or even a tree of concepts (Brusilovsky, 1992a; Brusilovsky *et al.*, 1998). A number of other projects declared that an IES should allow teachers to contribute their favourite examples and problems.

Thus, the idea to involve teachers as developers of educational content in IES has some clear roots in the field. The challenge, however, is to move this interest from the level of random ideas to the level of practical technology. To make this leap, the ideas must be supported by relevant architectures and tools. First, the new design paradigm requires an open architecture for IES – one that allows the inclusion of externally created educational content (*i.e.*, created by authors outside of the original development team). Secondly, it requires special authoring tools to help the teachers create this content.

We think that the development of IES content authoring tools for use by regular teachers is a challenge that should not be underestimated. While the problem of developing external content is less complicated than the one of developing core IES functionality, we must remember that teachers are much less prepared to handle the authoring process than professional IES developers are, and thus they require a significant level of support. The pioneering paper on this issue (Ritter *et al.*, 1998a) provides a good analysis of the problems involved, as well as a set of design principles that have been developed for authoring problems, specifically for a cognitive rule-based tutoring system.

The problem here is that the content to be created for an IES is really 'intelligent content'. To guide students to the most relevant content items as well as to provide assistance during work with the items they have selected (*e.g.*, presentations, examples and problems), an IES requires *knowledge* behind every content item and sometimes even for small sections of an item. Even the simplest presentation-style fragments of content should be connected to the proper elements of domain knowledge (concepts) so that an IES can understand what the fragment is about, when it is reasonable to present it, or when it is premature. More complicated kinds of content, such as examples and problems, may require the representation of additional knowledge, in order to enable an IES to run the example or to give extra support to a student who is trying to solve a problem.

The complexity of authoring intelligent content for a specific domain is determined by the ability of the authoring system to understand the content in the form in which it is presented to the student. There are some domains where problems and examples are originally written in a formal structured language that can be directly understood by the system: derivation in calculus (Brusilovsky, 1993), expression evaluation in C (Brusilovsky and Su, 2002), and equation solving in algebra (Ritter and Anderson, 1995). In these domains, the knowledge behind a fragment of educational content can be deduced automatically by the system from the problem statement. As a result, in these fortunate domains the authoring of additional problems or examples is easy – the author only needs to provide the problem statement in a traditional form. In less fortunate domains, the IES finds it difficult or impossible to understand the content items without help. In this case, the knowledge behind a content fragment cannot be automatically deduced and must be provided within the authoring process. The amount of knowledge to be provided by the author can range from merely specifying a few concepts behind a presentation fragment, as in adaptive hypermedia systems like KBS-Hyperbook (Henze and Nejdl, 2001) or SIGUE (Carmona *et al.*, 2002) to providing complicated descriptions of a programming problem in terms of plans and rules, formatted in a special way, as in ELM-PE (Weber and Bögelsack, 1995).

The necessity of providing the required knowledge for every fragment of content puts up a barrier for regular teachers who would otherwise want to author this content by themselves. While the core system developers are certainly skilled enough to provide the necessary knowledge 'behind content', the teachers may not be able to do this even after special training sessions. The challenge for the AI in the education community is to provide intelligent authoring support tools for teachers who are interested in contributing 'intelligent content'. From the viewpoint of the ITS authoring area, work on tools for teachers should extend beyond the traditional variety of ITS authoring tools (systems, libraries, toolkits) to include a *new layer* of teacher-oriented tools (Murray, 2003). These tools should help teachers enter the content by themselves and should provide all necessary knowledge for this content to be used by an adaptive or intelligent educational system. Tools that have been developed by a few pioneers in order to support teachers in the authoring of algebra problems (Arroyo *et al.*, 2001; Ritter *et al.*, 1998a) provide a good example of authoring support. However, more research in different domains must be performed in order to fully understand the problems and needs of intelligent authoring support. The goal of this paper is to demonstrate some of the problems and issues of intelligent authoring support in a more challenging domain: language learning. In the context of the Vocabulary Acquisition System ELDIT (Gamper and Knapp, 2003) we have explored several intelligent and non-intelligent mechanisms, which will support authors of educational content. We have implemented two authoring support components to match the two kinds of content that are essential for language learning: illustrative examples and educational texts. The following chapters present the ELDIT system, the needs and challenges of content authoring by teachers, and the two authoring support components that we have developed.

2 The ELDIT vocabulary acquisition system

2.1 ELDIT's approach to vocabulary acquisition

Vocabulary acquisition is an important part of foreign language learning. In 1950, lexicographers started to develop so-called 'learners' dictionaries' in which vocabulary coverage was limited, word definitions were simpler and often supported by pictures and carefully selected lexicographic patterns and examples demonstrated typical use of a word, *etc.*

ELDIT (*E*lektronisches *L*ernerwörterbuch *D*eutsch *IT*alienisch) (Abel and Weber, 2000) is a combination of an electronic learners' dictionary for German and Italian and an intelligent vocabulary acquisition system.[2] The primary users are language learners who already have basic skills in the target language, Italian or German. This is, for example, the case for all people in the bilingual Province of South Tyrol in the North of Italy who prepare for an examination on bilingualism – an exam that is required for employment in the public sector within this province. ELDIT is a semi-bilingual system especially designed to support learners who already know one of the two languages. Word meanings are defined in the target language and are translated to an equivalent word in the source language.

The core part of ELDIT is an extensive learners' dictionary. This glossary has been developed as learning material in order to prepare for the examination on bilingualism and contains approximately 3500 words for each language. For both the Italian and the German languages, a cross-matching of the different basic vocabularies has been built. Accomplishing this, approximately 95% of the words in a normal text should be covered.

Following the tradition of learners' dictionaries, the ELDIT dictionary stores a large set of information for each word entry, which is carefully collected, analysed, and selected by a group of linguists. This includes the most important and representative collocations (*i.e.*, frequent and usual word combination) and usage patterns. All these pieces of information are highly interlinked. Figure 1 shows a screenshot of the dictionary entry for the German verb *bauen* (to build). In the left-hand frame the lemma (*bauen*) together with morphological information and different word meanings are shown. Each word meaning is described by a definition, a translation, and an example sentence. In the right-hand frame, additional information regarding the correct usage of the word is shown. A tab metaphor is used to illustrate typical usage patterns, word relations, *etc*. For example, the tab selected in Figure 1 shows verb valency (information about the number and position of required subjects and objects to be used with this verb). The items are described by a grammatical explanation, a pattern in table form, and some lexicographic examples. More details can be found in (Gamper and Knapp, 2003).

Despite its huge amount of information, the basic dictionary remains principally a reference tool, specifically designed for language learners. The learner consults the dictionary if he/she encounters an unknown word or wants to know more about a word's usage. ELDIT extends the learner's dictionary by adding an adaptive and intelligent system for systematic vocabulary acquisition in foreign language learning. ELDIT is motivated by the belief that a combination of several methods is necessary for vocabulary acquisition (Gamper and Knapp, 2002): Words should be systematically learned on a word list, interactively practised, and read in authentic texts. This learning process consists of the following four steps:

1 Perception

  Being able to identify a word, to distinguish different word meanings and to differentiate words from related words, is necessary for the learner to understand spoken and written language correctly. In ELDIT, the learner can explore the various descriptions of a word meaning by checking definitions, translations, pictures and example sentences, and by examining related words in the same way.

2 Usage

  The next step is to learn how to use a word. Grammatical and collocational knowledge is necessary for this task. The learner should study typical patterns of word usage, listed as collocations and idiomatic expressions, as well as the conjugation and declension of a word.

