ELSA: An Internet Agent That Adapts World Wide Web Pages to Users’ Learning Styles

by

Chris Teplovs

A thesis submitted in conformity with the requirements for the degree of Master of Arts
Department of Curriculum, Teaching and Learning
Ontario Institute for Studies in Education of the University of Toronto

© Copyright by Chris Teplovs 1997
The author has granted a non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of this thesis in microform, paper or electronic formats.

The author retains ownership of the copyright in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author’s permission.

L’auteur a accordé une licence non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de cette thèse sous la forme de microfiche/film, de reproduction sur papier ou sur format électronique.

L’auteur conserve la propriété du droit d’auteur qui protège cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

0-612-28723-8
Abstract

Title: ELSA: An Internet Agent That Adapts World Wide Web Pages to Users' Learning Styles

Degree: Master of Arts Year: 1997

Name: Chris Teplov

Department: Education

University: University of Toronto

The Experimental Learning Styles Agent (ELSA) dynamically parses specially tagged HTML-like documents and renders them differently based on individual users' learning styles. Normally, these documents are parts of a self-contained learning module.

The agent is programmed using the perl programming language and employs an SQL-based back-end for access to user profiles. Appropriately tagged pages (i.e. those with <ELSA> tags) are dynamically parsed so the resulting page is customized for a particular user's learning style as stored in the database. As the user moves through a learning module on the World Wide Web (WWW), her user profile is updated based on her choices and subsequently requested pages are parsed according to the updated information.

The initial profile of a user is created using a WWW-based form that presents the Inventory of Learning Processes. Each user's profile comprises scores on four scales. The form uses JavaScript to check for valid data.
Acknowledgments

I would like to thank Prof. Bob McLean for his support and encouragement, and Prof. Ann Zimmerman for providing the flexibility required for me to conduct this research.

I would like to extend my sincerest thanks to my wife Esther Somogyi, whose love, support, and encouragement allowed me to complete my research.

Portions of Appendixes 1 and 2 are reprinted with permission from R.R. Schmeck.
# Table of Contents

Abstract ........................................................................................................ ii

Acknowledgments ....................................................................................... iii

1. Introduction ............................................................................................... 1

2. Literature Review ..................................................................................... 2

   2.1. Individual Differences .......................................................................... 2

      2.1.1. Learning Styles ............................................................................. 3

      2.1.2. Locus of Control .......................................................................... 6

   2.2. Aptitude-Treatment Interactions (ATIs) .............................................. 6

   2.3. Learning Styles and Computer-Assisted Instruction .......................... 10

   2.4. The World Wide Web ......................................................................... 12

   2.5. Agents ............................................................................................... 13

3. Design ....................................................................................................... 14

4. Implementation .......................................................................................... 18

   4.1. Hardware .......................................................................................... 18

   4.2. Support Software ............................................................................ 18
4.3. The ELSA Tag ................................................................. 20
4.4. The Inventory of Learning Processes Online (ILP-O) .................. 21
4.5. ELSA ................................................................. 22
5. Demonstration ................................................................. 24
5.1. The Inventory of Learning Processes Online (ILP-O) ............... 24
5.2. ELSA ................................................................. 25
6. Discussion ................................................................. 35
6.1. The ILP Online ................................................................. 35
6.2. ELSA as an Agent ................................................................. 35
6.3. ELSA as a Test-Bed for ATI Research ............................................. 36
6.4. Balancing Micro- and Macroadaptation ............................................. 36
6.5. Extensions to ELSA ................................................................. 40
7. Conclusions ................................................................. 43
8. References ................................................................. 44
9. Appendix 1: The Inventory of Learning Processes Online (ILP-O) .... 52
10. Appendix 2: The Inventory of Learning Processes Online (ILP-O) Data File ................................................................. 61
11. Appendix 3: ELSA ................................................................. 64
1. Introduction

The ability of computers to provide individualized instruction has been perceived as one of the most promising aspects of computer-assisted instruction (CAI). To date, however, efforts to provide individualized instruction by using CAI have had little success.

I propose that an autonomous agent, whose realm is the Internet, can assist instructors in creating truly individualized instruction. This agent, named ELSA (the Experimental Learning Styles Agent), is responsible for rendering specially coded documents that are retrievable via the HyperText Transfer Protocol (HTTP) in a way that is specific to the learning style profile of the user.

