EMERGING TECHNOLOGIES
TOOLS AND TRENDS IN SELF-PACED LANGUAGE INSTRUCTION

Robert Godwin-Jones
Virginia Commonwealth University

Ever since the PLATO system of the 1960's, CALL (computer assisted language learning) has had a major focus on providing self-paced, auto-correcting exercises for language learners to practice their skills and improve their knowledge of discrete areas of language learning. The computer has been recognized from the beginning as a patient and tireless -- if inflexible -- tutor, allowing students to practice repeatedly, completing drill exercises with the computer program providing feedback. Chunks of knowledge for which identifiable right and wrong answers can be provided enable the creation of self-correcting exercises. Unfortunately, human language is notoriously more difficult to adapt to this kind of computer-based instruction than are other areas of knowledge like mathematics. Language is so much more fluid and ambivalent than numbers and equations that basic pattern drills in language learning can accomplish only so much. Moreover, self-correcting computer drills seem inconsistent with the current model of communicative language learning based on meaningful task-based interactions with the language. Yet such exercises continue to be created and used and can still play a useful role, particularly when paired with more open-ended, communicative tools and integrated into a multimedia-rich, collaborative on-line language instruction environment.

The advent of the Web (and particularly of JavaScript) as well as of digital multimedia has provided new options and capabilities for creation of language exercises. At the same time, the Web has become the preferred platform for delivery of intelligent tutoring systems, which provide a rich context for self-paced instruction. The complexity of these systems has been a formidable obstacle to involvement of language professionals. Now authoring tools are becoming available which allow language teachers to become active participants in this process. At the same time, new initiatives in educational technology standards (3rd edition of SCORM 2004, IMS Common Cartridge) provide new opportunities for the creation and distribution of on-line language learning resources.

SELF-CORRECTING WEB EXERCISES

Most language drills used today are created for delivery through a Web browser, particularly if these are materials being created by language instructors themselves. The arrival of JavaScript (in 1996) has slowly displaced most other alternatives for creation of self-correcting exercises. In its decade of existence, JavaScript (the official standard is called ECMAScript) has steadily added features, becoming a robust, object-compliant scripting language with a full set of core functions. JavaScript now has the ability to do sophisticated pattern matching through standard "grep" (general regular expression parsing), which is important for parsing input to check for needed inflections, verb endings or word roots. Its ability to manipulate arrays gives it a rudimentary database-like functionality, important for tracking user actions, while its interaction with the elements of a Web page through the DOM (document object model), allows on the fly changes to Web page content. Developers who remember the days of HyperCard or Toolbook, however, will still miss in JavaScript some of the functionality available in those authoring environments; in particular, seamless multimedia integration and access to creating and manipulating elements of the GUI (graphic user interface) such as buttons. In contrast to those programs, however, JavaScript is thankfully independent of operating systems and largely Web browser neutral.

It would take an intrepid language teacher today to create from scratch interactive exercises using JavaScript. The existence of several easy-to-use authoring tools thankfully make that unnecessary, unless the supplied exercise templates do not fit the instructor's (and students') needs. The tools most often used by US language instructors are Quia and Hot Potatoes, both of which have been in existence for some time and have developed an ever-increasing array of exercise types. They also take advantage of the dynamic combination of JavaScript and CSS (cascading style sheets) to allow for quite sophisticated drag-and-drop type interactions. Both tools allow for integration of multimedia, although that requires some manual editing of the HTML code. The current resource list for shared exercises created with Quia demonstrates how active language instructors have been in creating exercises in that environment, with 32 languages represented, from the usual suspects to some unexpected finds, like Basque, Jersey French, and Samoan. The list for Hot Potatoes is similarly extensive. Other tools for creating interactive Web exercises include WebPractest (created by Gary Smith of the College of William and Mary) and WebAuthor (from the University of Pennsylvania).