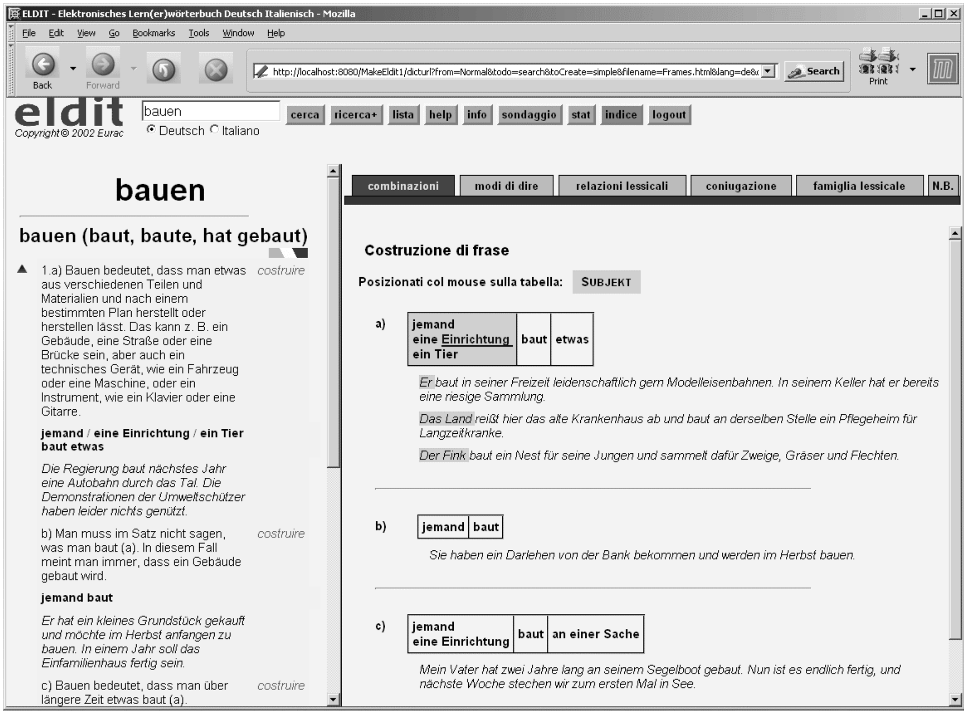
3 Characteristics

Finally, the learner should enlarge and complete his/her knowledge about a word. For that purpose, word characteristics, exceptions to a rule, false friends,[3] *etc*. listed in the footnotes of a word, should be studied.

4 Context

After the words have been learned in all their facets, the user should explore them in an authentic text. The system will select a text from the text corpus included in the system, which contains the words studied previously and, preferably, only a few new words at most.

Figure 1 Dictionary entry of the German word *bauen* (to build) with the *combinazioni* (verb valency) tab selected


ELDIT - Elektronisches Lern(er)wörterbuch Deutsch Italienisch - Mozilla
File Edit View Go Bookmarks Tools Window Help
Back
Forward
http://localhost:8080/MakeEldit1/dicturl?from=Normal&todo=search&toCreate=simple&filename=Frames.html&lang=de&
Search
Print
eldit
Copyright © 2002 Eurac
cerca
ricerca+
lista
help
info
sondaggio
stat
indice
logout
combinazioni
modi di dire
relazioni lessicali
coniugazione
famiglia lessicale
N.B.

bauen

● Deutsch ○ Italiano

bauen

bauen (baut, baute, hat gebaut)

1.a) Bauen bedeutet, dass man etwas aus verschiedenen Teilen und Materialien und nach einem bestimmten Plan herstellt oder herstellen lässt. Das kann z. B. ein Gebäude, eine Straße oder eine Brücke sein, aber auch ein technisches Gerät, wie ein Fahrzeug oder eine Maschine, oder ein Instrument, wie ein Klavier oder eine Gitarre. *costruire*

jemand / **eine Einrichtung** / **ein Tier baut etwas**

Die Regierung baut nächstes Jahr eine Autobahn durch das Tal. Die Demonstrationen der Umweltschützer haben leider nichts genützt.

b) Man muss im Satz nicht sagen, was man baut (a). In diesem Fall meint man immer, dass ein Gebäude gebaut wird. *costruire*

jemand baut

Er hat ein kleines Grundstück gekauft und möchte im Herbst anfangen zu bauen. In einem Jahr soll das Einfamilienhaus fertig sein.

c) Bauen bedeutet, dass man über längere Zeit etwas baut (a). *costruire*

Costruzione di frase

Posizionati col mouse sulla tabella: SUBJEKT

a)

jemand **eine Einrichtung** **ein Tier**	**baut**	**etwas**

Er baut in seiner Freizeit leidenschaftlich gern Modelleisenbahnen. In seinem Keller hat er bereits eine riesige Sammlung.

Das Land reißt hier das alte Krankenhaus ab und baut an derselben Stelle ein Pflegeheim für Langzeitkranke.

Der Fink baut ein Nest für seine Jungen und sammelt dafür Zweige, Gräser und Flechten.

b)

jemand	**baut**

Sie haben ein Darlehen von der Bank bekommen und werden im Herbst bauen.

c)

jemand **eine Einrichtung**	**baut**	**an einer Sache**

Mein Vater hat zwei Jahre lang an seinem Segelboot gebaut. Nun ist es endlich fertig, und nächste Woche stechen wir zum ersten Mal in See.

All necessary information to support the first three steps of this process is provided directly in the dictionary. The fourth step is supported by authentic texts with comprehension questions that ELDIT maintains, in addition to the dictionary. In the text, all the words are linked to the corresponding dictionary entry, so that unknown words can easily be inspected by the learner. The text can either be read as a whole or practised as a gap-filling exercise. Working with text corresponds to a problem-solving activity in an ITS.

The support for this learning process is personalised to make it as efficient as possible and to meet individual students' interests and needs. The adaptive vocabulary acquisition component in ELDIT adopts a specific form of problem-based learning. Depending upon user preferences, a smaller or larger group of words from a user-specific interest domain will first be offered for study according to Steps 1 to 3. Then the system adaptively selects a text for the student to study. This text should first of all be suitable to repeat the new words recently learned. Moreover, each text will probably contain some new words for the user to check. These new words are summarised by the component and are studied systematically according to Steps 1 to 3. After the user has studied these new items, Step 4 is applied again and a new unknown and interesting text, in which these words occur, is proposed to be read, and so on. We call this method of learning 'adaptive, contextualised vocabulary acquisition'. An early version of the system-driven adaptive vocabulary acquisition is reported in Gamper and Knapp (2002).

2.2 Two levels of content in ELDIT

The power of ELDIT is based on a very large amount of carefully authored pedagogical content. In the context of this paper, we distinguish between two types of content: core content and educational content.

By *core content* we mean the part that is essential for the functionality of the system. It forms the skeleton of the learning system. Removal of a core content fragment may break some functionality of the system. Core content in ELDIT includes the words (vocabulary), word definitions, collocations, idiomatic expressions, derivations, compounds, synonyms and translation equivalents. These pieces of information define the rules of a language, *i.e.*, what is the meaning of a word and how to combine words in order to form correct sentences.

While the core content forms the skeleton of the language learning system, it does not give the student enough to really learn a language. The core content has to be enriched by so-called *educational content*, which is comprised of lexicographic examples, illustrations, texts, pictures and sound, and exercises. All these bits of information are extremely important for language learning in that they show the meaning of words and language in common usage. In contrast to core content, educational content can be removed from the system without affecting its core functionality.

The development of the core content and a minimal set of educational content is a very difficult and time-consuming process, which has to be done by experts. In the ELDIT project we developed the core content for approximately 7000 word entries (3.500 German and 3.500 Italian) plus a minimum of educational content: one example sentence for definitions, collocations, idiomatic expression, verb valency, synonyms, *etc.*, and about 400 educational texts for each language. However, the content developed by the experts was just barely sufficient for the system to operate. In particular, we immediately realised that one example sentence for each piece of core content is not enough. We observed this need for additional examples also during an exhibition, where we presented the ELDIT language learning system. Many users explicitly expressed the need for more illustrative examples.

In our efforts to extend the volume of available educational content, we focused on the most needed illustrative examples and texts. A larger variety of examples and texts make any language learning system richer. For an IES such as our adaptive, contextualised vocabulary acquisition system, an abundance of educational content is essential and important, in order for the system to select the most appropriate example or text for a given student. At that time, we implemented mechanisms for extending the volume of both kinds of educational content: examples and texts. These mechanisms explored two different authoring approaches: automatic extraction of new educational content and assisted authoring of content.