The approach to individual differences in learners taken in this thesis is one of Aptitude-Treatment Interaction (ATI) research (Cronbach & Snow, 1977). ATI research postulates different learning outcomes for individuals with different aptitudes, and that these differences in learning outcomes are due at least in part to educational treatments (or approaches to teaching). Although interest in ATI research seems to have declined in recent years, ATI research may find new vigour given recent advances in network-based computer-aided instruction, for reasons presented in this thesis.

This thesis consists of this introduction, a literature review, a section explaining in detail the design of the software, a demonstration of the agent, and a discussion of the research. I end with conclusions about the utility of ELSA for ATI research. The source code for the ELSA system is presented as a series of appendixes.
2. Literature Review

Here I review and synthesize literature pertaining to the thesis. There are five sections of the literature review: (1) Individual Differences, which comprises discussions of cognitive controls, cognitive styles, personality types, learning styles, and locus of control, (2) Aptitude-Treatment Interactions, which discusses work based on Cronbach and Snow's (1977) theory of ATI, (3) Learning Styles and Computer-Assisted Instruction, which attempts to link some of the research on learning styles with research on computers in education, (4) the World Wide Web, which refers to the world-wide network of computer networks, and (5) Agents.

2.1. Individual Differences

Ayersman and von Minden (1995) provide a comprehensive review of literature pertaining to individual differences and instruction. They trace the origin of research on individual differences to Jung’s (1926) work on psychological types, Lewin’s (1935) work on individual personality differences and Dewey’s (1938) work on the role of experience in learning. Ayersman and von Minden (1995) assume that "all three of these variables are intricately interwoven to contribute to individual differences" (p.373), but they do not provide a concise definition of "individual differences". Instead, they enumerate some of the ways in which individual differences have been described in the relevant literature.

Ayersman and von Minden (1995) summarize Witkin’s (1973) definition of cognitive styles as "our typical ways of processing information" (p. 373), and point out
the similarity to Messick's (1984) notion of styles as predictable patterns of information processing that are extensions of a person's personality (which ties back to Jung's idea of personality types).

Jonassen and Grabowski (1993) define individual differences as the nexus of cognitive controls, cognitive styles, personality types, and learning styles. In this thesis I will focus on learning styles as the basis of individual differences.

2.1.1. Learning Styles

Learning styles are the results of self-reported instruments. (Jonassen & Grabowski, 1993). Alternatively, they can be defined as "general behavioral dispositions that characterize performance in mental tasks; they are intellectual personality traits" (Baron, 1985).

Various authors have demonstrated that there are significant differences in adult learning styles (Claxton & Murrell, 1987; Corno & Snow, 1986), and that these different learning styles respond best to different teaching techniques (Hunter & McCants, 1977; Dille & Mezack, 1981; Campbell 1991).

Ayersman & von Minden (1995) advocate that instruction should be crafted in such a way as to possess sufficiently overlapping and diverse modalities to appeal to a wide variety of learner styles, rather than adapting extant instruction to a diverse audience. They mention a number of categorization schemes and conclude that "the multiplicity of individual difference categories and the fact that many of these overlap on the basis of shared characteristics often make it difficult to distinguish among them" (p. 373).
Jonassen and Grabowski (1993) reviewed five instruments designed to measure individual differences in learning styles. They reviewed: Hill’s Cognitive Mapping (Hill, 1976), Kolb’s Learning Styles (Kolb, 1985), Dunn & Dunn Learning Styles (Dunn & Dunn, 1974), Grasha-Riechmann Learning Styles (Grasha & Riechmann, 1975), and Gregorc Learning Styles (Gregorc, 1984).

Sims and Sims (1995) provide a more comprehensive review of instruments, and Hickcox (1995) provides a review of eighteen different learning style inventories, making extensive reference to Curry’s (1987) review and three-layer categorization. Many of these instruments attempt to classify individuals in some sort of multidimensional space (Briggs-Myers & McCaulley, 1992; Canfield, 1980; Dunn et al., 1986; Friedman & Stritter, 1976; Goldberg, 1963; Riechmann & Grasha, 1974; Hill, 1976).

Of the eighteen learning style inventories reviewed by Hickcox (1995), the Inventory of Learning Processes, (Schmeck et al. 1977) was the most robust in terms of validity and reliability. The population of individuals on which the instrument was tested was drawn from university students. The instrument was revised (Schmeck, 1983), and although reference was made to a further revision (Schmeck et al., 1991) no published version of it is available.