JavaScript has become popular as a means to create authoring templates because it is text-based and need not be compiled, making it a simple process to create a means for users to plug in their own materials alongside the JavaScript, CSS, and HTML needed for interactive functionality. This is not possible using alternatives such as Java or Flash, since they are distributed as compiled programs not as text. However, it is possible to write a generic Java applet or Flash movie, which could read in data stored in an external text or XML file. This technique, common in some learning environments, seems rarely to be used in language learning. One of the advantages of using one of these two options is easier multimedia integration, especially in the case of Flash. However, the complexity of authoring in Java or Flash, particularly to create a template-based system, has likely been the main factor that has sent developers to JavaScript instead. An exception to this pattern is SMILE from Michigan State University, a program for creating Flash-based interactive language-learning exercises.

Another technique in wide usage in other areas, but not in language learning is Ajax (short for Asynchronous JavaScript and XML). The use of Ajax offers the compelling possibility of integration of server-based resources into Web-delivered activities, thus enabling real database integration including possible features such as random exercise generation, dynamic sentence generation, or persistent user tracking. One of the few language learning examples of Ajax use I have seen is a Flashcard program from LikeThought called Lexicon. The advantage of adding a server back-end capacity to JavaScript language drills through Ajax is the ability to store data on a Web server. JavaScript does not offer a good way to store significant amounts of data, as browser cookies have several limitations. The use of Ajax also overcomes the delays associated with server-based interactions as data is downloaded in the background as needed and kept in the browser memory (through JavaScript) until ready to be displayed or to be called upon in user interactions.

SELF-PACED INSTRUCTIONAL MATERIALS AND COURSES

Tools such as Quia and Hot Potatoes are used principally for the creation of learning exercises for the practice of grammar, vocabulary, or reading/listening comprehension. For the most part the exercises are targeted at language learners at the novice level. The exercises created with these tools tend to be used independently of one another. A more complete set of self-instructional materials can be found in commercial products, which offer a full range of learning exercises from novice to advanced levels. Products such as ELLIS, Tell Me More, or Rosetta Stone offer learning environments which allow for development of all language skills. These and other such programs are built around pre-planned lessons with distinct goals prescribed in a linear, guided path. This kind of self-directed study is the traditional model for CBT or computer-based training. Of course, the products mentioned above go far beyond the simple text-based interactions associated with CBT. The addition of graphics, audio, and video, as well as in some cases, advanced technologies such as voice recognition, adds a powerful dimension to the programs.

The Critical Language Series are multimedia CD-ROMs/DVDs developed at the University of Arizona for less commonly taught languages, which provide, through rich use of multimedia and self-correcting exercises, the equivalent of a self-paced beginning textbook and workbook. Each of the programs was developed in association with the National Association of Self-Instructional Language Programs (NASILP), which promotes self-directed language study in the US. The Critical Language series was built using the teacher-friendly MaxAuthor authoring system, which allows for creation of a variety of learning activities and supports 47 different languages. Programs developed in MaxAuthor can be delivered on digital media or over the Internet.

In addition to such courseware, several Web sites created by public service institutions, universities, or private companies provide extensive language learning environments, which focus on learners working their way through a set of prepared lessons. One of the best of these sites is that created by the BBC, which offers online courses in seven languages. The BBC course, as well as other sites such as Unilang, offers additional complementary features such as discussion boards. The fact that the language learning resources are on the Web allows for multi-user communication and peer interactions not usually available with software delivered on CD, DVD, or a local area network. The disadvantage of the language Web sites, however, is the lack of integration among the various exercises, tools and services provided. An exception is the online elementary French course offered by Carnegie Mellon University. The course offers a fully integrated on-line environment, with rich use of original multimedia to support the self-paced instructional program.

