Student Modeling and *ab initio* Language Learning

Trude Heift
Linguistics Department
Simon Fraser University
E-mail: heift@sfu.ca

Mathias Schulze
Dept. of Germanic & Slavic Studies, University of Waterloo
E-mail: mschulze@uwaterloo.ca

Abstract: This paper provides an overview of student modeling techniques that have been employed in Computer-Assisted Language Learning (CALL) over the past decade. We further discuss two of our own systems and show how different types of CALL programs can, nonetheless, share similar conceptual designs of a student model. First, we describe the ***German Tutor***, an Intelligent Language Tutoring System (ILTS) for German as a Second Language which contains a parser and a grammar that analyze student input. The Student Model is based on student subject matter performance and provides feedback and remedial exercises suited to learner expertise. We further report on a study of the ***German Tutor***in which we determined the extent to which our Student Model addresses the need for an individualized language learning environment. Second, we provide an overview of ***Geroline,*** a Web-based distance education course for ***ab initio*** German learners, and its student model. We show how a student model supports computerized adaptive language testing for diagnostic purposes.

Keywords: computer-assisted language learning, student modeling, web-based learning systems, German as a Second Language, diagnostic adaptive language testing

1. Introduction

> Student Modelling is a centrally important issue in ITS research. Without a student model, an ITS would be doomed to follow a preset sequence of steps regardless of impact of its actions on student's learning. Unfortunately, student modelling is also a very difficult problem. It touches many of the great issues of artificial intelligence and cognitive science: diagnosis, belief revision and truth maintenance, qualitative reasoning, mental modelling, temporal reasoning, non-monotonic and probablilistic reasoning, testing and evaluation, etc. (Greer and McCalla 1994, V)

Individualized language instruction has long been recognized as a significant advantage of Computer-Assisted Language Learning (CALL) over workbook tasks. A "one size fits all" approach is not appropriate for a learning environment. Students learn at their own pace and often, work for their own purposes. Learners also vary with respect to prior language experience, aptitude, and/or learning styles and strategies. According to the Individual Differences Theory as described by Oxford (1995), if learners learn differently, then they likely benefit from individualized instruction.

Despite the need for an individualized learning environment, student modeling has not been a strong focus of CALL. One likely reason is that in order for a computer program to adapt itself to different learner needs, the system needs a dynamic model of the strengths and weaknesses of the learner (McCalla & Greer, 1992). Commonly, this is achieved by an Intelligent Language Tutoring System (ILTS) that contains a grammar and a parser that analyze student input. However, even when it comes to ILTSs, only a few have employed student models to individualize the learning environment. According to Holland & Kaplan (1995), this is likely due to the challenging task of representing the domain knowledge itself, the module which contains facts and information about the language being taught. If the grammar is not accurate and complete, even a precise student model cannot compensate. For instance, Holland (1994) states that a system which does not detect ambiguous errors accurately will obscure a student model.

There are a number of modeling techniques that can be implemented in a computer program. The system can model subject matter performance, students' learning strategies and/or cognitive styles. ILTSs primarily model subject matter performance, that is, students' surface errors. While such a student model might not be complete, it assists in individualizing the language learning process and "is sufficient to model the student to the level of detail necessary for the teaching decisions we are able to make" (Elsom-Cook, 1993: 238).

In this paper, we describe the Student Model of the ***German Tutor***, our Web-based ILTS for German as a Second Language. The ***German Tutor*** analyzes sentences from the student and detects grammatical and other errors. The feedback modules of the system correlate the detailed output of the linguistic analysis with an error-specific feedback message. The Student Model is based on student subject matter performance. It provides feedback and remediation suited to learner expertise. We further introduce ***Geroline***, a Web-based course for beginner German (currently under development). ***Geroline's*** student model shares some of the underlying ideas of the ***German Tutor*** in that it also aims to provide the student with information on relevant exercise material. This system, however, makes use of techniques developed in computer-aided language testing and does not rely on a sophisticated parser-based grammatical analysis.

In the following section, we provide examples of CALL systems that employ student models and discuss their distinct emphasis. In section 3, we describe the architecture of the Student Model of the ***German Tutor***. Section 4 reports on a study in which we determined the extent to which our Student Model addresses the need for an individualized language learning environment. In section 5, we describe ***Geroline's*** student model and show how it can be utilized in diagnostic self-testing. Concluding comments are provided in section 6.

2. Student Modeling and Intelligent Language Tutoring Systems

In analyzing student models, McCalla (1992) makes a distinction between implicit and explicit student modeling which is particularly useful in classifying the student models in ILTSs.