The original Inventory of Learning Processes yielded four scales: Synthesis-Analysis, Study Methods, Fact Retention, and Elaborative Processing (Schmeck, Ribich, & Ramaniah, 1977). The Synthesis-Analysis scale was subsequently renamed as the Deep Processing scale, and the Study Methods scale was subsequently renamed as the Methodical Study scale (Schmeck, 1983). Intercorrelations among the scales range from
The Deep Processing scale reflects the learner's ability to critically evaluate, conceptually organize, and compare and contrast information. The Methodical Study scale reflects the degree to which learners employ systematic study techniques such as studying every day, or at the same time of day, or in the same place. The Fact Retention scale reflects the degree to which a learner processes details, with high-scoring individuals excelling on tests in which the recollection of such details is important (Schmeck, 1983).

Schmeck (1983) defines learning style as "a predisposition on the part of some students to adopt a particular learning strategy regardless of the specific demands of the learning task" (p. 233). He defines learning strategy as "a pattern of information-processing activities used to prepare for an anticipated test of memory" (p. 234). He stresses that "style is simply a predisposition to favor a particular strategy" and that that style is modifiable.

Verheij et al. (1996) noted that learning styles are often derived from questionnaires administered to learners (Schmeck, 1983; Schmeck et al. 1991; Vermunt & Van Rijswijk, 1987), and examined the relationship between learning styles and actual study strategies that involved searching for information in hypertext. They found that "surface processors" make decisions at a local level, whereas "deep processors" navigate through the material by adopting a more holistic point of view.
2.1.2. Locus of Control

Rotter's social learning theory (Rotter, 1954) introduced the concept of locus of control. Individuals with an internal locus of control tend to accept responsibility for their own actions, whereas those with an external locus of control tend to attribute their own success or failure to extrinsic factors.

Henry (1995) presented two versions of the same hypermedia-based tutorial to 36 college-aged students. The versions differed in the amount of learner control. The "low control" tutorial presented material in a highly linear fashion, whereas the "high control" tutorial presented material with many branching options available to the students. He found that although both groups benefited from the tutorials (in terms of achievement scores), both tended to choose a linear path even when many different branches were available.

2.2. Aptitude-Treatment Interactions (ATIs)

Much of the research on individual differences has focused on the area of Aptitude-Treatment Interactions (ATIs). Research on ATIs can be traced to early work by Cronbach (1957), in which he examined the relationship between "individual differences and achievement on different experimental treatments" (Jonassen & Grabowski, 1983, p. 23).

There have been two major works summarizing research on ATIs: Cronbach and Snow's (1977) handbook and Snow's (1989) review article.
Cronbach and Snow (1977) provide a detailed report of research from the 1950s through the 1970s on ATIs. They explain the theoretical and statistical foundations of ATI research. They noted that "research methods in common use are inefficient and produce misleading results" (p. 509), and they call for the study of treatments that are of longer duration and of less artificiality. They note that "it is not profitable to contrast techniques defined only by gross labels" (p. 510).

With respect to choosing an aptitude variable, Cronbach & Snow (1977) note that reliability is important, and that where possible multiple measures of a particular aptitude should be employed. They also suggest that pilot studies into correlations between outcomes and scores derived from factor analysis are needed.

Tobias (1981) notes that various computer-assisted instruction systems purport to individualize instruction, whereas in reality the systems merely individualize the rate at which the student progresses through the material rather than the method of instruction. He also suggests that adapting instruction to specific psychological processes that may be present at the start of instruction may be undermined by rapidly shifting processes. Tobias (1981) also speculates on reasons for the decline in ATI research since the publication of Cronbach & Snow's (1977) review. He suggests that methodological problems and "difficulties in replicating and extending ATI findings" (p.214).

Shute (1992) discusses some of the relationships between ATIs and Intelligent Tutoring Systems (ITSs), particularly with respect to the diagnosis of cognitive skills. She notes that the usual approach to adaptive technology has been to represent in some sort of user profile the emerging (rather than incoming) cognitive skills of the user, and
notes that more sophisticated systems will likely take into consideration both emerging and incoming aspects of users. She notes that "valid and reliable cognitive diagnoses, then, are essential to computer systems that adapt to their users' needs" (p. 15). She presents what she calls the macroadaptive approach to cognitive diagnosis as the process by which certain measures of a learner's aptitudes are taken before the learner starts the instructional program.