In developing authoring components for educational texts, we explored a traditional authoring approach: an authoring system that assists teachers in adding new educational texts to the system. We felt, however, that this approach was not well suited to the authoring of new examples, because of the enormous number of word usage cases. As a result, we choose to explore the option of extracting examples from the already existing volume of educational content without any direct assistance from a teacher. We expected that the main source of example sentences would be educational texts that were already created by the teachers and added to the system. With this approach, even a fully automatic example-extraction mechanism can be fine tuned to the interests of a specific group of learners or a specific teacher. A system can be easily instructed to provide examples extracted from a specific subset of texts; for example, texts on a specific subject or texts authored by a specific teacher. Since at the moment of development a large volume of teacher-authored texts was not available, we explored the automatic extraction mechanism in a slightly different, but also very important context: extracting new examples from the body of already available educational examples. As we noted above, the expert linguists of the ELDIT's development team provided one example per case. Each example, however, was a full sentence that typically could serve as an example for two or more different cases. Coupled with the body of already existing examples, our automatic extraction mechanism produced a system for the re-use of lexicographic examples. Thus, we were able to explore both approaches in the context of language learning in order to increase the amount of educational content, as mentioned in the introduction: re-use of already developed educational content and the support of teachers as authors of new educational content. The following sections present the implementation of the example re-use and text authoring components.

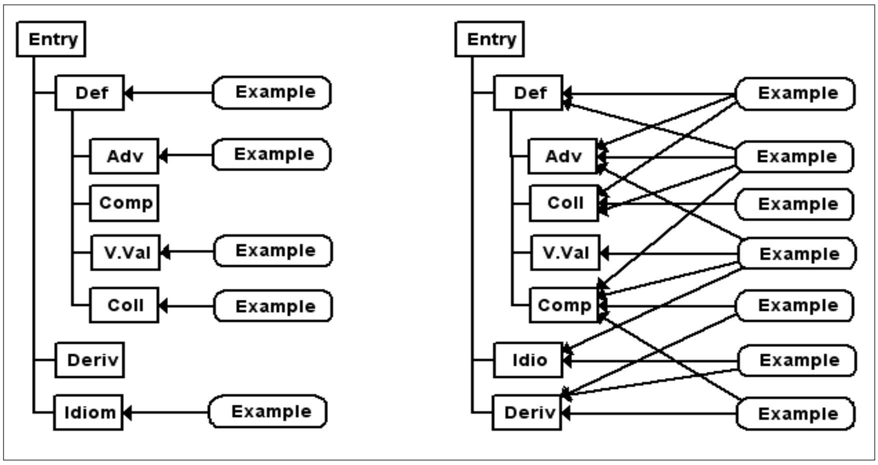
3 Re-using illustrative examples in ELDIT

3.1 The need for multiple examples

ELDIT contains a large number of text sentences and lexicographic examples, which are used to illustrate language in use, *i.e.*, how language rules can be applied to form correct and meaningful sentences. We call these example sentences *illustrative examples*. Authoring of these examples takes a large amount of effort. The linguists of the ELDIT development team were able to create just one illustrative example sentence for each of the following pieces of information in the dictionary: definitions, collocations, verb valency (see Figure 1), and idiomatic expressions. Altogether they have developed more than 60 000 example sentences, which are already included in ELDIT. Despite this impressive number, there is still only one example per case (such as shown in the verb valency in Figure 1). There are several reasons to wish for more examples per case. Even in a non-adaptive system, learners who have a specific context or sentence in mind might want to see an example sentence that is closer to their specific needs. In an adaptive system, which attempts to show the most relevant examples to the student (*i.e.*, examples with the smallest number of unknown words) a variety of examples is absolutely necessary.

Following our paradigm of authoring educational content by teachers, we attempted to develop a mechanism for the automatic extraction of illustrative examples from educational texts authored by teachers. To explore the feasibility of automated example authoring, we started with a slightly simpler target: example re-use. While in the original ELDIT glossary each example sentence was connected to one case only (Figure 2, left), most examples were relatively long full sentences and could serve as illustrative examples for several different words and cases. We developed a module that allows multiple re-use of the existing illustrative examples in ELDIT. Now for each piece of information, we have not only the main example added by the linguists, but also additional examples that are relevant to new cases, even though they were created for other contexts (Figure 2, right). The example extraction module is currently being extensively evaluated. The evaluation (and possible fine tuning) is necessary for any fully automatic tool that extends the body of educational content without any human involvement. Once fully evaluated, this mechanism will be used for example extraction from teacher-authored texts. Some early evaluation of automated example re-use was reported in (Knapp *et al.*, 2004).

Figure 2 Traditional and new methods of providing information and example sentences in ELDIT



3.2 *The mechanism for extracting illustrative examples from existing content*

The retrieval of additional example sentences for a specific case is not trivial. One might think that a simple search for the case pattern in a database of examples would return all additional examples that can now be connected to the case. However, there are at least the following problems that make it impossible to accept these examples at face value:

- *The patterns under consideration might be unstructured*

  For instance, they may contain meta information such as slashes to indicate variations. For instance, the pattern *gli occhi, la bocca, il viso, ... belli/bella/bello* indicates several patterns: *gli occhi belli* (beautiful eyes), *la bocca bella* (a beautiful mouth), and *il viso bello* (a beautiful face). All these patterns should be considered when searching for additional examples.

- *The words occur in declined or conjugated form*, both in the patterns and example sentences

  For example, the collocation *to go home* occurs in another form in the sentence *Yesterday I went home very late*. Therefore, this sentence should also be matched, but would have been missed if only the initial verb form was used.

- *It is not sufficient that all words of a pattern occur in a lexicographic example; they must also occur as a collocation*

  For instance, the word combination *to go home* occurs as a collocation in the sentence *I went home very late*, but not in the sentence *I went out and came home very late*.

- *Words usually have several meanings*

  For instance, the word *house* may be a building but also a dynasty. Hence, the sentence *The royal house of Norway is a branch of the princely family of Glücksburg* is not a good illustration of the definition *A house is a place to live and to work*.

To tackle these problems we built an intelligent text processing mechanism that applies some techniques of Natural Language Processing (NLP) and Information Retrieval (IR). The mechanism includes four steps:

1. Extract ‘clean’ patterns
2. Retrieve all example sentences
3. Recognise collocations
4. Disambiguate meaning.

We describe these steps in more detail and show some examples below.

3.2.1 Extract ‘clean’ patterns

The first step is to construct new, ‘clean’ patterns, which can be passed to our search engine. Three situations have to be distinguished:

- Case A: derivations, compound words and adverbs
- Case B: collocations and idiomatic expressions
- Case C: definitions and verb valency.

Case A is about words that are generated for a dictionary entry by word formation rules (*adverbs, derivations and compound words*). These words are listed within the entry, and the word formation rules are highlighted in order to communicate them to the learner. In this case, it is easy to get a ‘clean’ pattern, since the given pattern consists only of the generated word as it occurs in the citation form and different meanings are not considered. Hence, the word can be passed directly to the search engine.

In Case B, we have *multiple word expressions* that are given in an unstructured form. The first step is to unfold all meta-symbols. For example, unfolding the pattern *ein Auto fährt schnell/langsam* (a car runs fast/slowly) yields the patterns *ein Auto fährt schnell* and *ein Auto fährt langsam*. Then some very general words (such as articles) are removed and the remaining words are connected with an AND function and marked as obligatory. The result for our example is the following set of search patterns: *+Auto AND +fährt AND +schnell* as well as *+Auto AND +fährt AND +langsam*.

In Case C, different word meanings have to be considered that are described by a *definition*, such as *Eine Glocke ist ein Gegenstand aus Metall, der irgendwo hängt* (a bell is a metal device that hangs somewhere) *or verb valency patterns* such as *jemand baut etwas* (somebody builds something). In both cases, quite general words are frequently used, such as *Gegenstand* (thing), *irgendwo* (somewhere), *jemand* (somebody), or *etwas* (something). Moreover, in both cases, a specific word meaning is given, which should be matched in the retrieved example. The general words are definitely not suitable for building a good search expression and thus make it difficult to find appropriate examples for that specific word meaning.

The solution we applied was to build our search expression based upon some specific words within the main lexicographic example provided by the experts. Since they chose these specific words, we assumed that they were appropriate usage for the meaning of the word in consideration: We first extract the clause, which contains the word under consideration. From this clause we extract all verbs as well as the first occurrence of nouns on the left-hand and right-hand side of that word. The resulting set of words is then used as a search pattern, in such cases.