For many language instructors who want to create electronic learning resources for their students, the environment often used today is a learning management system (LMS) such as Blackboard or Moodle. All of these systems allow instructors to create exercises in a variety of formats. Increasingly, LMS assessment options include the possibility of importing questions and including multimedia. There are several advantages to creating exercises in an LMS, including performance tracking through an online grade book and integration into a full-featured learning environment with rich communicative tools. The assessments in an LMS are server-based and hence inherently more secure than JavaScript-based, client-side exercises. They do not, however, usually provide instant feedback to the user, nor do they generally have the flexibility present in other authoring options. Most significantly, the assessment tools available in an LMS were not designed with language learning in mind. An interesting project from the Netherlands, called ELLIPS (Electronic Language Learning Interactive Practising System), is an attempt to bridge that gap. It focuses on grammar training, listening and pronunciation drills and is designed to be used within a learning management system. It offers features not usually found in LMS assessments such as transcript-based exercises, voice recording, and audio feedback.

INTELLIGENT TUTORING SYSTEMS

Whether using a JavaScript authoring tool like Quia or Hot Potatoes, or a server-based LMS, the exercises created are generally presented separately from other content. While the exercises may be linked, for example, to reading materials, audio passages, or videos, they most often will not be fully integrated into those or other language learning resources such as on-line glossaries or communication services. They generally also present a single, one-size-fits-all path to the goals targeted for a particular set of exercises. A much more differentiated and individualized approach is possible through the use of an intelligent tutoring system (ITS). Such systems, more common in fields other than language learning, offer multiple pathways through the learning content, allowing for remediation, repetition, or skipping ahead based on user performance. ITS are dynamic, responding on-the-fly to user actions and decisions by changing the anticipated learning path, or assembling customized learning content for presentation. ITS makes use of artificial intelligence (AI) to drive the program. The reaction to user actions are not all spelled out in advance but are processed as the user progresses through the program and can allow for much more varied and personalized interaction with the user. Such systems often take advantage of natural language processing (NLP) to interpret user input as well as to generate responses. Some ITS use speech recognition and synthesized speech to enter into conversational exchange with the user.

Such systems have an immediately evident appeal and have been a goal in language learning technology for some time, often referred to as ICALL or intelligent computer-assisted language learning. Most of the work that has been done in this area has been either theoretical or very narrowly focused. There are examples, e.g., for teaching the passive voice in English or for Marathi, a language spoken in India. The difficulty in creating an ITS for language instruction lies not only in the process itself, which is quite complex, but also in the difficulty of creating machine intelligence around something as slippery as words and sentences. The first step in building an ITS is to create a content model or domain knowledge for the particular content area targeted for learning. This is a hierarchical construct, which represents expert knowledge in the field. Clearly, if the targeted area is shipboard damage control, this is much more manageable a task than if the goal is some aspect of human language use. Defined models of language instruction consistent with the current state of knowledge in second language acquisition are needed. ITS is widely deployed in fields such as engineering/computer science education, math/science, or management and related areas. It tends to be used often in the military and in some corporate environments, more than in education. In recent years, at least in the U.S., the government has funded projects for creation of quite extensive independent language learning systems. Alelo, for example, a company spun off from the University of Southern California, has developed quite sophisticated programs for teaching language and culture to the US Armed Forces, combining aspects of ITS with advanced gaming.

ITS is most often used for narrowly defined areas of technical training. ITS holds considerable promise for those aspects of language learning, which deals with discrete areas of knowledge: grammar, vocabulary development, comprehension checking, specific cultural knowledge. The value for the language learner increases if an ITS is used in an online environment which takes advantage of Internet resources and communicative possibilities. A step in that direction is the REAP project, a reading tutorial system. The system keeps track of students' vocabulary learning lists and individually tailors instructional materials for the vocabulary the student is working to learn. One way it achieves that goal is through automatic retrieval, display, and annotation of reading selections from the Internet. The system searches the Web for examples of the targeted item's use in context and presents the texts, along with comprehension aids, to the student. This is clearly a difficult task, given the proliferation of texts on the Web and the challenge of finding texts at the student's reading level and in an appropriate context (not a list, ad, timetable, etc.) and requires the use of quite sophisticated and intelligent parsing tools.