An implicit student model is static, in the sense that the student model is reflected in the design decisions inherent to the system and derived from a designer's point of view. For instance, in an ILTS the native language of the learner can be encoded as a bug model that includes frequently made errors and ultimately diagnoses them.

In contrast, an explicit student model is dynamic. It is a representation of the learner which is used to drive instructional decisions. For ILTSs, for instance, the student model can assist in guiding the student through remedial exercises or it can adjust instructional feedback suited to the level of the learner. In either case, the decisions are based on the previous performance history of the learner. The following discussion will provide examples of ILTSs which have implemented implicit and explicit student models.

2.1. Implicit Student Models

Implicit student modeling has been applied to ILTSs to diagnose errors. For example, in Catt & Hirst's (1990) system ***Scripsi*** the native language of the student represents the learner model. It is used to model the learner's interlanguage. With regard to student modeling, the pitfall of such an implementation is that it is a static conception. The system's view of the learner cannot change across interactions with the system. It has no impact on instructional decisions and provides only a gross individualization of the learning process when ideally, a student model is dynamic (Holt et al., 1994).

In a more individualized example, Bull (1994) developed a system that teaches clitic pronoun placement in European Portuguese. The student model is based on the system's and the student's belief measures, language learning strategies, and language awareness. The system's belief measure is comprised of the proportion of incorrect/correct uses of the rule; the students provide the data for the student's belief measure, being required to state their confidence in their answer when entering sentences. Learners also identify their preferred learning strategies when using the program. According to Bull (1994), language awareness is achieved by allowing the student access to all information held in the system. The information, however, is not used to drive the instructional process. A number of studies have also shown that students tend to not take advantage of the option to access additional information. For example, Cobb & Stevens (1996) found that in their reading program learners' use of self-accessible help was virtually non-existent, in spite of their previously having tried it in a practice session, and also having doubled their reading performance as compared to either a no help or dictionary help option in the practice session.

2.2. Explicit Student Models

In developing an explicit student model one typically starts by making some initial assumptions based on pretests or stereotypical postulations about the learner. For example, initially every student could be assessed as an intermediate. During the instructional process, the student model adjusts to student's behaviour moving to a novice or expert profile, as appropriate. This technique is used in explicit student models to make instructional decisions.

Explicit student modeling has been used in a number of ILTSs, primarily in the form of tracking. Tracking can be as simple as calculating percentages of correct answers or more sophisticatedly, identifying particular errors which occurred in the student's input. The information is then used to alter the instructional process, either in the form of further language tasks or feedback.

Explicit student modeling is found in the system ***The Fawlty Article Tutor*** (Kurup, Greer & McCalla, 1992) which teaches correct article use in English. The system presents the student with scenarios whereby the student must select the correct article form and the appropriate rule. The tutor keeps an error count and selects the scenarios on the basis of the performance of the student; thus the path through the program is individualized by altering the instructional process according to prior performance of the student.

Bailin (1988, 1990) in his system ***Verbcon/Diagnosis*** also employs the tracking method. ***Diagnosis*** provides practice in using English verb forms in written texts. All verbs are presented in their infinitival form challenging the student to provide the appropriate verb form. The system tracks the most frequent error occurrence and the context in which the error occurred. The information is used to provide informative feedback based on contrasting correct and ungrammatical uses of tenses. In addition, ***Diagnosis*** suggests exercises to help with the remediation process.

Virvou et al. (2000) use a student model for their ILTS. The student model in this multimedia tutoring system for the passive voice in English supports the error diagnosis process. "In order to provide individualized help, the system holds a profile for every student ..." (ibid.)

In the following section we describe the ***German Tutor*** and discuss the modeling technique used.

3. The *German Tutor*

The goal of the ILTS we have developed for German is to provide meaningful and interactive vocabulary and grammar practice for second language learners. The ***German Tutor*** analyzes sentences from the student and detects grammatical and other errors. The feedback modules of the system correlate the detailed output of the linguistic analysis with an error-specific feedback message.

Figure 1: System Feedback for the Advanced Learner

In the ***German Tutor***, feedback is individualized through an adaptive, explicit Student Model, which monitors a user's performance over time across different grammatical constructs. This record of strengths and weaknesses is used to tailor feedback messages to learner expertise within a framework of guided discovery learning: a beginner student will receive the most explicit feedback while the instructional messages for the expert will merely hint at the error. The feedback aimed at the beginner will also contain less technical terminology than that for the intermediate and expert. For example, Figure (1) shows a feedback message for an advanced student, which indicates incorrect determiner-noun agreement.

