Developing an Intelligent Language Tutor

TRUDE HEIFT
Simon Fraser University

ABSTRACT
This article discusses the development and functionalities of an intelligent computer-assisted language learning (ICALL) system. By way of example, we describe E-Tutor, an ICALL system for L2 learners of German that has been in use for a decade. Based on advancements in technology, user studies with L2 learners of German as well as feedback provided by language instructors, students, and researchers, the system has undergone several system updates since its initial implementation in 1999. This article focuses on the pedagogical benefits of ICALL systems by also providing reflections on the development of such systems that resulted from the lessons we have learned in building and maintaining E-Tutor.

KEYWORDS
Intelligent Computer-assisted Language Learning (ICALL), Natural Language Processing (NLP), Parsing, German as a Second Language

INTRODUCTION
E-Tutor, an intelligent computer-assisted language learning (ICALL) system for learners of German, integrates natural language processing (NLP) and artificial intelligence (AI) modeling into CALL. NLP techniques model "understanding" of human language by computer, while AI techniques can be used to model the individualized learning experience, thus aiming at learning programs that come closer to natural language interaction between humans than has been the case in traditional CALL. The following discussion focuses on the functionalities of E-Tutor and the benefits of ICALL systems to language learners and describes some of the challenges that are posed by their underlying technologies. E-Tutor has been in use for a decade, and, not surprisingly, it has undergone many system upgrades and expansions since its initial implementation in 1999. This article reflects on the development, implementation, and maintenance of such systems.

NLP and AI modeling have played a significant role in the development of our thinking about CALL, its design and implementation. For instance, Nerbonne (2003), in his chapter on NLP in CALL in the Oxford Handbook of Computational Linguistics, argues that recent advances in NLP have much to contribute to CALL. Specifically, Nagata (1996) concludes from one of her user studies that only CALL programs that make use of the full potential of the computer, mainly by providing immediate and appropriate feedback (something a workbook cannot easily do if at all), will produce higher learning results. However, the development of an NLP system along with its integration into a CALL package is a very complex, onerous, and extremely time-consuming endeavor, largely due to its sophisticated underlying technology. Given these increased development efforts, what makes ICALL systems so desirable from a pedagogical point of view? What benefits do these systems provide over more traditional CALL systems? What are the challenges that we face in designing and maintaining such systems?
E-TUTOR: LEARNING CONTENT

E-Tutor is a noncommercial, web-based ICALL system for L2 learners of German that covers the content of the first three university courses of German during which the main components of the L2 grammar are generally taught. The system follows the grammatical and vocabulary sequence of Deutsch: Na klar! (Di Donato, Clyde, & Vansant, 2004), a textbook commonly used in North America for L2 learners of German. The system is commonly used in conjunction with regular face-to-face instruction.

E-Tutor (initially called ‘German Tutor’ and renamed after a complete makeover) covers learning content that is distributed over a total of 15 chapters. Each chapter begins with an introductory text (e.g., story, dialogue) that highlights the focus of the chapter. For instance, Figure 1 shows the introductory page of Chapter 3 of E-Tutor, which centers around the topic of family and friends. From here, learners then have access to the chapter contents (Contents tab) as well as a bilingual dictionary (Dictionary tab) that contains approximately 20,000 entries.

Figure 1
Introductory Screen of Chapter Three
Each chapter offers different learning activities that allow students to practice chapter-related vocabulary and grammar. In addition, each has learning activities for listening and reading comprehension, culture, and writing. There are currently nine activity types implemented in the system (e.g., sentence building, reading comprehension, essay). For example, in the sentence-building activity, students are asked to construct a sentence from grammatical cues and words that are provided in their base forms. For the reading comprehension activity, students study a chapter-related text and answer comprehension questions. For the essay, students write a minimum of 50 words at the introductory level. As with the other activity types, the essay topics vary by chapter and are always closely related to the content, vocabulary, and grammar included in each chapter. Students have also access to web links that relate to the current chapter content, authentic pictures, grammatical notes, statistics on system use and user performance, as well as learner progress reports. The content of the system has been added incrementally. The first version of E-Tutor covered only the learning content for the introductory course and contained only a limited set of activity types, mainly aimed at practicing grammar. By contrast, in the current version the learning content and activity types alone easily replace the workbook that usually accompanies textbooks.