The complexity of creating an ITS has resulted in work in this area being done principally by specialists in computer sciences, rather than by experts in second language acquisition or related fields. There are efforts to bridge the gap, as is the case in the projects out of the ICALL research group based at the Dublin City University. The group is especially interested in leveraging existing NLP tools in CALL applications for specific groups of language learners with particular needs or backgrounds. One project of the group is a plurilingual ICALL system for three Romance languages that includes a plurilingual parser, animated grammar presentations, and the use of specialized corpora. It assumes the learner already is an advanced-level speaker of one of the languages and leverages that knowledge for learning a related language. Another interesting project involves the use of NLP tools to build an artificial "co-learner". The idea is that the advanced learner can "teach" the artificial co-learner when it makes errors, and in the process improve both the human's and computer's knowledge of the target language. The participation of actual language learners in the shaping of language learning resources would be of significant interest, as it has long been recognized how powerful it is for learners to be active participants in the instructional process.

It is an unlikely proposition that mainstream language learning professionals will become the principal drivers behind creation of an ITS project for language learning. Here, too, there have been efforts recently, however, to bring subject matter experts and teachers into the process of ITS creation. A compelling development in this area comes out of work done at Carnegie-Mellon University (CMU). A number of ITS programs have been built using a Cognitive Tutor Authoring Tool (CTAT) developed there. The Algebra Cognitive Tutor, for example, is one of the most widely used ITS programs in math education. CTAT tools enable subject matter experts to develop electronic tutors with some level of AI. The tools allows for creation of two different types of tutors, the more basic example being a tracing tutor (also called pseudo-tutor) which can be created without programming but requires problem-specific authoring, and the other being cognitive tutors, which require AI programming to build a cognitive model of student problem solving. The pseudo-tutor can be used for rapid proto-typing and can serve as the basis for creation of a cognitive tutor if sufficient examples allow for generalized rules to be developed and applied. Examples of these tutors, such as the politeness tutor or sentence jumbles, can be experienced on-line. The feedback can vary from basic and formulaic to complex and adaptive, depending on the type of tutor used. The on-line elementary French course at CMU makes extensive use of cognitive tutors (mostly pseudo-tutors), as do other on-line courses in the CMU Open Learning Initiative.

Related to intelligent tutoring system are what are often called adaptive systems, which generally have more flexibility and individualization than an ITS. In an adaptive system, the learner has choices, both in deciding on goals to be reached, and on the preferred path, so that a personal learning style and context can be taken into consideration in the delivery of content. In adaptive systems different learners' situations can influence progress towards outcomes, and this information is encoded into the program, altering the flow and presentation of materials to the learner. The CAPE visual authoring system, out of Vanderbilt University, is based on this model. A demonstration of CAPE in the form of a Socratic tutor dynamically adapts to user learning options based on responses to questions. CAPE can be used to author online learning content for what is referred to as "anchored instruction", in which the emphasis is placed on working with real problems and situations. This clearly offers an interesting opportunity for task-oriented language learning. Such a system could benefit from the work Lindy Woodrow has done on models for adaptive language learning.

SCORM 2004 AND THE IMS COMMON CARTRIDGE

One of the approaches to creating a basic version of an ITS is to use the sequencing system built into the SCORM 2004 standard (see LLT volume 8). Advanced Distributed Learning (ADL) has recently released the third (and final) edition of SCORM 2004. They have also announced that they would be turning over the responsibility for the SCORM standard to international standards-setting groups. The principal new feature in SCORM 2004 is the addition of "simple sequencing" from IMS (IMS Global Learning Consortium). This allows for creation of content with a non-linear path, such as courseware with complex branching options. In some ways creation of content using the IMS sequencing system is similar to the steps in creating an ITS. The content model is mirrored by the Content Organization spelled out in the (XML) manifest file, which is the central document of a SCORM package. Since SCORM content is designed to be used within an LMS, the Content Organization in the manifest file is parsed by the LMS and becomes the "Activity Tree" for that content. Each of the items in the Activity Tree corresponds to a learning activity (such as a question or a series of questions). Rules for sequencing can be attached to each activity, i.e., to indicate what happens given different responses to activities. Activities can be associated with one or more Learning Objectives. Depending on responses to learning activities, learning objects may be satisfied or not, leading to different branching options.