In contrast to Figure (1), the feedback message for the beginner learner will provide less linguistic terminology and state that ***The determiner DER is not correct here.*** For the intermediate learner, the feedback will provide less of a clue and simply display ***There is an agreement error between the determiner and the noun.***

In the following section we discuss the technique employed in the Student Model of the ***German Tutor***.

3.1. The Student Model of the *German Tutor*

The Student Model of the ***German Tutor*** dynamically evolves based on the student's performance. The information in the model is used for two main functions: modulation of instructional feedback and assessment and remediation.

The Student Model keeps track of an individual student's performance on a variety of grammatical phenomena (e.g.: agreement, modals, verb particles) based on the information obtained from the grammar and the parser. The goal of the parser and the grammar is the generation of phrase descriptors, each of which describes a particular grammatical constraint, its presence or absence in the input sentence and the student's performance on this constraint. Phrase descriptors correspond to structures in the Student Model and are the interface medium between the Student Model and the grammar of the system.

A phrase descriptor is implemented as a frame structure that models a grammatical phenomenon. Each member of the frame consists of a name followed by a value. For example, subject-verb agreement in number is modeled by the frame [number, ***value***] where ***value*** represents an as yet uninstantiated value for number. If the grammatical phenomenon is present in the student's input, the value is either ***correct*** or ***error*** depending on whether the grammatical constraint has been met or not, respectively. If the grammatical constraint is missing, the feature value is ***absent***. Consider examples (1a) and (1b):

(1a) *Er gehen.
(1b) Er geht.
He is leaving.

The phrase descriptor for subject-verb agreement in number in example (1a) is [number,***error***], while that for the sentence in (1b) is [number,***correct***]. For either sentence, (1a) or (1b), the information will be recorded in the Student Model. A system presented with (1a), however, will also instruct the learner on the nature of subject-verb agreement in number.

In addition to the grammatical features defined, the grammar uses a type ***descriptor*** representing the description of the phrase that the parser builds up. This type is set-valued and is initially underspecified in each lexical entry. During parsing, the values of the features of ***descriptor*** are specified. Ultimately, descriptor records whether the sentence is grammatical and what errors were made.

For each grammar constraint, the Student Model keeps a counter for each student with a score for each grammar skill. This score ranges from 0 - n, where we have set n to 30. The score increases when the student provides evidence of a successful use of that grammar skill, and decreases when the student provides evidence of an unsuccessful use of that grammar skill. The amount by which a student's score increases or decreases can vary depending on the current value of the score. Initially, we set all scores to an intermediate level but pre-testing can determine individual differences from the offset.

For the purposes of modulating instructional feedback, we identify 3 categories of scores. Scores from 0-10 are assigned to the novice category, 11-20 to the intermediate category, and 21-30 to the expert category. When a student makes an error on a particular grammar skill, the message they receive depends on their score for that skill. If they are ranked as novice, they will receive a more informative message than if they are ranked as an expert. Since the score for each grammar skill is independent of the score for the other grammar skills, a student may be expert at subject-verb agreement, but novice at forming the passive - and receive the appropriate message.

The score information is also used for a variety of remediation and assessment tasks. By comparing the Student Model at the beginning and end of a session, we can provide a summary of the mistakes that a student made during that session (see Figure 2). In our current system, these are summarized into general categories such as "Verb Tenses", "Pronouns", etc. These groups are set by means of a parameter file. Similarly, we can also identify the grammar skills where the student was correct and provide a "positive" of what the student did right. At present we show a list of the errors at the end of each exercise set.

Further, one can also examine the Student Model overall and identify the current strengths and weaknesses of the student. We identify the strengths of a student as the five highest scoring grammar skills that have a score greater than 15 (half of the total scale). We identify the weaknesses of a student as the 5 lowest scoring grammar skills that have a score less than 15. Students can access this information.

Finally, the Student Model information can also be used to provide exercises to the student which focus on their areas of weaknesses. Instead of repeating the same exercise which the student made the mistake on, the ***German Tutor*** has the capacity to identify examples which require the same grammar skill. This avoids the problem of the student rote learning the solution to a particular example, without actually learning the general solution. Figure (2) shows an example of a student who is asked to practice subject-verb agreement in an additional exercise set.