Shute suggests that a learning skills taxonomy that she co-developed (Kyllonen & Shute, 1989) "provides a framework for conducting systematic and controlled ATI studies that was not possible prior to the arrival of ITSs" (p. 41). Whether the instrument that was developed is truly necessary is debatable (any number of learning styles instruments could, in theory, be used), but the notion that ITSs provide a relatively controlled environment is important.

Shute (1993a) suggests that the decline in popularity of ATI research can be attributed largely to the problems of confounding variables, such as teacher personality, instructional materials, and classroom dynamics. She used a computer-based tutoring system as a means to minimize the effects of these variables. She concluded that learning was superior when the users' learning styles (derived from their exploratory behaviours) were matched to the learning environment. Her main conclusions were "that learning outcome and efficiency may be optimized by considering an individual's learning style in the assignment of person to learning environment" (p. 70). She notes, however, that it is necessary to develop some "decision rule(s)" for appropriate placement of individuals. She suggests one mechanism to determine placement by making no a priori decisions regarding placement, but rather to use information gained by the
intelligent tutoring system to make such placement decisions.

In other research, Shute (1993b) discusses what she refers to as the *macroadaptive* approach to tutoring. This approach differs from the more traditional *microadaptive* approach by having as its basis the idea of aptitude-treatment interactions. Macroadaptation refers to the overall approach to teaching or learning, and encompasses the concept of learning style. Microadaptation, in contrast, refers to the moment-by-moment decisions about how to tailor instruction to the learner's needs. These decisions are typically derived from observation of the learner's progress. Shute studied the aptitudes of working-memory capacity and general knowledge. The treatments were constrained (few problems per problem set) and extended (many problems per problem set). Shute found that there was "a significant aptitude-treatment interaction involving working memory, general knowledge, and environment" (p. 61). She also notes that the macroadaptive approach costs less than the traditional microadaptive approach in terms of time and money.

Veenman and Elshout (1995) found an ATI in their study of 99 first-year psychology students in a simulation environment. They found that subjects with low intelligence and low metacognitive skillfulness benefited from a structured environment, whereas those subjects with low intelligence and relatively high metacognitive skillfulness were impeded by the structured environment. Subjects of high intelligence were unaffected by the treatments.
2.3. Learning Styles and Computer-Assisted Instruction

The potential for individualized instruction facilitated through appropriate technology has generated much interest for educators (Bork, 1980). Nevertheless, research about multimedia and learning styles is scarce (Litchfield, 1993).

Ayersman and von Minden (1995) note that "it is nearly impossible to adapt instruction to the learner when the typical public school teacher has 30 or more students to whom instruction must be suited" (p. 380), and note that prior to hypermedia it was not possible to create such individualized instruction through the use of technology. They suggest that hypermedia allows users of different learning styles to exploit the multiplicity of path choices when navigating through a hypermedia document in such ways as to maximize their learning. Note, however, that in Henry’s (1995) study students who were expected to take advantage of a highly branched hypermedia design chose a more linear one.

Rowland and Stuessy (1988) classified elementary education majors as either holists or serialists (sensu Pask, 1976). The authors defined the holist strategy as "a global approach to learning, where the learner first builds broad descriptions and then fits in details" (p. 36), whereas a serialist strategy "uses a local approach, where the learner concentrates on narrow procedures before an overall picture emerges" (p. 36). Rowland and Stuessy (1988) then assigned each subject to one of two modes of computer-assisted instruction: tutorial or simulation. Rowland and Stuessy assumed that holists would perform better under the simulation treatment, and that serialists would perform better under the tutorial treatment. Their results suggest that individuals perform best when
their learning style is matched to the mode of instruction.

In another study, Rowland and Stuessy (1987) identified field independence and locus of control (among other factors) as important factors in determining performance on simulations.

Cordell (1991) derived the learning styles of health care employees using the 4MAT Learning Styles Inventory, which she attributes to Kolb (1985), although the 4MAT Learning Styles Inventory is more properly attributed to McCarthy (1980). Subjects were assigned to either a "branching" or a "linear" tutorial. Analysis of the posttest results showed a significant difference between the branching and the linear tutorial groups, but no significant difference between learning styles and no significant interaction between learning style and instructional design. However, she noted that "if the branching and linear programs could be more distinct in their design, results for interaction of design and learning style might be detected" (p. 180).