In an ITS there is also a "student model", which spells out what the learner knows and progress made toward achieving the learning goals. In SCORM 2004, this is represented by the Activity State and Tracking Models (spelled out in the manifest file), which follow, record, and assess students' traversal of the learning content. The final component of an ITS is the instructional model, which defines the order in which content is presented, how errors are analyzed and responded to, and typical mistakes and recourses, as well as provides help and remediation options. In SCORM 2004 this is provided by the Sequencing Definition Model, which like everything else, is designated in the structure of the manifest file. The sequencing model in SCORM 2004 provides for considerable flexibility in how content is presented and how the system interacts with the user. Fundamentally, the model is one of content mastery, established by the learning objectives, and assessed through questions (learning activities). In this sense, language skills such as grammar knowledge or reading comprehension could lend themselves quite well to being developed into SCORM-compatible lessons. Such lesson could be set up to provide choices to learners in how to work with the material; for example, the options to chose a guided flow in which a learning path is pre-determined or an exploratory mode in which the user is provided more choices. This would move in the direction of an adaptive learning environment.

One of the goals of SCORM is that SCORM-compatible content be portable to different learning systems and be able to be combined in different ways depending on the needs of the user. For this reason, SCORM learning objects (called SCOs for Sharable Content Object) are designed to be small units, with flexibility to be used in different contexts. This, however, goes counter to the highly contextualized environment of an ITS. There is a trade-off between high content and low reusability or high reusability (i.e. small units) and low context. The nature of a given project and its projected use will determine what an appropriate level of granularity will be for the content. The importance of adding SCORM compatibility to learning objects is highlighted by the Merlot project's analysis grid for reusability of resources, which emphasizes the use of SCORM. A key advantage of using SCORM is the availability of authoring tools, which allow individual users to create SCORM-compatible content. There are both free products such as Reload, Xerte and eXe as well as commercial products such as ReadyGo Web Course Builder, Course Genie (from Wimba Horizon) and the SoftChalk LessonBuilder. A future version of Hot Potatoes promises SCORM integration. It is easier to build template-based authoring tools for SCORM than for a traditional ITS, since the logic of the earning path and learner interactions are contained on one highly-structure text file (the manifest) rather than being embedded in compiled programs.

The SCORM standard is one of the principal constituent parts of the newly emerging standard for content exchangeability, the IMS Common Cartridge, for which a preliminary specification has recently been announced. The standard combines SCORM with a meta-data standard (LOM), the IMS Questions and Test Interoperability standard, and the recent IMS Tools Interoperability Guidelines. This last specification allows integration into SCORM type content of different (server-based) learning tools and services. This is a key element of the Common Cartridge, as it allows for tools outside the LMS to be brought into an on-line course and fully integrated, in the same or similar ways in which third-party tools are now integrated (such as the Blackboard "building blocks"). This holds the promise of opening up, somewhat, closed systems and the prospect of customized tools being developed that could be deployed across a range of LMS's. At the most recent alt-i meeting (the annual IMS demo session), the Common Cartridge was shown being used to port content into a variety of systems, including the open source SAKAI and commercial products such as Blackboard. There have been efforts in the past to enable this kind of interchangeability of structured learning content, which have been unsuccessful largely due to insistence on the part of commercial vendors to keep proprietary implementations. Backboard's recent patenting of key aspects of on-line learning environments, as well as its acquisition of main rival WebCT have resulted in fears that the company could force out rivals and maintain a near-monopolistic hold on learning systems. Open-source LMS alternatives, however, most notably Moodle and SAKAI, have garnered considerable support in recent years, and Blackboard has signed on to the Common Cartridge initiative and has been an active participant in development and testing.