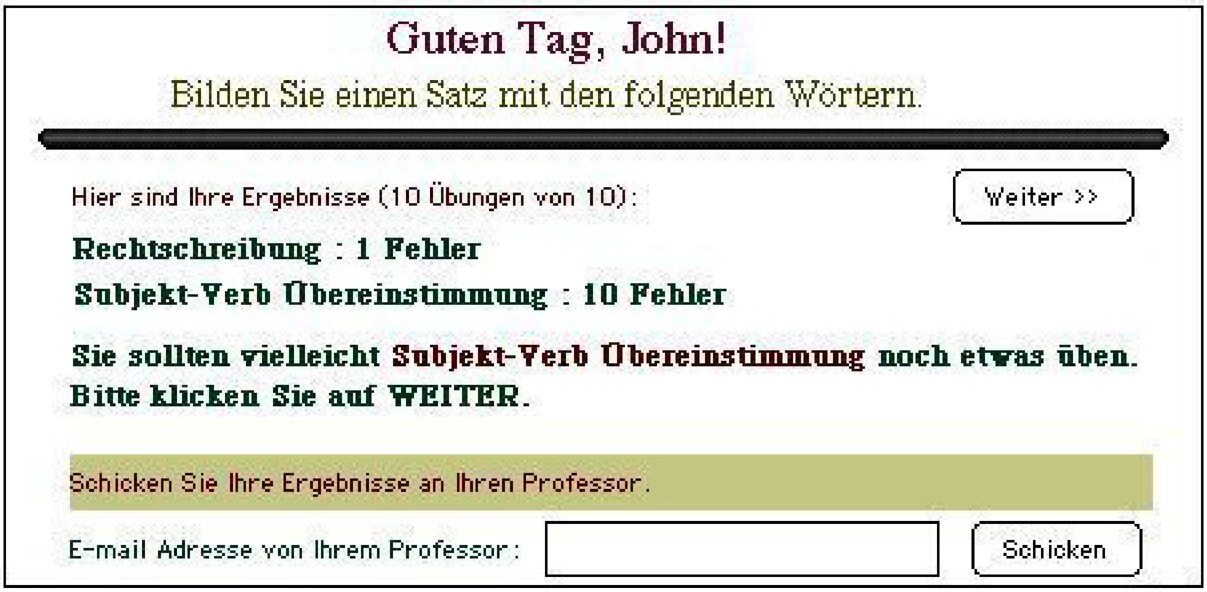


Figure 2: Summary Page

4. Evaluation

The ***German Tutor*** (Heift & Nicholson, 2001) has been tested extensively. In one of the studies, we determined the extent to which the Student Model addresses the need for an individualized language learning environment. 33 students participated in the study and a total of 1352 sentences were considered for analysis.

When analyzing the data with respect to individualized instruction, we were interested in the types of errors that occurred during practice and their distribution with respect to the three learner levels: beginner, intermediate and advanced.

The error break-down in Table (1) shows that students were most often at the intermediate level, which is not surprising since each student is initially placed at the intermediate level. Nonetheless, approximately one third, or 30%, of the time, students either required more elaborate feedback suited to the beginner learner, or, in the case of the advanced learner, less detailed feedback was sufficient to correct the errors. Moreover, and although not illustrated in Table (1), ten students or 30.3% of all participants received remedial exercises for at least one of the six chapters.

The data further indicate that most errors occurred with direct objects (21.5%) and subject-verb agreement (20.6%). However, these were the most frequent constructions contained in the 120 exercises of this study. For instance, only chapters 5 and 6 (40 exercises in total) focus on the present perfect and modals. These constructions are not contained in any of the previous chapters, thus there is less opportunity for errors with these grammar topics than, for example, subject-verb agreement.

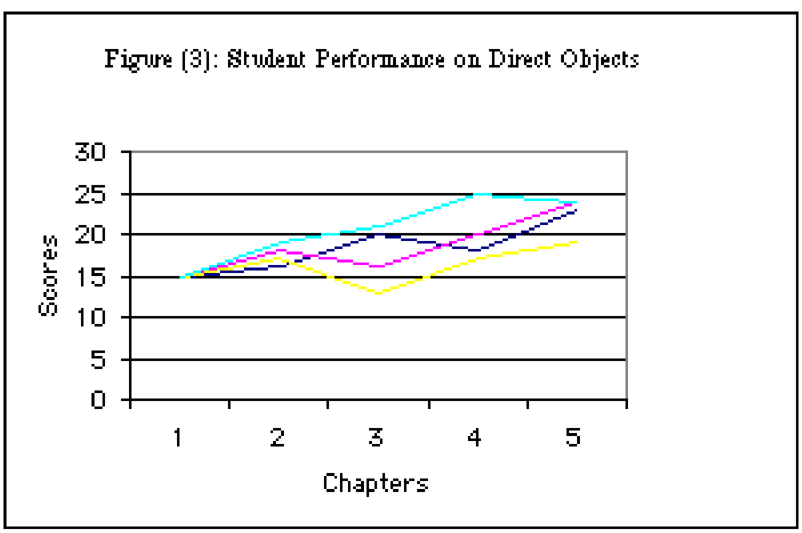
	Beginner	Intermediate	Advanced	Total	%
Direct Objects (gender, number, case)	64	226	1	291	21.5%
Subject-Verb Agreement (person, number)	27	188	63	278	20.6%
Prepositional Phrases: Dative (gender, number, case)	48	185	1	234	17.3%
Indirect Objects (gender, number, case)	42	97	7	146	10.8%
Subjects (gender, number, case)	3	82	43	128	9.5%
Missing Words	17	37	12	66	4.9%
Prepositional Phrases: Two-way (gender, number, case)	21	47		68	5.0%
Prepositional Phrases: Accusative (gender, number, case)	15	39		54	4.0%
Extra Words	11	19	11	41	3.0%
Word Order	10	16	10	36	2.7%
Auxiliaries (*to have*vs. *to be*)	1	6		7	0.5%
Verb complements (infinitive vs. past participle)	1	2		3	0.2%
	260 (19.2%)	944 (69.8 %)	148 (11%)	1352	100%