Liu and Reed (1995) examined the differences in the way field-dependent and field-independent learners used hypermedia. They found that learners from these two groups employed different learning strategies, and the learners chose different modes to accomplish their learning tasks. They concluded that hypermedia has the potential to match instruction to individuals' learning styles. The authors note that the use of hypermedia to match learning styles to instruction is a relatively new area of research, and that "much more research is needed" (p. 432).

Shute and Gawlick-Grendell (1994) investigated differences in the performances of students who were assigned to one of two groups: one group that learned material from
an ITS, and one that learned material using a pencil-and-paper workbook. They found that high-aptitude subjects who were assigned to the ITS group performed better on outcome tests than did high-aptitude subjects assigned to the pencil-and-paper group. The same was not true for low-aptitude subjects.

2.4. The World Wide Web

The World Wide Web (WWW) is the collection of documents, images, sound bites, and other media as well as the standard protocols for exchanging that information between computers (Boutell, 1996). The WWW is a hypermedia system, in which text or images can be defined as links to other documents. The WWW utilizes the Internet as its medium for the transmission of this information. The Internet can be defined as the global "network of networks", which provides a number of Internet services using a relatively small number of well-defined protocols. Examples of these protocols include HTTP (HyperText Transfer Protocol), FTP (File Transfer Protocol), and SMTP (Simple Mail Transfer Protocol).

Access to the WWW is gained through a client program (also called a browser) such as Netscape Navigator, Mosaic, Microsoft Internet Explorer or Lynx. These client programs access information that is provided by a remote server (the http server, or http daemon).

Neilson et al. (1996) note that current uses of the WWW seem to focus more on the expositional use of the WWW (i.e. the delivery of information) rather than on the potential for highly interactive programs. They note that "the Common Gateway
Interface vastly increases the range of objects that can be potentially communicated over the [WWW]" (p. 114). The Common Gateway Interface (CGI) is "a standard for interfacing external applications with information servers, such as HTTP or Web servers" (NCSA, 1995). It allows one to create dynamic web content, as opposed to static pages. Some examples of CGI processes include search engines, online quizzes, and online surveys. User input is facilitated through the use of "forms", which refers to the standardized way of representing various field types (e.g. text fields, check boxes, pop-up menus) that can be rendered by a WWW browser.

2.5. Agents

According to Foner (1993), an agent is executable code that performs any number of predefined tasks. Autonomous software agents complete their tasks without continued input from users. Adaptive autonomous software agents change the nature of the tasks they perform in response to input from their environment. An adaptive autonomous Internet agent is an adaptive autonomous agent that interacts with users via the Internet.

The term "agent" is preferred over "intelligent CAI system" (ICAI) for several reasons. To date, most ICAI systems have been designed around a specific learning module (although see Diessel et al. (1994) for a more theoretical approach to ICAI). ELSA, in contrast, is not designed around a particular learning module. Instead, ELSA is based on principles and specific parameters about the module content. Thus, the "intelligence" is provided by the individual who codes the ELSA tags, and ELSA merely acts as an agent on that individual's behalf.
3. Design

Given the existence of individual differences, some of which are manifest as differences in learning style, my goal is to construct an autonomous agent that will render web pages in a manner specific to a user’s learning style. Under normal conditions, these web pages would be parts of a learning module. By adapting the content of a learning module to a user’s learning style, I expect to see gains in learning outcome. Moreover, the agent will be able to track changes in apparent learning style over time.

This software will provide a test bed for ATI research, as well as a functional system for the delivery of instructional modules that are adapted to individuals’ specific learning styles.

To characterize a particular user’s learning style a variety of learning style inventories (of which there are many; see Hickcox, 1995) were evaluated. I chose the Inventory of Learning Processes developed by Schmeck et al. (1977) for three main reasons:

(1) It was developed using undergraduate university students as the test population, which is the same population for whom ELSA is designed.

(2) Of the various psychometric instruments available to characterize learning styles, this one was rated most highly by Curry (1987) in terms of psychometric reliability and validity. The reliability and validity of this particular instrument seem to stem from the relatively large sample population used by the authors.
(3) The Inventory of Learning Processes describes an individual’s learning style as quantitative scores on four scales. Changes to the characterization of the individual’s learning style can therefore be represented as numerical changes to the values of these scores. Furthermore, the degree to which the agent is sensitive to macroadaptation relative to microadaptation can be modeled by the weighting of previous and updated ILP scale scores.