Let us have a look at an example sentence provided with the definition of *house* as the group of people in a theatre: *Als die Vorstellung zu Ende war und der Vorhang zu ging, klatschte das ganze Haus begeistert Beifall.* (*When the performance was finished, the whole house applauded enthusiastically.*)

The main clause is *klatschte das ganze Haus begeistert Beifall*. The verb in this clause is *klatscht* (to applaud), there is no noun on the left-hand side of *Haus*, but there is a noun on the right-hand side of *Haus*, namely *Beifall* (applause). From this phrase, we generate the search patterns '+Haus AND +klatscht' and '+Haus AND +Beifall'.

3.2.2 Retrieve examples

The second step is to retrieve all possible examples from the ELDIT dictionary by passing the search pattern obtained in Step 1 to the search engine. Our search engine applies NLP techniques both on the search pattern and on the learning material in order to retrieve results that might also occur in inflected form. The result of a query is a list of corresponding citation forms followed by the corresponding category (*i.e.*, word class):

query → ging

result → gehen (Cat V).

We have first transformed the ELDIT data to its lemmatised form. Then we have generated two indices. One index contains the data in the original form, the other one contains all data in the lemmatised form (*i.e.* the citation form). The two indices include cross-references to each other. The search engine first lemmatises the obtained search pattern, for instance *Nest AND baut* becomes *Nest AND bauen*, then the lemmatised pattern is searched in the lemmatised index, where the example sentence *Der Vogel bauen mein Nest aus Ast, Blatt und Erde* is found. Last, by using the cross-reference system, the corresponding original example sentence is retrieved, namely *Die Vögel bauten ihre Nester aus kleinen Ästen, Blättern und Erde*, and given back to the module.

3.2.3 Recognise collocations

The third step is to recognise collocations. A sophisticated concordance tool is required to check whether a word combination forms a collocation in a sentence or not. Currently, we are using a rather simple approach based on a set of rules used to identify real collocations. The following are two typical rules:

1. The words of a pattern have to occur within one main sentence or within a subordinate clause.
2. There should not be more than a determined number of other words between the words of the pattern.

We are currently working with three words in the case of noun-verb combinations and zero words in the case of noun-adjective and verb-adjective combinations. The application of these rules eliminates invalid sentences for a given pattern. For example, let us consider the pattern *ein großes Haus* (a big house). Among others, the search engine retrieves *Meine Eltern zogen in ein Haus mit einem großen Garten* (my parents moved into a house with a big garden). In this sentence the garden, not the house, is big. Applying the zero-words rule identifies this sentence as invalid for our specific pattern.

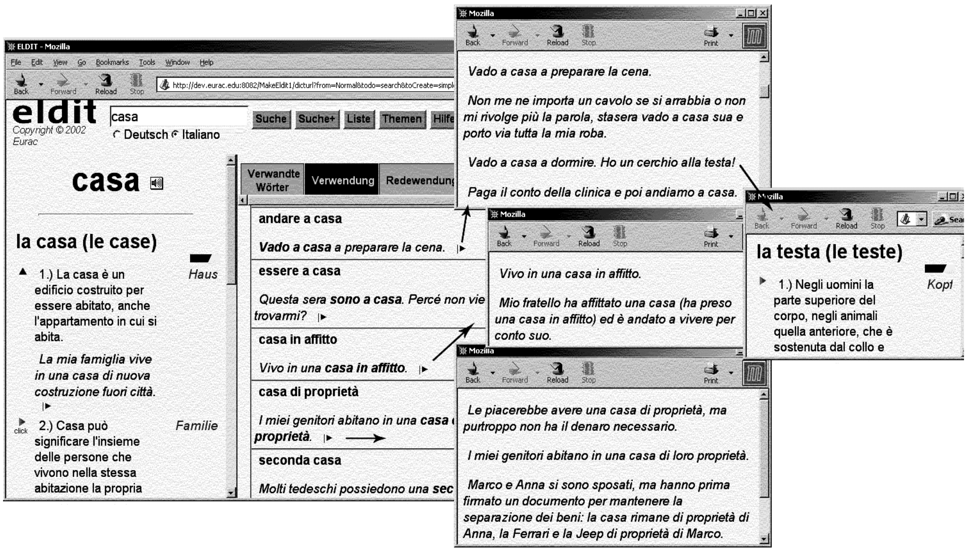
3.2.4 Meaning disambiguation

The last and most difficult step is the disambiguation of word meanings. Currently this step uses search patterns derived from nouns and verbs in the original example sentences (see Case C) in Section 3.2.1, above). This is a first step in the creation of a programme that is able to perform meaning disambiguation. Further steps are possible: Meaning disambiguation programmes include context vectors, which are lists of words that are semantically related to the specific meaning of a word. Such context vectors can be generated, *e.g.*, by listing the nouns, verbs, and adjectives of the *collocations* collected for a specific word meaning. In ELDIT the general context vector of the word *house* as *a place to live and work* would be {*bauen, renovieren, wohnen, kaufen, mieten, vermieten…*}, and the general context vector of the word *house* as *a group of people in a theatre* would be {*Theater, Vorstellung, Beifall, toben, klatschen, Begeisterung, …*}. On the other hand, the context vector of the example sentence *Als die Vorstellung zu Ende war und der Vorhang zu ging, klatschte das ganze Haus begeistert Beifall.* (*When the performance was finished, the whole house applauded enthusiastically.*) Could be the list of *words used within* this example sentence, namely {*Vorstellung, Ende, Vorhang, zugehen, klatschen, Haus, begeistert, Beifall*}. The context vectors of the obtained example sentences could be compared with the general context vector of the word meaning in consideration, and in this way a better indication would be obtained as to whether an example sentence really matches that meaning or not.

3.3 The use of multiple examples

Figure 3 displays the new version of the ELDIT interface, which has multiple examples per case. The interface distinguishes the *primary* example, provided by expert linguists, from *additional* examples automatically extracted from educational content. For the Italian word *casa* shown in Figure 3 the first collocation is *andare a casa* (to go home) with the primary example *Vado a casa a preparare la cena* (I will go home to prepare dinner). In the additional examples, the expressions 'I will go home to sleep…' and '…let us go home' can be found. For the word combination *casa in affitto* (rented flat) not only the primary example *Vivo in una casa in affitto* (I am living in a flat), but also the additional example *Mio fratello ha preso una casa in affitto ed è andato a vivere per conto suo* (My brother has rented a flat and is now living there on his own) can be found. In the first list of additional examples, the word *testa* (head) was unknown to the user. Since all words in the example sentences are linked to the dictionary entry, a click on the unknown word shows its definition and translation. Once our example-selection mechanism can be used to extract relevant examples from teacher-provided texts, a range of sentences from a teacher's favourite texts will also be available as illustrative examples, naturally tuning the system to the needs of a specific class.

Figure 3 Access to primary and additional (extracted) illustrative examples in ELDIT



ELDIT - Mozilla
File Edit View Go Bookmarks Tools Window Help
Back Forward Reload Stop
http://dev.eurac.edu:8082/MakeEldit1/dicturl?from=Normal&todo=search&toCreate=simpl
eldit
Copyright © 2002 Eurac
casa
Suche Suche+ Liste Themen Hilfe
○ Deutsch ● Italiano
Verwandte Wörter
Verwendung
Redewendung
Haus
Familie
click
Mozilla
Print
Kopf

casa

la casa (le case)

1.) La casa è un edificio costruito per essere abitato, anche l'appartamento in cui si abita.

La mia famiglia vive in una casa di nuova costruzione fuori città.

2.) Casa può significare l'insieme delle persone che vivono nella stessa abitazione la propria

andare a casa

Vado a casa a preparare la cena.

essere a casa

Questa sera ***sono a casa****. Percé non vie trovarmi?*

casa in affitto

Vivo in una ***casa in affitto****.*

casa di proprietà

I miei genitori abitano in una ***casa*** *(*
proprietà*.*

seconda casa

Molti tedeschi possiedono una ***sec***

Vado a casa a preparare la cena.

Non me ne importa un cavolo se si arrabbia o non mi rivolge più la parola, stasera vado a casa sua e porto via tutta la mia roba.

Vado a casa a dormire. Ho un cerchio alla testa!

Paga il conto della clinica e poi andiamo a casa.

Vivo in una casa in affitto.