Table 1: Break-down of Grammar Errors

It is interesting, however, to consider the number of grammar errors made by each learner level. From a learning perspective, the data in Table (1) indicate three distinct groups:

1. those grammar topics where the error distribution for the beginner and advanced levels is fairly balanced (missing words, extra words, word order, auxiliaries, verb complements),
2. those grammar points where students are far more often at the beginner than the advanced level (direct and indirect objects, accusative, dative, two-way prepositions), and
3. those grammatical constructions where students are far more often at the advanced than the beginner level (subject-verb agreement, subjects).

The data of our Student Model also allow us to gain insight into students' performance on a particular grammar skill over time. For example, Figure (3) illustrates the performance on direct



objects by four students who were randomly selected from our data set.

The x-axis displays the five chapters that contain direct object constructions and the y-axis shows the scores which correspond to the three learner levels. The graphs indicate that one of the students stayed at the intermediate level throughout practice. In contrast, the remaining three students shifted from the intermediate to the advanced level. The data confirm that while there is variation across learners each student also changes performance levels as s/he progresses through the course.

The data support the need for an individualized system which makes subtle distinctions between learners and error types. Our Student Model has a number of advantages. It takes into account students' past performance, and by adjusting the score value to be incremented or decremented, it is adaptable to a particular grammatical constraint in an exercise or the pedagogy of a particular instructor. For example, a language instructor might rate some errors more salient than others in a given exercise. In such an instance, the increment/decrement of some grammar constraints can be tuned to change their sensitivity.

The main strength of our Student Model, however, is that a single erroneous message will not drastically change the overall assessment of the student. The Student Model indicates precisely which grammatical violations occurred, allowing for a fine-grained assessment of student competency. In consequence, a student can be at a different level for each given grammar constraint reflecting her performance of each particular grammatical skill. This subtlety of evaluation is desirable in a language teaching environment because as the student progresses through a language course a single measure is not sufficient to capture the knowledge attained and to distinguish among learners. The Student Model aids in directing each student toward error-specific and individualized remediation.

In the following we show how the underlying idea of phrase descriptors can easily be transferred to a student model of a Web-learning environment that is not parser-based.