E-Tutor provides a more traditional learning environment in which learning activities are performed individually. This contrasts with Web 2.0 learning environments (e.g., blogs, webquests) in which language learners create and design their own place by working collaboratively with peers. A recent study by Peters, Weinberg, and Sarma (2009) that involved five Canadian universities investigated language students’ perceptions about the effectiveness of technology. The study found that traditional types of computer-assisted activities are more appreciated and judged more useful than newer types of activities such as blogs and webquests. Moreover, the study revealed that students tend to prefer and judge as more useful computer-assisted activities that are noninteractive or performed individually. These results emphasize the value of more traditional learning environments and their place in the current language classroom, although a combination of the different kinds of technologies will likely prove most useful.

DESIGN CONSIDERATIONS AND SYSTEM ARCHITECTURE

The design underlying E-Tutor is strongly motivated by pedagogical considerations. We aimed at a CALL system that emulates a learner-teacher interaction by focusing on individualized interaction between the learner and the CALL system. For this purpose, two main design criteria have to be met. First, the system needs a sophisticated answer-processing mechanism because it is simply not feasible to anticipate every mistake a student might make. Second, the system needs to collect and maintain information about its users and their behavior while they are working with the CALL program.

Colpaert (2006), in describing different approaches to software development, also advocates a pedagogically driven approach but, at the same time, alludes to a problem of system design, namely bridging the gap between language pedagogy and technology. A CALL program might include the latest technological fads but lack language pedagogy, or it might reflect sound language teaching pedagogy but not effectively exploit the technology. On the other hand, even the best team of CALL software designers cannot always anticipate the ways in which learners will use a CALL system. Many CALL studies have shown a discrepancy between the intention behind certain software features and students’ actual use of them. For this reason, E-Tutor followed a cyclical process of development, implementation, and evaluation (see Colpaert, 2006). Accordingly, the system has undergone many changes over the
past 10 years which mainly resulted from the outcomes of our user studies, in particular, with regard to interface design, error analysis, help options, and system feedback. Some of these changes were as simple as modifying the font size for increased visibility, making adjustments to the NLP component of the system, or adding help options.

The underlying NLP and modeling components (i.e., the expert module and learner model) form part of the backbone of the system and have to be developed in addition to the learning content and the user interface. Yet, while these system components have their respective responsibilities, it is the interaction between them that lends unique power to an ICALL system. For instance, the front end of E-Tutor consists of a web interface that contains the overall layout of the ICALL program, displays its learning content (e.g., activity types and additional learning tools) and provides a means for the interaction between the user and the ICALL system (e.g., see Figure 1 above). In this front end are the first differences the user encounters between ICALL and more traditional CALL systems. For an ICALL system to individualize instruction by providing a unique set of system responses and interactions, it must keep a record of each user and exchange that information among all system components. At login, the web interface of E-Tutor then constructs or updates the learner’s profile (with information such as performance history, previous access, and completion of exercises) and obtains from the teaching module the learning content appropriate for the student.

The core of E-Tutor provides the computational and analytical power of the system. The expert module contains the NLP component which performs a linguistic analysis of learner input by checking for correct syntax and morphology1 (for a more detailed description of the expert module, see Heift & Nicholson, 2001). Ideally, this component is designed in a modular and fairly general way such that the underlying linguistic analysis does not have to be altered for each activity type (see also Amaral, 2009). In E-Tutor, all activity types use the identical NLP component, which, however, allows flexibility in the feedback generated for each activity type.2 Once the NLP component has identified the correct and incorrect structures in the learner’s input, it communicates with the learner model to obtain learner and task-specific information for feedback generation. The learner model also receives ongoing updates from the teaching module concerning the learning activities the student has worked on or completed.

The learner model provides a dynamic assessment of each learner by considering past and current learner performance and behavior in relationship to a particular learning activity (see also Amaral & Meurers, 2007; Heift & Schulze, 2007). Thus the system’s interaction with each student is individualized as to the kinds of errors found in the student input as well as the ways those errors are communicated to the student. For instance, the system might change the order in which errors are displayed for a particular learner depending on the learner’s past performance or the focus of the learning activity. In addition, the wording of the feedback may differ for different skill levels of learners because some students require more detailed feedback on their errors than others. The feedback messages in E-Tutor can also be geared to a particular activity type, that is, the generic message for each error can be altered and adjusted to a particular exercise. By separating the feedback generation from the linguistic analysis and learning activity, an ICALL system obtains a great deal of flexibility in that, for instance, some errors might even remain unreported for some students and certain learning activities if reporting is deemed unnecessary from a pedagogical point of view.