Mio fratello ha affittato una casa (ha preso una casa in affitto) ed è andato a vivere per conto suo.

la testa (le teste)

1.) Negli uomini la parte superiore del corpo, negli animali quella anteriore, che è sostenuta dal collo e

Le piacerebbe avere una casa di proprietà, ma purtroppo non ha il denaro necessario.

I miei genitori abitano in una casa di loro proprietà.

Marco e Anna si sono sposati, ma hanno prima firmato un documento per mantenere la separazione dei beni: la casa rimane di proprietà di Anna, la Ferrari e la Jeep di proprietà di Marco.

4 Authoring of educational texts in ELDIT

4.1 The challenge of authoring educational texts

Educational texts are an important component of the vocabulary acquisition approach, as implemented in ELDIT (Gamper and Knapp, 2002). As we mentioned above, the system alternates between word study directly related to the ELDIT glossary and work with text. Each text unit includes a text to study and several questions, which the learner must answer by producing complete sentences in the target language. The text units allow the learner to apply new words and to produce target language output, which is very important for retaining the acquired vocabulary. As we stressed above, texts in ELDIT play the same role as educational problems in algebra tutors.

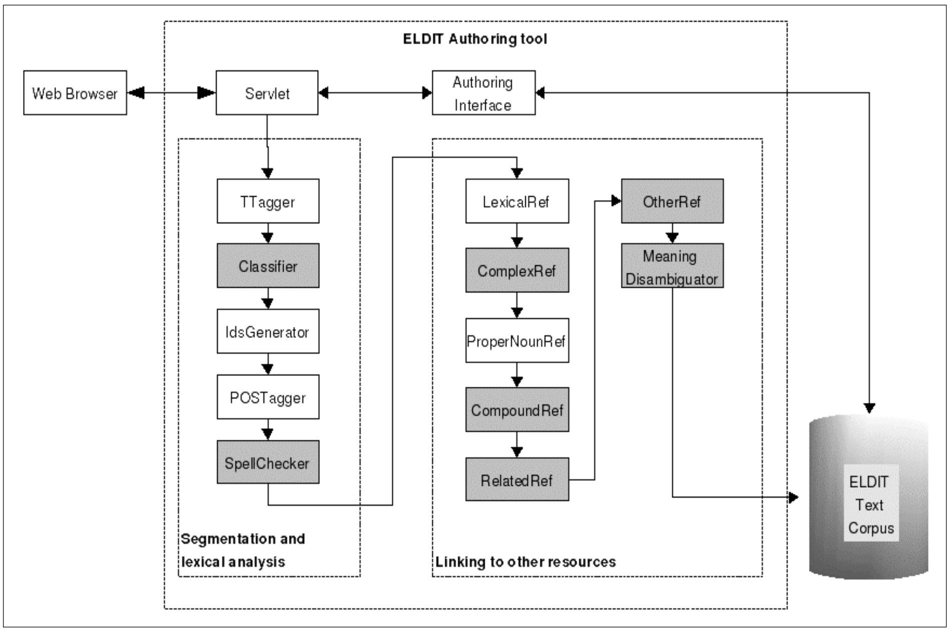
The first version of the text corpus in ELDIT included about 400 texts for each language. They were provided by the core development team. While selecting or creating the text is not a complicated task, integrating a new text into the ELDIT structure requires that each word used in the text is annotated with lemma and word class and is linked to the corresponding dictionary entry. This metadata and links are important for both the students and ELDIT's adaptation mechanisms. Students need links to dictionary entries to check the meaning of new or forgotten words. The adaptation mechanisms need to know about every word used in the text in order to select the most appropriate text to study (*i.e.*, a text that includes a good number of target words and some optimal mixture of already learned and unknown words). As a result, the integration of each text into ELDIT is very time consuming and requires both sophisticated computational linguistics tools and a professional linguist. Because of these reasons, the ELDIT's core development team was only able to develop a minimal number of educational texts.

As we argued above, the opportunity for teachers to provide educational texts is beneficial for at least two reasons. First, students can be offered a greater variety of texts to study – which is necessary, both for students (who are interested in choosing texts that are closer to their interests) and for the adaptation mechanism. Second, by contributing their favourite texts, teachers can adapt the system to their needs and their established methods of teaching. To support our 'teacher as authors' paradigm, we have implemented an automated tool to enable teachers to contribute new educational texts.

4.2 *The mechanisms to support assisted authoring of educational texts*

In designing a tool to support content authors, our intention was to use computational linguistics techniques to automate both the routine parts of text authoring (such as word annotation) as well as the knowledge-demanding portions of it (such as word sense identification). We wanted to leave only the pedagogical side of text authoring in the author's hand, by presenting the text and associated questions in a simple form-based interface that could easily be used by any language teacher. Figure 4 shows the overall architecture of the authoring tool.

Figure 4 Architecture of the ELDIT text authoring tool

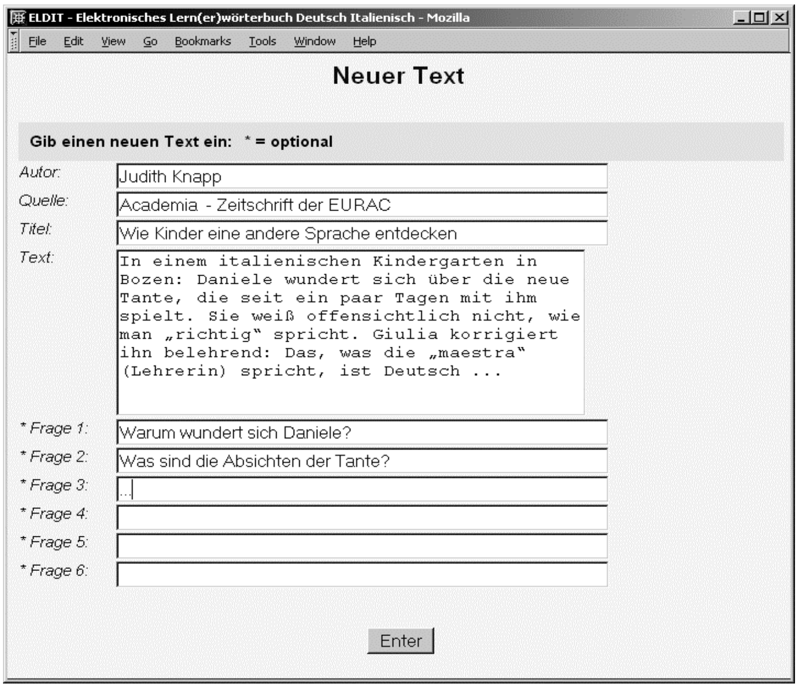


To insert a new text into the ELDIT vocabulary acquisition system, the author starts with a simple form-based interface (Figure 5), which requests the following information: the text, the name of the author, the reference, the title, and up to six associated questions. Once filled, the HTML form is forwarded to a dedicated servlet, which then passes it to a chain of text processing tools. The overall text processing is composed of a sequence of steps, in which the author-provided data is transformed into the final format, to be used by ELDIT. We can distinguish between two groups of processing operations:

1 *Segmentation and lexical analysis* – texts are annotated with basic linguistic information.

2 *Linking to other resources* – texts are integrated into ELDIT and linked to already available resources, most often the learner's dictionary.

Figure 5 Form-based interface for authoring educational texts



ELDIT - Elektronisches Lern(er)wörterbuch Deutsch Italienisch - Mozilla
File Edit View Go Bookmarks Tools Window Help
Enter

Neuer Text

Gib einen neuen Text ein: * = optional

Autor: Judith Knapp

Quelle: Academia - Zeitschrift der EURAC

Titel: Wie Kinder eine andere Sprache entdecken

Text: In einem italienischen Kindergarten in Bozen: Daniele wundert sich über die neue Tante, die seit ein paar Tagen mit ihm spielt. Sie weiß offensichtlich nicht, wie man „richtig" spricht. Giulia korrigiert ihn belehrend: Das, was die „maestra" (Lehrerin) spricht, ist Deutsch ...

* Frage 1: Warum wundert sich Daniele?

* Frage 2: Was sind die Absichten der Tante?

* Frage 3: ...

* Frage 4:

* Frage 5:

* Frage 6:

At the time of this writing, the core processing operations have been fully implemented and tested. The authoring tool works and has already been used for authoring new texts. The focus of our current work is the implementation and testing of a few advanced steps, which will increase the educational power of the educational texts and the precision of the adaptation algorithm. These steps are marked grey in Figure 4, and are still under development.