5. Geroline

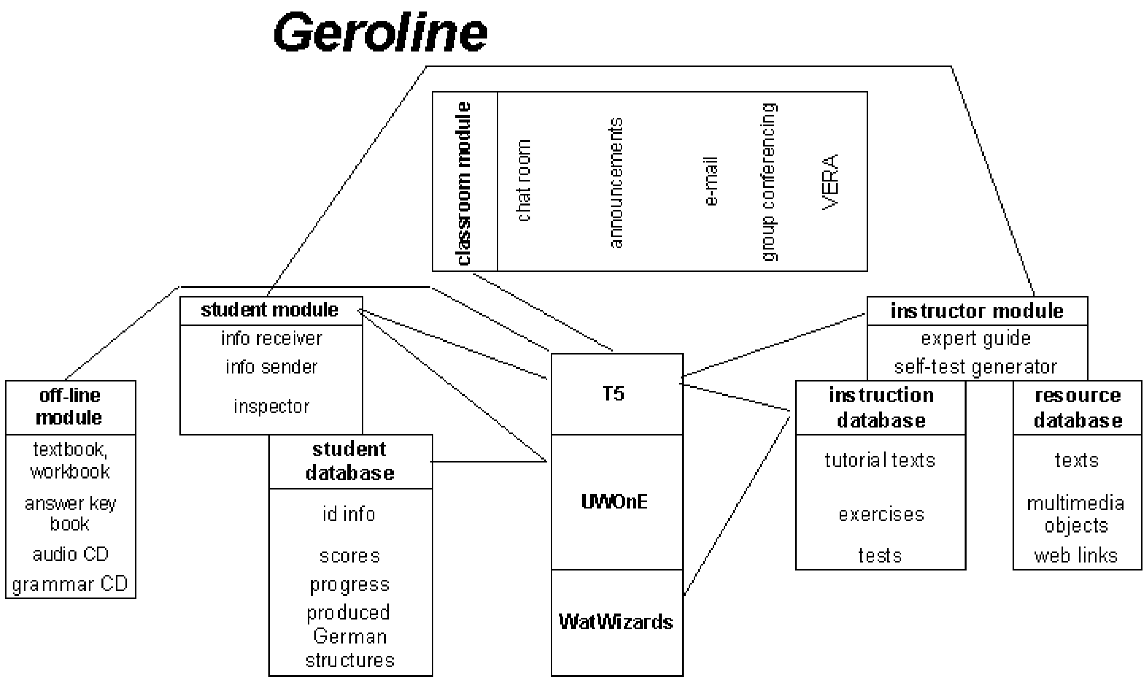


Figure 4 Geroline - General Architecture

Geroline (which is currently under development) is a distance education course for ***ab initio*** German students at university. The web-based course components (figure 4) complement the material of the textbook (Lovik et al. 2002) and and its accompanying resources (off-line module). This is done by providing students with a medium to communicate with one another and the instructor (classroom module) and by integrating interactive learning objects (tasks, tests, exercises) (Liebscher and Schulze, 2002).

The course (figure 5) relies on a task-based learning approach (Lee, 2000)**.** The materials are structured for the students in such a way that they are asked to prepare and complete certain communicative tasks. Each task comprises two self-tests, topics (learning objects to prepare task completion) and a task completion phase.

The computerized diagnostic tests (**intro test** and **self-test**) consist of a variety of sentence-based language exercises. Feedback is provided by the computer, student answers and other relevant information is stored to be able to monitor students'progress. Feedback the student receives after completing the first diagnostic self-test (intro-test) is in the form of guidance as to what resources will prove particularly useful to her or him for the successful completion of the task at hand. This 'tailored' guidance, which is provided in addition to the guidance within the T5 templates[1] which is identical for all students. We are confident that a short test at the beginning of each task is not only a useful language exercise, but it will also improve student motivation by

[1] The material is structured within the **T5 model**. This model was developed at the University of Waterloo as a template for new on-line courses. The five Ts stand for Tasks, Tutoring, Topics, Teamwork, Tools.

helping them to concentrate on the material that proves most difficult to them. Since with some students we might be testing knowledge which has not been taught yet, we are allowing them to pass all or selected questions. The second test (self-test) provides the student with information on how successfully or otherwise s/he covered the material that was provided for successful task completion. The student is advised to only attempt the task completion after having passed the second test with success.

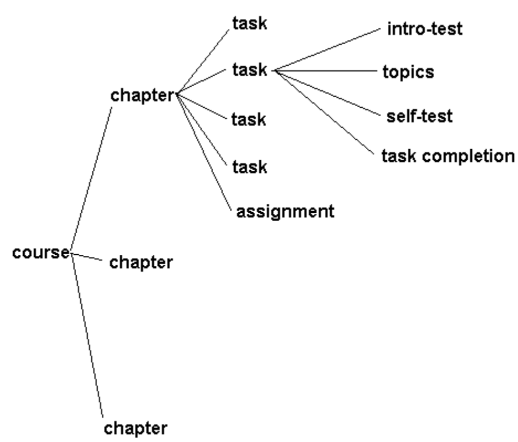


Figure 5 Geroline — Course Structure

The items in the two tests are calibrated, i.e. their difficulty (Matlock-Hetzel, 1997) is computed. The statistics are based on approaches in classical testing theory (Lord and Novick, 1968, 37) to keep computations simple. This is possible because of the relatively low risk invilved in diagnostic assessment in this course. In order to have calibrated items immediately after creation, instructors can also record an expert opinion as to the difficulty of the item. This estimate is automatically refined once more students have answered this item. Item designers can also record a maximum of three keywords per item identifying the learning domain and learning hierarchies (e.g. grammar, case, accusative). These keywords enable the system to compile dynamic adaptive tests on a given domain. These keywords are employed similarly to the phrase descriptors in the ***German Tutor***.