The adaptation will be accomplished by de-emphasizing those sections of a specially-encoded HTML document that are deemed to be "outside the range of a user’s learning style". The text will be de-emphasized by either (a) "greying out" text if the user is using a graphics-enabled browser (such as Netscape) or (b) enclosing the text in a text block labelled "Aside" if the user is using a non-graphics enabled browser (such as Lynx).

I assume that the process of de-emphasizing text is a reasonable way to adapt material to a user’s learning style because of previous work (Henry, 1996) that shows users tend to choose the more linear path of moving through hypermedia even when more highly branched alternatives are available. Users may require some orientation before they are comfortable with the process of de-emphasizing. Even then, users may become intrigued by the de-emphasized content. This poses little problem for the system, which is designed to adapt to the user. If the user habitually chooses the de-emphasized text, it may be that the ILP produced invalid scores for the individual, or that the coding for a particular section is invalid.
This latter point is tied to another key assumption: that materials can be encoded in such a way that deemphasizing portions of them makes sense, and that the range of ILP score configurations can be "covered" in a way that makes sense. In the first implementation of the software, it will be up to the individual who is encoding the material to ensure that the ILP score configurations are adequately covered. In future implementations, the process of ensuring adequate coverage may be assisted with other software (see Discussion).

The choice to deemphasize text rather than to hide it completely is an important one for a variety of reasons. Some of these reasons are: (a) user preferences can change over time, (b) user preferences can change according to content, and (c) the individual responsible for encoding material could do so incorrectly.

There are several models of how a document might be encoded. The following schematic example shows material that would be deemphasized for all users except those whose profile indicates that they of a particular learning style.

Some text

<tag specifying material specific to users with a particular learning style>

Some text for these users

</tag>

Some other text

In contrast, the following schematic example shows alternative forms for material that is tailored to users with different learning styles.
Some text

<tag specifying material specific to users with one learning style>

Some text for users of this learning style.
</tag>

<tag specifying material specific to users with some other learning style>

Some text for users with this other learning style.
</tag>

The author or the material can construct arbitrarily complex combinations of ILP score ranges to specify the rendering of the materials.

A further assumption is that an individual’s ILP scores are changeable in a way that is meaningful. That is, when an individual selects a hypermedia link from a deemphasized section of the materials, her scores are updated to reflect a slightly different learning style. I believe this is a reasonable way to track such changes for several reasons. Firstly, Schmeck (1983) believes that learning styles are mutable. Secondly, random changes to a user’s ILP scores should result in no change of mean score over a period of time. If the changes are non-random in nature, the mean score will shift to that of a slightly different learning style. Finally, changes to a user’s ILP scores are analogous to having the user answer one or more questions differently upon re-administration of the ILP, which is easily testable in practice.
4. Implementation

4.1. Hardware

The ELSA system was designed on a 90-MHz Intel Pentium system. The hardware consisted of the main CPU, 32 MB of RAM, an ATI Mach64 graphics card, an Adaptec 2940 PCI SCSI controller, and a 2.0 GB SCSI hard drive.

4.2. Support Software

ELSA was designed using the Linux operating system (version 2.0). Linux is a widely available, free, Unix-like operating system. An X-Windows environment (XFree86) was used in the development of the software.

Two independent pieces of software were developed in the course of this project: the ILP-Online and the ELSA agent. Together, the ILP-Online and the ELSA agent are referred to as the "ELSA system". The ELSA system relies on other background and back-end software.

The ELSA system was deployed using the Apache web server (version 1.1.1). This web server possesses several enhancements over the NCSA web server, on which it is based. These enhancements include the ability to act as a caching proxy server, and the ability to use persistent TCP connections, known as "keep-alive" connections. These two features enhanced the response time (and therefore performance) of the ELSA system. The current ELSA design is known to work with the first beta release of Apache 1.2.
Both the ILP-Online and the ELSA agent are written in perl. Perl is an interpreted language that excels at text manipulation. It is a common choice for CGI programming, and it is available for many platforms, including most Unixes, DOS, Windows, and OS/2. The perl language is similar to C, although many mundane (but tedious) tasks such as memory allocation and deallocation are handled by the perl interpreter. Since perl is interpreted (rather than compiled), it facilitates rapid prototyping and development work. Its advanced text manipulation routines are far superior to those found in C and comparable to the facilities available in a number of standard Unix utilities (such as awk, sed, and grep).