In the following section, we will briefly describe the various text processing steps. While we omit some low-level detail, which would be of interest only to linguists, we provide sufficient information to allow experts in other domains to appreciate the complexity of the process. Our goal is to demonstrate one good example of how to create sophisticated, intelligent authoring support, required to allow the authoring of content by teachers within the richer domains. While assisted authoring in other domains will certainly require different steps (Arroyo *et al*., 2001; Ritter *et al*., 1998a), the overall complexity of the process and the need to involve several intelligent and non-intelligent processing techniques are probably quite typical for a range of domains.

4.2.1 Segmentation and lexical analysis

The first group of operations includes rather standard lexical analyses and generates some additional metadata, which is required specifically for ELDIT.

The *TTagger* is the first module that is invoked by the TextAuthor. This is a kind of segmentation tool that encodes the data in XML, where the different parts of a text including the title, the body, question elements, *etc*. are explicitly encoded. Apart from these macro-structure divisions, which are already separated in the input form, the segmentation tool splits all text elements into smaller units, namely paragraphs, sentences and words.

Then the *Classifier* is invoked, adding some metadata to the texts, including the language, the difficulty level and the domain. This information is obtained dynamically by applying a text classification algorithm. Such a classifier has been implemented and tested in two projects at the European Academy Bozen/Bolzano: BISTRO[4] and MIRIS,[5] and is now being adapted for use in ELDIT.

Next, the *IdsGenerator* generates a unique ID for each XML element. The ID consists of the path along which the element can be found in the XML tree and a number, which indicates the placement of the element in a list of equal elements.

The *Part-of-Speech (POS) Tagger* adds lemma and word class information to each word of the text. This is done in several sub-steps. Lemma and word class are obtained from Word Manager, which is a lexical analysis tool that has recently been added to ELDIT through a collaboration with the 'Scuola Universitaria Professionale della Svizzera Italiana – SUPSI'. Since the Word Manager is operating on a word level, not all ambiguities can be resolved. For example, for the Italian word *posto,* two lemmas and word classes are assigned, namely 'il posto/noun' and 'porre/verb'. To improve this situation, we implemented a simple, rule-based *POS Disambiguator* that eliminates many ambiguities. An example of a disambiguation rule is that an article must be followed by a noun or adjective.

The last step in this first group of text processing operations is to invoke the *SpellChecker*, which provides feedback about common spelling errors in the text. Again, the Word Manager will be used since it contains information about old German spelling rules that were valid until 1998. Also, word forms that cannot be found in the Word Manager database are marked as critical by this module.

4.2.2 Linking to other resources

The second group of text processing operations is devoted to the integration of the text into ELDIT, *i.e.*, to link the words of the text to already available resources, such as dictionary entries. Several tools have been implemented to create these links to the various resources. Direct lexical references to main entries in the learner's dictionary are set by the class *lexicalRef.* For example, the word *Bank* is linked to the corresponding dictionary entry, which contains different lexical meanings for this word.

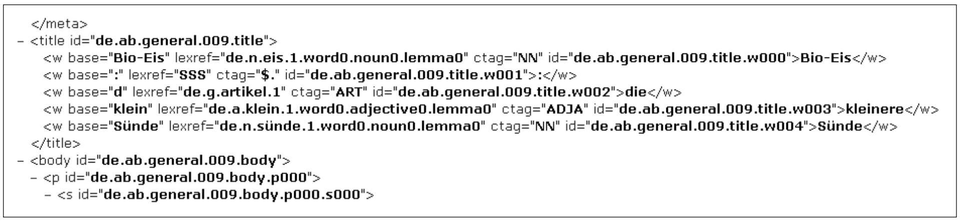
Counting the main entries and words collected into word groups, the dictionary provides about 20.000 words per language. For words that are not (directly) included in the learner's dictionary we have already planned and partially implemented other ways to provide descriptive information. Proper nouns, such as *Judith*, are linked to the dictionary entry 'name'. Compound words such as *Schatzinsel* (treasure island), and derivations such as *Beherrschbarkeit* (controllability) can be analysed by Word Manager and links can be set to the composing entities (*i.e.*, to *Schatz* and *Insel* or to *beherrschen*).

A number of other resources can be imagined to provide useful word meaning descriptions if a certain word is not included in the ELDIT dictionary. For instance, in the EuroWordNet project[6] German and Italian word nets have been developed. In these lexical reference systems, words are organised into synonym sets, each of which represents one underlying lexical concept. Different relations link the synonym sets. Including such word nets in ELDIT, we can link a word to a synonym if the word itself is not included in learner's dictionary.

Last but not least, meaning disambiguation is performed. An ELDIT dictionary entry contains several meanings of a word. A computational linguistics technology called 'context vectors' is now being explored in the context of text authoring in order to disambiguate different meanings of a word and to link directly to the correct word meaning. This technology has already been used and tested in the context of illustrative example re-use (presented above in Section 3.2.4.) and has already demonstrated good results in this new context. Until a reliable disambiguation component is fully developed, ELDIT simply collects and shows all ambiguous meanings to the learner.

Figure 6 shows an excerpt from an XML encoded text sample after it has been processed by the authoring tool. The excerpt shows the title of the text, namely *Bio-Eis: die kleinere Sünde*. Note that each word is annotated with various pieces of information, including a unique ID, lemma information, and a link to the corresponding entry in the ELDIT core dictionary, *e.g.*, the value of the 'lexRef'-attribute of the last word *Sünde* is the ID of the dictionary entry of the word *Sünde*. The explicit annotation of the texts with all these informative tags is necessary in order to provide advanced and adaptive learning support in an efficient way.

Figure 6 XML encoding of a piece of text



```
  </meta>
- <title id="de.ab.general.009.title">
    <w base="Bio-Eis" lexref="de.n.eis.1.word0.noun0.lemma0" ctag="NN" id="de.ab.general.009.title.w000">Bio-Eis</w>
    <w base=":" lexref="SSS" ctag="$." id="de.ab.general.009.title.w001">:</w>
    <w base="d" lexref="de.g.artikel.1" ctag="ART" id="de.ab.general.009.title.w002">die</w>
    <w base="klein" lexref="de.a.klein.1.word0.adjective0.lemma0" ctag="ADJA" id="de.ab.general.009.title.w003">kleinere</w>
    <w base="Sünde" lexref="de.n.sünde.1.word0.noun0.lemma0" ctag="NN" id="de.ab.general.009.title.w004">Sünde</w>
  </title>
- <body id="de.ab.general.009.body">
  - <p id="de.ab.general.009.body.p000">
    - <s id="de.ab.general.009.body.p000.s000">
```

5 Automatic versus automated authoring support

While we report our experience with implementing example extraction and educational text authoring, we would like to discuss an important additional issue: the level of the author's involvement in the process of authoring intelligent content. During the first stage of our work on authoring support, we attempted to automate the authoring process to the fullest degree. Our approach to extracting illustrative examples left the authors entirely out of the explicit process of adding new examples (although the authors could influence this process implicitly by providing text content for example extraction). Our approach to authoring educational texts has so far permitted authors to provide pedagogical content only, such as choice of text, accompanying questions and source citations. However, our experience now pushes us to reconsider our original intentions. Should we really strive to fully automate the intelligent part of authoring, leaving the author only the pedagogical input into this process? Traditionally, there have been two ways to support humans in performing complicated tasks: the AI approach (*i.e.*, make an intelligent system that will do this task for the user) and an HCI approach (*i.e.*, provide a better interface for the humans to accomplish the task). While both approaches are feasible, we are now becoming more interested in a hybrid approach – a 'cooperative' intelligent authoring system that splits the work between a human author and an intelligent authoring support tool, so that both human and AI agents are able to cooperate.

This system can be called automated rather than automatic. Automated or 'cooperative' authoring is now emerging as the main focus of our work on the authoring of intelligent content. We have recently explored some approaches to automated authoring in the fields of computer programming education (Sosnovsky *et al.*, 2004) and the automated authoring of educational texts in ELDIT. The following is a scenario for the type of automated authoring that we are currently exploring.