Consequently, ***Geroline's*** student model can rely on the following information about the learner:

- which tasks, exercises, items have been done by the student and in what order;
- performance of a particular student in a particular test or in a particular learning domain.

Depending on performance in relevant tests and domains, the student model will guide the learner through the wealth of material the course provides by pointing out low performance areas to the student and recommend relevant learning resources. This is necessary because all students are given access to the same learning objects.

After a pilot run of this course in Fall 2002 and a thorough evaluation of the test items as well as the recording mechanisms, the items are used for adaptive testing. Adaptive testing relies on the assumption that students' ability can be estimated based on the difficulty of items they can answer. Instead of showing all students the same set of questions, the student gets a more difficult question if he or she answered an item correctly and an easier one if they were wrong.

This is based on the assumption that if a student can answer an item of difficulty X, then it is unlikely that we are going to obtain any new information on the student's ability by presenting him or her with an item of difficulty X-Y.

Due to the low risk involved with diagnostic testing in such a course, it is sufficient to have a rough estimate of the students' ability and hence it is possible to say if a student is consistently presented with items of difficulty ~X in this adaptive testing approach, then the student's ability is ~(100%-X) (see formula for ability calculation at test termination below). For a different approach to adaptive testing in an ITS see Collins et al. (1996).

For the student model to present the learner with the right kind and number of items the 'pace' of moving a testee through the items of varying difficulty as well as the conditions for terminating the test have to be determined. We are starting off with the following assumptions:

- if a student consistently answer items correctly we would want to increase the difficulty more quickly; and vice versa, if a student consistently fails to answer items we would want to decrease the difficulty more quickly (see calculation of the progression value below);
- if a student moves forward and backward in a small band of difficulties, we can be sufficiently certain that this difficulty corresponds to his ability (see termination condition I below)
- if a student has failed to answer many of the easiest items or has answered many of the most difficult ones then we can terminate the test (see termination condition II below)
- if after a certain high number of questions the system is unable to establish the student's ability then it is unlikely this is going to be possible by asking the student even more questions because by then fatigue and lack of concentration are going to skew the results even further (see termination condition III below)

The following formulae and conditions are the result of a test simulation with various hypothesized value and formulae. They are to be tested on empirical data from the pilot study.

answers per item	N_i	item difficulty in %	$D_i = A_i / N_i$
answers per student	N_S	progression value (two correct or incorrect answers in a row)	${}_{N}I_S = {}_{(N-1)}I_S + sqrt({}_{(N-1)}I_S)$
correct answers per item	A_i	progression value (one correct one incorrect answer in a row)	${}_{N}I_S = {}_{(N-1)}I_S - sqrt({}_{(N-1)}I_S)$
initial progression value	${}_{0}I_S=5\%$	item difficulty of next item shown to student	${}_{(N+1)}D_i = {}_{N}D_i + {}_{N}I_S$ and 100% ? ${}_{(N+1)}D_i \leq 0\%$
initial difficulty value	${}_{0}D_{iS}=50\%$	termination condition I	${}_{(N-2)}I_S + {}_{(N-1)}I_S + {}_{N}I_S \leq 4$
		termination condition II	100% ? ${}_{(N-2)}D_{Si} + {}_{(N-1)}D_{Si} + {}_{N}D_{Si} \leq 30\%$
		termination condition III	$N_{Si} = 20$ (impossible to determine student ability)
		student ability at termination (I and II)	$P_S = 100\% - D_i$

Geroline's student model will record all the test data for all students, choose test items according to the algorithm sketched above, and provide feedback in the form of learning recommendations after calculating the student's ability on the relevant learning/testing domain.

6. Conclusion

In this paper we provided an overview of student models that have been employed in CALL over the past decade. We introduced our ***German Tutor***, an ILTS that provides error-specific and individualized feedback. The Student Model of the ***German Tutor*** is based on learner performance history and makes system decisions accordingly.

A study in which we evaluated the extent to which our Student Model addresses the need for an individualized language learning environment emphasizes the importance of an adaptive language learning system that considers user diversity. Approximately one third, or 30%, of the time, students either required more elaborate feedback suited to the beginner learner, or, in the case of the advanced learner, less detailed feedback was sufficient to correct the errors. The data further confirm that while there is variation across learners individual students also change performance levels as they progress through a course.

Our study also provided some interesting insights into the error typology of different learner levels. Due to the constrained environment of the exercises of our system where students select from a given pool of vocabulary and grammatical structures, errors in omission, insertion and word order were less frequent than other grammar errors. Fewer errors occurred overall and thus the error distribution with respect to beginner and advanced levels was fairly balanced.