In the following we will focus on three aspects of E-Tutor that highlight the pedagogical benefits of an ICALL system by concentrating on its feedback modules, tracking devices for learner progress, and a learner corpus that the system generates during system use.
Pedagogical Benefits of ICALL Systems: Feedback

One of the main advantages of ICALL systems lies in their capability to generate informative and meaningful feedback. This is due to the linguistic analysis the system performs on the learner’s input. Over the past two decades, a number of studies have investigated different feedback types and their effect on learning outcomes (e.g., Bowles, 2005; Heift, 2004; Heift & Rimrott, 2008; Nagata, 1996; Nagata & Swisher, 1995; Pujolà, 2001; Rosa & Leow, 2004). The results of these studies generally support the claim that students benefit from the more explicit feedback because they subsequently perform better on particular target language structures or because their grammatical awareness is subsequently raised.

In E-Tutor, the feedback generator, which is part of the expert module, correlates the detailed output of the linguistic analysis that the system performs with an error-specific feedback message. For instance, in Figure 2, the learner was asked to construct a sentence with the words provided by the prompt. The interface for the learner feedback consists of a display field with four tabs that are consistent for all activity types: feedback, history, grammar help, and dictionary.

Figure 2
Learner Feedback
Under the feedback tab, the system displays feedback by highlighting the error in the learner’s input followed by a meta-linguistic explanation of the error. In the case of a lexical error (e.g., spelling), the system provides a link to E-Tutor’s bilingual dictionary (Dictionary tab). If the learner committed a grammatical mistake, the system links to the grammatical paradigm that corresponds to the learner’s error. Thus, in Figure 3, the learner made an error with the main verb werden (to become) by entering the second person plural form instead of the third person singular. The system provides a link to the inflectional paradigm of werden (Grammar Help tab). These context-sensitive grammatical paradigms are made possible due to a fully specified lexicon which is part of the NLP component of the system and from which the paradigms are constructed during runtime.

Figure 3
Verb Paradigm for werden

For pedagogical reasons, the error checking process of E-Tutor is iterative, that is, the system identifies and communicates one error at a time to learners. Once learners have revised the input, they resubmit the sentence for further analysis. The iterative process con-
continues until the sentence is correct, or until learners click the SOLVE button. For this reason, the system also displays a History tab (see Figure 4) that allows learners to examine their previous submissions for an exercise by also still linking to the context-sensitive help options that the system displayed for each individual submission. This then allows learners to revisit the grammatical paradigms or vocabulary items that were violated in their previous input.

Figure 4
History of Student Submissions

Finally, and due to the dynamic learner model of the system, E-Tutor also provides preemptive focus on form by presenting a feed-forward system shown in Figure 5 (for a discussion of preemptive feedback, see Ellis, Basturkmen, & Loewen, 2001).
On the basis of the learner's performance history and the overall difficulty of a given learning activity, *E-Tutor* alerts students if it believes that the activity is a particularly difficult exercise for them. In Figure 5, the system response prompts learners to first review the dative case before attempting the exercise. The overarching goal here is to prepare students for activity types gradually by providing additional resources along the way as deemed necessary.

**Learner Progress**

In addition to informative and meaningful feedback on learner input, an ICALL system can also provide ongoing assessment and individualized feedback on learner progress due to its NLP component. *E-Tutor* contains an extensive tracking system: besides a unique student ID and time stamp, the log records the entire interaction between the computer and the student. This includes the activity type, the student input, the system feedback, and navigation patterns. This information is stored in the Report Manager that forms part of *E-Tutor’s* learner
model. The Report Manager provides the interface to an inspectable learner report with the following functions:

1. The learner report bookmarks and tracks exercise completion by collecting and retaining information on the learner’s progress and performance that is saved between visits.
2. It contains a journaling system that records prior inputs along with a detailed error analysis that can be printed or emailed to an instructor.

The Report Manager provides access to a bookmarking system that tracks exercise completion for each user and exercise set. Figure 6 shows the typical content of a chapter in E-Tutor. The content page for each chapter provides a link to the journaling part of the Report Manager ("Manage your reports") and a brief explanation of the bookmarking and tracking symbols used in the system.