In the case of text authoring, the challenge for the intelligent support component is POS and meaning disambiguation. Even with an elaborate 'context vectors' technology, we expect the system to be able to disambiguate correctly only up to a certain degree. The rest of the links will lead to ambiguous word meaning descriptions. Humans, however, can easily tell which one of several possible meanings of the word should be used in a text. With a proper interface, teachers will be able to assist the system in the remaining ambiguous circumstances.

The manual disambiguation interface that we are developing consists of two frames. The text to be revised is shown in the left frame. Links in the text are shown in different colours. Words with unambiguous links will be presented in black. Words with ambiguous links will be shown with coloured links. When the author clicks onto a word on the left-hand side, information, which is linked to the clicked word appears on the right-hand side.

Ambiguous information (*e.g.*, two different lemmas for an inflected word form) or ambiguous links (*e.g.*, different meanings of one word) are listed. The author can choose the correct option and record the choice. Moreover, we will give the author an opportunity to correct automatically determined information. For instance, the results of the text classification step can be inspected and are malleable. Finally, misspelled words may be corrected in a simple edit field.

The manual disambiguation interface we are developing provides a good example of cooperative intelligent authoring that we believe will be usable in multiple domains, although in different forms. While we expect that this step will be easy for teachers to perform, we plan to keep it as an optional one. Our goal is to support teachers who are ready to invest more time in order to improve the quality of the material. For more complacent teachers, as well as for those who may be confused with disambiguation or have other reasons to not participate, the system will continue to perform automatic disambiguation the best that it can.

6 Summary and future challenges

In this paper, we advocated a new paradigm for authoring practical, intelligent educational systems. It suggests that the core authoring of an IES should stay in the hands of well-prepared design teams, while the majority of 'intelligent educational content' should be developed by teachers who use the system in their classes. The new design paradigm offers teachers an expanded place in the process of IES authoring by providing a chance to extend the system and to fine tune it to their local needs. We argue that the current bottleneck is the lack of intelligent authoring support tools that would allow regular teachers (who are not IES experts) to contribute content along with all the necessary knowledge that an IES needs to use this content. Extending the pioneer stream of work on authoring support in the field of algebra, we have explored the challenges and the opportunities of intelligent authoring support in the field of language learning.

The main body of the paper presented two implemented examples of intelligent authoring support in the Vocabulary Acquisition System, ELDIT (Gamper and Knapp, 2003). The examples demonstrate that a combination of several intelligent and non-intelligent techniques is required to provide a sufficient level of support. While it is clear that different domains will require different authoring support techniques, we think our description of the overall complexity of the process and the need to involve a range of techniques will be quite typical for a range of domain areas. We hope that our paper will be helpful for development teams in several domains by offering a good example of the elaborate intelligent authoring support that is required to open up the authoring of 'intelligent content' to teachers in relatively rich domains.

The work that we have already completed implements a fully automatic approach to authoring support. After providing the pedagogical content itself, authors do not participate in further knowledge extraction. At this time, we think that one of the future challenges of developing an intelligent authoring tool should be to split the work between a human author and the tool so that both 'agents' are able to do their share of the workload. It is desirable for an authoring system to perform most of the routine operations, leaving the creative part for a teacher. However, our experience shows that in some cases, a teacher may be required to do some routine part of the necessary work, simply because the underlying AI technology is not able to do it. However, the AI technology can dramatically narrow down the amount of necessary manual work. In the ELDIT text authoring tool the problems of automatic meaning disambiguation caused us to start work on manual teacher-assisted disambiguation. Similarly, in the Algebra Tutor authoring tool (Ritter *et al*., 1998a) a teacher is expected to provide a set of variable labels because with the current level of natural language processing technology, it is too expensive to produce an authoring tool that can generate a reasonable set of labels directly from a natural language problem statement. We think that a 'cooperative' approach to content authoring may offer the 'best of both worlds', aiding performance in many domain areas.

One anticipated challenge of this new paradigm is quality control. If the author of intelligent content is the only teacher who uses it in his/her class, the problem of quality is quite small. However, if all contributed content is stored in a central repository and used by multiple teachers, it becomes necessary to establish some kind of quality control mechanism. Since adaptive and intelligent systems are often able to pick out the most appropriate content from the repository 'on the fly' without consulting a teacher, it is necessary to prevent some pedagogically poor content from entering a shared repository.

Some interesting suggestions concerning quality control in an intelligent authoring tool are offered in (Arroyo *et al*., 2001). However, the problem of quality control is not limited to intelligent authoring, for it occurs in any shared repository where authors can contribute their content, such as in pools of educational objects (Verhoeven *et al*., 2001) and in educational digital libraries (Zia, 2004). Modern research on Educational Digital Libraries (EDL) pays a lot of attention to quality control (Sumner *et al*., 2003) and has explored a range of useful methods that could also be applied to intelligent authoring systems.

Another challenge to intelligent authoring that has already been explored in the field of EDL, is the need to broaden the range of content metadata. Unlike EDL, which focuses on 'simple' metadata, such as author, institution and language (Duval and Hodgins, 2003), intelligent authoring tools are focused mainly on knowledge-level metadata. However, the availability of 'simple' metadata could be very useful in practical IESs. For example, if the proper metadata is provided, a teacher who uses an IES with a shared content repository can restrict the selection of content by the adaptive engine to a specific set of authors, institutions or languages.

The final challenge of intelligent authoring research is the evaluation process. Currently, an empirical evaluation is considered to be an important part of a research contribution in the field of IES. The problem is that evaluating an authoring system developed for teachers is a real challenge. It is usually possible to solicit a sufficient number of students to work with an IES for the 1–2 hours that is often sufficient to evaluate a system. However, when the target user of an intelligent authoring system is a teacher, any serious evaluation will need to find teachers to use the system for some considerable amount of time. The need to involve a good number of teachers for a reasonable amount of time provides a serious obstacle for any empirical study. Only when the authoring system is released for practical use may one expect to collect sufficient data! Unfortunately, our team is far from this stage. Even a team that manages to collect some data from teachers using the system faces a complete lack of rubrics or evaluation guidelines. What should be considered a success for an intelligent authoring system? The ability for teachers to use the system? The quality of developed content? The success of students who work with this content? We think that the problem of evaluating intelligent authoring systems should be in the focus of our research community and we hope that this will lead to the establishment of commonly acceptable evaluation guidelines similar to the guidelines that are used for the evaluation of adaptive systems (Chin, 2001).

References

Abel, A. and Weber, V. (2000) 'ELDIT, prototype of an innovative dictionary', *Proceedings of the 9th EURALEX International Congress on Lexicography (EURALEX'00)*, pp.807–818.

Ainsworth, S. and Grimshaw, S. (2004) 'Evaluating the REDEEM authoring tool: can teachers create effective learning environments?', *International Journal of Artificial Intelligence in Education*, Vol. 14, pp.279–312, http://aied.inf.ed.ac.uk/abstract/Vol_14/Ainsworth04.html.

Arroyo, I., Schapira, A. and Woolf, B.P. (2001) 'Authoring and sharing word problems with AWE', in J.D. Moore, C.L. Redfield and W.L. Johnson (Eds.) *Artificial Intelligence in Education: AI-ED in the Wired and Wireless Future*, *(Proceedings of AI-ED'2001)*, Amsterdam: IOS Press, pp.527–529.

Brusilovsky, P. (1992a) 'A framework for intelligent knowledge sequencing and task sequencing', in C. Frasson, G. Gauthier and G. McCalla (Eds.) *Intelligent Tutoring Systems, (Proceedings of Second International Conference on Intelligent Tutoring Systems (ITS'92), Montreal, 10–12 June)*, Berlin: Springer-Verlag, pp.499–506.

Brusilovsky, P. (1992b) 'Intelligent tutor, environment and manual for introductory programming', *Educational and Training Technology International*, Vol. 29, No. 1, pp.26–34.

Brusilovsky, P. and Miller, P. (2001) 'Course delivery systems for the virtual university', in T. Tschang and T. Della Senta (Eds.) *Access to Knowledge: New Information Technologies and the Emergence of the Virtual University*, Amsterdam: Elsevier Science, pp.167–206, http://www2.sis.pitt.edu/~peterb/papers/UNU.html.

Brusilovsky, P. and Peylo, C. (2003) 'Adaptive and intelligent web-based educational systems', in P. Brusilovsky and C. Peylo (Eds.) *International Journal of Artificial Intelligence in Education*, Special Issue on *Adaptive and Intelligent Web-based Educational Systems*, Vol. 13, Nos. 2–4, pp.159–172.