RESOURCE LIST

Web Language Exercise Tools and Programs

- Some Sites built with Half-Baked Software tools List of language exercises using Hot Potatoes

- Quia Popular Exercise generator
- SMILE For creating Flash-based interactive language-learning exercises
- Developing Language Learning Tools By Dennie Hoopingarner
- WebPractest From Gary Smith
- DynEd English Language Learning Program
- Language Learning Arabic Deluxe Version From logoi.com
- French Online Online French class using cognitive tutors
- Self-Directed Language Learning From CAL Resource Guides Online
- The Impact Of Self-Instructional Technology On Language Learning: A View Of Nasilp By Alexander Dunkel, Scott Brill, & Bryan Koh (PDF)
- NASILP The National Association of Self-Instructional Language Programs
- MaxAuthor Multimedia authoring system for language instruction developed at the University of Arizona
- Critical Language Series Multimedia CD-ROMs and DVD-ROMs for less commonly taught languages
- ELLIPS Language tools from the Netherlands

Intelligent Tutoring Systems and SCORM

- Adaptive Learning Designs Article by Adriana J. Berlanga and Francisco J. García; deals with IMS LD
- Language Learning: Challenges for Intelligent Tutoring Systems By Michael Heilman and Maxine Eskenazi (PDF)
- Using language resources in an intelligent tutoring system for French By Chadia Moghrabi
- Intelligent Tutoring From ADL
- Intelligent Tutoring Systems From the Association for the Advancement of Artificial Intelligence
- ITSPOKE: An Intelligent Tutoring Spoken Dialogue System By Diane Litman
- The Nihongo Tutorial System An Intelligent Tutoring System for Technical Japanese Language Instruction By Anthony A. Maciejewski and Nelson K. Leung
- Expert Tutoring and Natural Language Feedback in Intelligent Tutoring Systems By Xin LU
- Web Passive Voice Tutor: an Intelligent Computer Assisted Language Learning System Over The WWW By Maria Virvou and Victoria Tsiriga
- PRABODH: An Intelligent Tutor for Teaching Language Skills to Young Children By Rekha Govil and Madhavi Saxena
- Supporting Intelligent Tutoring in CALL by Modeling the User's Grammar By Lisa Michaud and Kathleen McCoy
- Student Modelling in an Intelligent Tutoring System for the Passive Voice of English Language By Maria Virvou
- An Intelligent Tutor for Classical Physics by Kay G. Schulze
- A conceptual framework for Web-based intelligent learning environments using SCORM-2004 By Sabbir Ahmed Kazi (PDF)
- SCORM 2004 Sequencing & Navigation Special issue of "Learning Technology"
- Intelligent Tutoring System Capabilities Using SCORM Sequencing and Navigation On converting an ITS to SCORM 2004 (Microsoft Word file)

- SCORM 2004 Primer: A (Mostly) Painless Introduction to SCORM By Gord Mackenzie
- IF2006 REUSABLE DESIGN REVIEW AND RATING FORM Form used by Merlot to check resources for re-usability
- A SCORM-based English Remedying Learning System for Pre-Junior High School Students By J.-M. Hsu
- Using Learning Objects to Design Self-Paced Online Language Courses By Delphine Reni__ (Powerpoint)
- Self-paced Language Courses From the Unviersity of Wisconsin
- Patents and IMS Common Cartridge By Michael Feldstein
- Common cartridge - the future or five years too late? By Sheila Macneill
- Common Cartridge IMS information
- AutoTutor Advanced ITS that uses an animated agent
- Cognitive Tutor Authoring Tools From Carnegie Mellon University
- Politeness Tutor An example of use of the Cognitive Tutor
- Sentence jumbles An example of use of the Cognitive Tutor
- Advanced Distributed Learning Home of the SCORM standard