The perl interpreter can easily be extended by the addition of "perl Modules". Such modules are collections of subroutines or objects that facilitate some process. A comprehensive listing of all publicly-available routines is available via the Comprehensive Perl Archive Network (CPAN), accessible at http://www.perl.com/CPAN. Three such modules were used in this project: LWP.pm, Msql.pm, and CGI.pm.

The LWP.pm module is a perl 5 adaptation of the perl 4 libwww-perl library. This module provides an application programming interface (API) to the World Wide Web (Aas & Koster, 1996). It is designed "to provide objects and functions that allow you to write WWW clients" (Aas & Koster, 1996), although several of the objects are more widely applicable. The module contains many features, of which the following are the most germane to this project:

- provides an object-oriented interface to HTTP-style communication
supports access through proxy servers

-provides a simple access to a user agent

According to the documentation, "MiniSQL, or mSQL, is a lightweight database engine designed to provide fast access to data with low memory requirements" (Hughes, 1996). mSQL does not provide a complete SQL environment. Instead, it provide a subset of SQL that is adequate for general database update and query. Furthermore, mSQL provides an API through which other languages (e.g. perl, C) can access the back-end database engine. The Msql.pm perl module provides an interface to the mSQL functions.*

CGI.pm is designed to assist in the construction and parsing of HTML forms. It is also useful for creating other HTML tags.

4.3. The ELSA Tag

An HTML-like tag, consisting of a beginning <ELSA ... > and ending </ELSA> define blocks of HTML that are to be parsed by the ELSA agent. The tags are ignored by standard WWW browsers, and therefore pages containing ELSA tags will be rendered normally on browsers even if they are not first passed through the ELSA agent.

A specifically-delimited series of numbers are contained in the <ELSA ... > tag. These numbers define the minimum and maximum values for each ILP factor. If the user's ILP scores fall within the pairs of minima and maxima, the ELSA block will be

*The capitalization for "mSQL" (the database engine) and "Msql" (the perl interface) is correct, and is useful for distinguishing between the two.
displayed. If at least one of the user’s ILP scores falls outside the range specified by the <ELSA ... > tag, the block will be de-emphasized.

The minimum and maximum values are separated by commas, and the pairs of minima and maxima are separated from each other by semi-colons. The set of pairs are enclosed in double-quotation marks. Text within the ELSA tags can be a mix of upper and lower case. Note that <ELSA ... > ... </ELSA> tags may not be nested.

An example of an ELSA tag is <ELSA 0,1:0.5,2:-1,1:-3.3>.

4.4. The Inventory of Learning Processes Online (ILP-O)

The ilp.cgi script presents the Inventory of Learning Processes that was created by Schmeck et al. (1977) as an online form using perl. The complete, annotated listing is shown in Appendix 1. The data file for the script is listed in Appendix 2.

If the script is called without any input parameters, the 62 questions of the ILP are presented to the user. The user then responds to each of the questions by selecting either the "True" or the "False" radio-button. When the user presses the "Calculate Scores" button, a JavaScript function is called (only if the user is accessing the page with a JavaScript-enabled browser) that warns the user that resulting scores may be inaccurate if any of the questions have been left unanswered. The same script (ilp.cgi) is called once the "Calculate Scores" button is pressed, although the presence of POSTed data invokes the "processing" part of the script rather than the "form presentation" part of the script.

All information regarding the ILP (questions, factor loadings, etc.) is stored in an external data file to enhance generalizability.
The JavaScript itself is *dynamic* in that the number of questions read from the data file is substituted for \( \text{nquestions} \) when the JavaScript is assigned to the \$javascript variable (Line 41).

ILP scores are calculated as positive if the user responded "True" to a question with a positive factor loading or "False" to a question with a negative loading. Thereafter, I calculate the user's profile (z-scores) using the published means and standard deviations. The user's profile is then saved in the SQL database. The remainder of the script merely prints out the user's profile and a brief aid to its interpretation.

### 4.5. ELSA

The software responsible for retrieving web pages and rendering them according to information about the user's learning style is called nph-elsa.cgi. The complete annotated listing of the software is shown in Appendix 3.

The software was written using perl and in addition to the CGI and Msql modules, it employs the LWP module. The LWP objects facilitate (among other things) the creation of a mini-HTTP client, which can retrieve other documents via HTTP. (In fact, the objects allow one to retrieve documents via other protocols, although doing so sometimes requires additional modules to be installed.) The CGI module is imported to facilitate the printing of HTML elements within the agent. Its form-generating and form-parsing capabilities are not employed by the agent.