Figure 6
Content Page for Chapter 3
The pencil icon in Figure 6 shown in front of the link “Ex. 7: Read a story and answer questions” under the “Context” heading indicates that the reading comprehension exercise is in progress but the student has not yet completed it. The report card icon displayed in front of “Ex. 6: Listen to a short story” under the “Context” heading indicates that the exercise set has been completed. The dotted square icon that appears in front of "Ex. 1: Find the matching word” under “Vocabulary” indicates that the student has not yet accessed this particular exercise type. For each activity type, the system saves the information for each student and each session, thereby allowing students to always continue their work from the point at which they stopped in their previous session. This is a feature that learners appreciate given that each chapter of *E-Tutor* contains approximately 70-80 individual exercises which would be cumbersome to complete in one uninterrupted session.

In addition to the learner’s progress, the Report Manager also saves information on the student’s performance. The journaling system of the Report Manager records prior inputs along with detailed error reports. Figure 7 shows the summary page for the activity types that are either completed or in progress. Exercise types that the student has not yet worked on do not appear in the report.

**Figure 7**
Progress Report and Performance Summary

The Report Manager displays the name of the learning activity, the completion date, and a final performance summary for each activity. For example, in the “Vocabulary: Ex. 2” activity type in Figure 7 the student scored 90%. For activity types that are in progress students can resume the activity type (“resume”). If a student chooses to repeat an entire exercise set, the associated report has to be deleted first by clicking the “delete” button. This will delete the previously saved summary of the student’s performance in this exercise type. Finally, students can also obtain a more detailed summary of their performance on a completed activity type by clicking on “view report.” Figure 8 shows a comprehensive report for the build-a-sentence exercise type. In addition to the completion date, students receive a summary of the total number of correct, incorrect, peeked, and skipped exercises followed by a detailed breakdown of their errors.
In Figure 8 the student did not skip any exercises (skipped) or look up the answer (peeked) during exercise completion. The summary in Figure 8 further shows that for four out of the 10 exercises, the student obtained the correct answer on first try. For the remaining six exercises, the student required several tries committing the errors listed in the error breakdown in the lower half of Figure 8: the student made mistakes with capitalization, modals, spelling, verb inflection, and word order. The results for each exercise type can also be printed or emailed to an instructor. In addition to individual student assessment, the instructor can compile the information into a class error history by combining the correct or incorrect responses that were made by all students in the class.

Finally, learners can also inspect their overall performance since initial system use and compare it to that of previous users. In some instances, this might be even more informative than individual learner reports because the difficulty of the activity type is leveraged by a user comparison. For example, Figure 9 shows the number of correct responses as well as a breakdown of all errors (totals and percentages) for a particular user and those for all previous users (4,378 in total).
Figure 9 further indicates that, for all exercises completed, the student performed much better on word order than previous users while making more grammar and fewer spelling mistakes. The search terms can be adjusted to display only the results for certain chapters, exercise types, and individual exercises. These statistics are made possible due to the learners’ profiles that are maintained by the Report Manager in the learner model.

**Learner Corpus: Exploring Learner Language**

Intelligent Language Tutors are primarily designed to support form-focused instruction (see Schulze, 2008). However, ICALL environments are certainly not limited to this kind of instruction; they allow for more diverse learning environments. For instance, Dickinson, Eom, Kang, Lee, and Sachs (2008) designed an ICALL system that is embedded in a synchronous computer-mediated communication (CMC) environment. The system provides feedback on particle usage for first-year L2 Korean learners while they chat in CMC. Moreover, Harbusch, Itsova, Koch, and Kuhner (2008) designed a virtual writing environment for German for elementary-
school children, *The Sentence Fairy*, which deploys natural language generation technology to evaluate and improve the grammatical quality of student output. Most other ICALL systems provide both form- and meaning-focused instruction. For instance, the activity types in *E-Tutor* combine form-focused instruction (e.g., grammar) with activities that emphasize primarily meaning (e.g., activity types for practicing reading comprehension or cultural knowledge). In addition, *E-Tutor* also supports discovery learning in the form of exploration of learner language (see Leech, 1997). For this purpose, all user submissions over a time period of 5 years were compiled, and a common learner corpus was constructed from those submissions that allows students to explore learner language according to various parameters. In addition, each student constructs his or her own corpus while using the system. Figure 10 displays the interface of the searchable learner corpora.