Brusilovsky, P. and Su, H-D. (2002) 'Adaptive visualization component of a distributed web-based adaptive educational system', *Intelligent Tutoring Systems*, *(Proceedings of 6th International Conference on Intelligent Tutoring Systems (ITS'2002), Biarritz, France, 2–7 June 2002)*, Berlin: Springer-Verlag, Vol. 2363 pp.229–238.

Brusilovsky, P., Eklund, J. and Schwarz, E. (1998) 'Web-based education for all: a tool for developing adaptive courseware', *Computer Networks and ISDN Systems (Proceedings of Seventh International World Wide Web Conference, 14–18 April)*, Vol. 30, Nos. 1–7, pp.291–300.

Brusilovsky, V. (1993) 'Task sequencing in an intelligent learning environment for calculus', *Proceedings of Seventh International PEG Conference*, Edinburgh, Edinburgh, 2–4 July, pp.57–62.

Carmona, C., Bueno, D., Guzmán, E. and Conejo, R. (2002) 'SIGUE: making web courses adaptive', in P. De Bra, P. Brusilovsky and R. Conejo (Eds.) *Proceedings of Second International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems (AH 2002) Proceedings*, Málaga, Spain, 29–31 May, pp.376–379.

Chin, D. (2001) 'Empirical evaluations of user models and user-adapted systems', *User Modeling and User-Adapted Interaction*, Vol. 11, Nos. 1–2, pp.181–194.

Duval, E. and Hodgins, W. (2003) 'A LOM research agenda', *Proceedings of the Twelfth International World Wide Web Conference, WWW 2003*, Budapest, Hungary, 20–24 May, pp.88–98.

Elsom-Cook, M.T. and O'Malley, C. (1990) 'ECAL: bridging the gap between CAL and intelligent tutoring systems', *Computers and Education*, Special Issue on *Computers Educ.*, Vol. 15, No. 1, pp.69–81.

Gamper, J. and Knapp, J. (2002) 'Tutoring in a language learning system', *Proceedings of 6th World Multiconference on Systemics, Cybernetics and Informatics (SCI 2002)*, pp.465–470.

Gamper, J. and Knapp, J. (2003) 'A data model and its implementation for a web-based language learning system', *Proceedings of the Twelfth International World Wide Web Conference, WWW 2003*, Budapest, Hungary, 20–24 May, pp.217–225.

Henze, N. and Nejdl, W. (2001) 'Adaptation in open corpus hypermedia', in P. Brusilovsky and C. Peylo (Eds.) *International Journal of Artificial Intelligence in Education*, Special Issue on *Adaptive and Intelligent Web-based Educational Systems*, Vol. 12, No. 4, pp.325–350, http://cbl.leeds.ac.uk/ijaied/abstracts/Vol_12/henze.html.

Knapp, J., Gamper, J. and Brusilovsky, P. (2004) 'Reuse of lexicographic examples in a web-based learners' dictionary', in J. Nall and R. Robson (Eds.) *Proceedings of World Conference on E-Learning, E-Learn 2004*, Washington, DC, USA, AACE, 1–5 November, pp.776–783.

Koedinger, K.R., Anderson, J.R., Hadley, W.H. and Mark, M.A. (1995) 'Intelligent tutoring goes to school in the big city', in J. Greer (Ed.) *Proceedings of AI-ED'95, 7th World Conference on Artificial Intelligence in Education*, Washington, DC, AACE, 16–19 August, pp.421–428.

Lewis, M.W., Milson, R. and Anderson, J.R. (1987) 'The teacher's apprentice: designing an intelligent authoring system for high school mathematics', in G.P. Kearsley (Ed.) *Artificial Intelligence and Instruction: Application and Methods*, Reading, MA: Addison-Wesley.

Murray, T. (1999) 'Authoring intelligent tutoring systems: an analysis of the state of the art', *International Journal of Artificial Intelligence in Education*, Vol. 10, pp.98–129, http://cbl.leeds.ac.uk/ijaied/abstracts/Vol_10/murray.html.

Murray, T. (2003) 'Principles for pedagogy-oriented knowledge based tutor authoring systems: lessons learned and a design meta-model', in T. Murray, S. Blessing and S. Ainsworth (Eds.) *Authoring Tools for Advanced Technology Learning Environments: Toward Cost-effective Adaptive, Interactive and Intelligent Educational Software*, Dordrecht: Kluwer, pp.439–467, http://www2.sis.pitt.edu/~peterb/papers/KluwerAuthBook.pdf.

Murray, T. and Woolf, B.P. (1992) 'Tools for teacher participation in ITS design', in C. Frasson, G. Gauthier and G.I. McCalla (Eds.) *Proceedings of Second International Conference, ITS'92, Berlin*, Montreal: Springer-Verlag, 10–12 June, pp.593–600.

Ritter, S. and Anderson, J.R. (1995) 'Calculation and strategy in the equation solving tutor', in J.D. Moore and J.F. Lehman (Eds.) *Proceedings of the Seventeenth Annual Conference of the Cognitive Science Society*, Hillsdale, NJ: Erlbaum, pp.413–418.

Ritter, S., Anderson, J., Cytrynowicz, M. and Medvedeva, O. (1998a) 'Authoring content in the PAT algebra tutor', in J. Spohrer, T. Sumner and S.B. Shum (Eds.) *Journal of Interactive Media in Education*, Special Issue on *Educational Authoring Tools and the Educational Object Economy*, Vol. 98, No. 9, http://www-jime.open.ac.uk/98/9/.

Ritter, S., Brusilovsky, P. and Medvedeva, O. (1998b) 'Creating more versatile intelligent learning environments with a component-based architecture', in B.P. Goettl, H.M. Halff, C.L. Redfield and V.J. Shute (Eds.) *Lecture Notes in Computer Science*, *(Proceedings of 4th International Conference on Intelligent Tutoring Systems (ITS'98), San Antonio, TX, 16–19 August 1998)*, Berlin: Springer Verlag, Vol. 1452, pp.554–563.

Sosnovsky, S., Brusilovsky, P. and Yudelson, M. (2004) 'Supporting adaptive hypermedia authors with automated content indexing', *Proceedings of Second International Workshop on Authoring of Adaptive and Adaptable Educational Hypermedia at the Third International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems (AH'2004)*, Eindhoven, The Netherlands.

Sumner, T., Khoo, M., Recker, M. and Marlino, M. (2003) 'Digital libraries in the classroom: understanding educator perceptions of "quality" in digital libraries', *Proceedings of the Third ACM/IEEE-CS Joint Conference on Digital Libraries*, Houston, TX, May, pp.269–279.

Verhoeven, B., Cardinaels, K., Van Durm, R., Duval, E. and Olivié, H. (2001) 'Experiences with the ARIADNE pedagogical document repository', *Proceedings of ED-MEDIA'2001 – World Conference on Educational Multimedia, Hypermedia and Telecommunications*, Tampere, Finland, AACE, 25–30 June, pp.1949–1954.

Weber, G. and Bögelsack, A. (1995) 'Representation of programming episodes in the ELM model', in K.F. Wender, F. Schmalhofer and H-D. Böcker (Eds.) *Cognition and Computer Programming*, Norwood, NJ: Ablex, pp.1–26.

Zia, L.L. (2004) 'The NSF national science, technology, engineering and mathematics education digital library (NSDL) program: new projects in fiscal year 2003', *D-Lib Magazine*, Vol. 10, No. 3.

Notes

[1] By core content we mean content that is essential for the functionality of the system. Core content is quite common in adaptive hypermedia systems. For example, presenting a concept in a canonical way is part of the core content. Removal of a core content fragment may break some functionality of the system. In contrast, multiple problems related to a concept are non-core content. The availability of many problems makes a system richer; however, any problem can be removed from the system without affecting its core functionality.

[2] http://www.eurac.edu/eldit

[3] False friends are pairs of words in two languages that look or sound similar but differ in meaning. For instance, the word ‘stipendium’ means ‘grant’ in German, while ‘stipendio’ means ‘salary’ in Italian.

[4] http://www.eurac.edu/bistro

[5] http://www.eurac.edu/miris

[6] http://www.illc.uva.nl/EuroWordNet/