Finally, we described ***Geroline's*** student model for diagnostic language testing and showed that such a model does not require a parser and a grammar to provide an individualized language learning guidance. The conceptual design of phrase descriptors are easily transferable to a nonparser based CALL system and can be used to augment the information on language testing items.

References

Bailin, A., 1990. Skills in context and student modeling. In: CALICO, 8, pp. 7-22.

Bailin, A., and Thomson, P., 1988. The use of natural language processing in computer-assisted language instruction. In: Computers and the Humanities, 22, pp. 99-110.

Baker, D., 1989. Language Testing. A Critical Survey and Practical Guide. London, Edward Arnold.

Bull, S., 1994. Student modelling for second language acquisition. In: Computers and Education, 23 (1-2), pp.13-20.

Catt, M., and Hirst, G., 1990. An intelligent CALI system for grammatical error diagnosis. In: C.A.L.L., 3, pp. 3-27.

Collins, J. A., Greer, J. E. and Huang, S. X., 1996. Adaptive Assessment Using Granularity Hierarchies and Bayesian Networks. In: Frasson, C., Gauthier, G. and Lesgold, A. (Eds.). Intelligent Tutoring Systems. Third International Conference, ITS'96, Montréal, Canada, June 1996, Proceedings. Berlin, Springer-Verlag, pp. 569-577.

Cobb, T., and Stevens, V., 1996. A principled consideration of computers and reading in a second language. In: Pennington, M. C. (Ed.). The Power of CALL. Houston, Athelstan Publications, pp. 115-137.

Elsom-Cook, M., 1993. Student modelling in intelligent tutoring systems. In: AI Review, 7 (3-4), pp. 227-240.

Greer, J. E. and McCalla, G. I., 1994. Student Modelling: The Key to Individualized Knowledge-Based Instruction. Berlin, Springer-Verlag.

Heift, T. & Nicholson, D., 2001. Web delivery of adaptive and interactive language tutoring. In: International Journal of Artificial Intelligence in Education, 12, in press. Also available online at: http://cbl.leeds.ac.uk/ijaied/abstracts/Vol_12/heift.html (May 1, 2002).

Holland, M. V., Kaplan, J.D., and Sama, M.R. (Eds.), 1995. Intelligent Language Tutors: Theory Shaping Technology. Mahwah, NJ, Lawrence Erlbaum Associates, Inc.

Holt, P., Dubs, S., Jones, M. and Greer, J., 1994. The state of student modelling. In: Greer, J. E. and McCalla, G, (Eds.) Student Modelling: The Key to Individualized Knowledge-Based Instruction. Berlin, Springer Verlag, pp. 39-35.

Holland, M. V., 1994. Lessons learned in designing intelligent CALL: Managing communication across disciplines. In: C.A.L.L., 7 (3), pp. 227-256.

Kurup, M., Greer, J. E., and McCalla, G., 1992. The Fawlty Article Tutor. In: Frasson, C., Gauthier, G., and McCalla G.I. (Eds.). Intelligent Tutoring Systems. Berlin, Springer Verlag, pp. 84-91.

Lee, J.F., 2000. Tasks and Communicating in Language Classrooms. Boston, McGraw Hill.

Liebscher, G. and Schulze, M., 2002. Challenges in designing an online environment for beginners' German courses. In: Forum Deutsch (in press).

Lord, F. M. and Novick, M. R., 1968, Statistical Theories of Mental Test Scores. With a Contribution by Allan Birnbaum. Reading, Mass., Addison-Wesley Publishing Company.

Lovik, T. A., Guy, J. D. and Chavez, M., 2002. Vorsprung. Boston, Houghton Mifflin.

McCalla, G., 1992. The centrality of student modelling to intelligent tutoring. In: Costa, E., (ed.). New Directions for Intelligent Tutoring Systems. Berlin, Springer Verlag, pp. 107-131.

McCalla, G. and Greer, J., 1992. Special issue on student modelling: Editors' introduction. In: Journal of Artificial Intelligence in Education, 3 (4), pp. 377-380.

Oxford, R. L., 1995. Linking theories of learning with intelligent Computer-Assisted Language Learning (CALL). In: Holland, M. V., Kaplan, J.D. and Sama, M.R. (Eds.). Intelligent Language Tutors: Theory Shaping Technology. Mahwah, NJ, Lawrence Erlbaum Associates, Inc., pp. 359-369.

Virvou, M., Maras, D. and Tsirigia, V., 2000. Student modelling in an intelligent tutoring system for the passive voice of English language. In: Educational Technology and Society, 3 (4), http://ifets.ieee.org/periodical/vol_4_2000/virvou.html. (July 3, 2002)