Figure 10
Learner Language Display

As indicated in Figure 10, learners can choose between the common learner corpus that combines all user submissions, or they can limit the search to only their own corpus. If they select to search all user records, they can also specify the time period for which they want to execute their search (e.g., all 5 years or one particular year). In addition, users can filter their search by exercise type, chapter, and error category and thus inspect what kind of errors commonly occurred in which chapters and activity types. Finally, learners can also
choose between the hardest and easiest sentences. The hardest sentences are those in which either most learners committed an error (sort by “number of users”) or those that contained the most errors (sort by “number of errors”). Conversely, the easiest sentences are those that were submitted correctly by most learners (sort by “number of users”) or the sentences that contained the fewest errors in total (sort by “number of errors”). The lower half of Figure 10 displays the search results; in this instance, the exercise type and instruction, the number of users and errors, and the sentence that, according to the parameters selected, was answered incorrectly by students.

While this tool is fairly easily constructed due to the linguistic analysis of user input and the log that the system constructs from all user inputs, it has multiple applications. Most important, learners can examine interlanguage or task-specific phenomena, the benefits of which have been well documented (see Borin & Dahllöf, 1999; Braun, 2005; Gaskell & Cobb, 2004; Granger, 2003). In addition, by comparing their own data to those of all users, learners can determine in which specific ways their interlanguage differs from that of other users. It also allows language instructors and/or researchers to examine the design of language learning material because the statistics reveal much about the degree of difficulty of each exercise, in particular, if they have been tested on approximately 5,000 learners. Finally, these large data collections also allow for investigations of a wide range of additional research topics (e.g., use of help options, interlanguage studies) to be explored by researchers.

CONCLUSION

Over the past decade, E-Tutor has been used successfully with L2 learners of German along with regular face-to-face instruction. While system maintenance is inevitable due to technological innovations, a number of fundamental design decisions paid off for us in this regard. Most important, the modular design of E-Tutor (e.g., the NLP module itself but also the separation of the NLP module from the feedback generator and other pedagogical components) have allowed us to adjust and extend the system without having to revisit each and every system component when upgrades were performed. In fact, the NLP analysis with its lexicon and rules has remained almost identical since its initial implementation. We have been able to concentrate on extending the learning content solely by adjusting some of the feedback responses to particular activity types or adding further functionalities to the system in the form of help tools and record-keeping facilities toward the goal of enhancing students’ overall learning experience. Any full-fledged CALL program certainly requires thorough initial planning and a fair amount of flexibility and anticipation of the final functionalities of the system on the part of the program designer(s).

Moreover, there is no doubt that building any CALL system that is used in the actual language learning classroom is an onerous task. In addition to the many design considerations during the development phase, the system also has to run smoothly without demanding too much technical assistance. This is even more important if little designated technical support is available, which is the case in most academic settings. For this reason, E-Tutor has intentionally avoided proprietary software tools (e.g., Flash), mainly because user support can become cumbersome if the user is not equipped with the latest computer technology. Instead, E-Tutor relies on a web browser with no additional plug-ins and system requirements and, to this end, has demanded little ongoing technical user assistance.

While an NLP component adds impressive power and benefits to a CALL program because of its capacity for a deep analysis of learner input and individualization of the learning process, in the final analysis the system must exploit any technological component to the
fullest with respect to language learning pedagogy. This means that both learners and instructors must experience immediate benefits. In E-Tutor, it is the coherence and interaction of individual system components that led to a pedagogically driven NLP system.

ACKNOWLEDGMENTS
The author owes deep gratitude to the many students, friends and, last but not least, family members who worked with her on the development of E-Tutor over the last decade. Most important is Chris Rolfe, who programmed most aspects of the current version of E-Tutor and performed the audio recordings. Parts of this research were also supported by Social Sciences and Humanities Research Council (SSHRC), Canada, grant 632209. Finally, I would like to thank the editor of this special issue for her insightful comments on an earlier draft of this article.

NOTES
1 For a system to analyze meaning, a semantic component is required, which has not been the focus of E-Tutor so far (for a description of such a component, see Meurers, 2009).
2 Some of the activity types, however, are not graded by the NLP component of the system (e.g., the writing assignment).

REFERENCES


AUTHOR’S BIODATA
Trude Heift is Associate Professor in the Department of Linguistics at Simon Fraser University. Her work bridges applied and computational linguistics by studying different aspects of learner-computer interaction in adaptive parser-based CALL systems.

AUTHOR’S ADDRESS
Trude Heift
Linguistics Department
Simon Fraser University
Burnaby, BC
Canada V7M1K6
Phone: 778 782 3369
Email: heift@sfu.ca