In the event that the user's profile is unset, the default ILP z-scores are set to 0.0 (which is the 50th percentile). The path name of the document to be parsed is derived
from the extra path info sent as part of the URL.

Since the agent may tie up http server resources while the agent retrieves material across (potentially slow) networks, the user agent uses a proxy server, which is a caching proxy server. The URL that the user requested to be parsed is stored in a variable.

The while loop searches for instances of `<ELSA ...> ... </ELSA>` in this variable. If an ELSA tag is found, the ELSA parameters and the text enclosed by the ELSA tags are stored in separate variables. The user's ILP is checked to see if it falls within the document part's ILP range. If it does, then the ELSA tags are removed. If it does not fall in the specified range, the range is displayed using the muted colour specified in the initialization section (if the user is using a graphical browser) or as an "Aside" block if the user is using a less sophisticated browser. An "aside" block is a definition list (delimited with `<DL> ... </DL>`) prefixed with the word "Aside".

In both cases, all HREFs within the ELSA block are modified by adding to the URLs they contain information about the range of ILP scores associated with the ELSA block. This modification is necessary so that ELSA can record changes to the user’s profile when she selects a link from within an ELSA block.

This loop is repeated until no more ELSA tag pairs are found in the document. Once the loop is finished, the parsed document is printed to the output stream, along with a horizontal rule and the ELSA version (so the user knows the document has been parsed). Note that the horizontal rule and ELSA version are printed after the end of the document, which may violate certain conventions regarding the placement of material after the `</BODY></HTML>` tags.
5. Demonstration

5.1. The Inventory of Learning Processes Online (ILP-O)

The Inventory of Learning Processes Online (ILP-O) consists of 62 statements to which the subject responds either "True" or "False" through the use of an online form accessed through a WWW browser such as Netscape. A sample screen dump of a Netscape Gold session running in an XFree86 (X-Windows) environment on a Linux 2.0.25 machine is shown in Figure 1. The statements that form the ILP-O are contained in the left-hand column of cells in a table. The right-hand column contains pairs of "radio buttons" (so called because they function in a similar way to the push-buttons found on older radios in that only one of the set can be selected at any one time). When first rendered for the user, none of the radio buttons is depressed. The user may select either "True" or "False" by moving the mouse pointer over the corresponding box and clicking the left mouse button. When the user does so, the radio button is selected. The user may change his or her selection simply by selecting the other radio button of the pair. The user may select the "Reset" button located at the bottom of the form to erase all responses and start anew.

Once the user is satisfied with the responses she provided, she should press the "Submit" button. Upon pressing this button, the JavaScript code is executed, as directed by the "onSubmit" statement of the form. If the user has left any of the questions unanswered, she will be shown a warning similar to the one shown in Figure 2, indicating that the results that are about to be calculated may be inaccurate because at least one question remains unanswered. The user has the option of continuing the submission by
selecting "Ok" or cancelling the submission by selecting "Cancel". If the user elects to cancel the submission, she is returned to the ILP-O form.

If all items of the ILP-O are complete, or if the user elects to continue the submission on an incomplete questionnaire, her responses will be sent to the server where they will be processed, and results similar to those shown in Figure 3 will be presented to her. The results include the user's scaled scores on each of the four ILP scales, as well as an indication of the percentile corresponding to these scaled scores. This tabular information is followed by brief interpretive statements indicating the significance of the user’s scores on each of the scales.

The user is also presented with a link to the catalog of existing ELSA-encoded modules.

5.2. ELSA

To date, only a demonstration page has been constructed. The demonstration page may be accessed via:

http://envinfo.forestry.utoronto.ca/elsa/envinfo.forestry.utoronto.ca/~chris/thesis/testelsa.html

Note that ELSA is invoked as

http://envinfo.forestry.utoronto.ca/~chris/thesis/nph-elsa.cgi/some.machine/elsaencodeddoc.html

or
Inventory of Learning Processes (ILP)


The following series of questions was developed by Schmeck et al. as a measure of behavioural learning style. Feel free to complete this form and then press "Calculate Scores" to find out your scores. The scores are valid only if you answer all questions. The data you provide are permanently stored on the host computer. You can also view the source, as well as the question data file.

1. I can easily handle questions requiring comparison of different concepts.  
   - True  - False

2. I have trouble making inferences.  
   - True  - False

3. I have trouble organizing the information that I remember.  
   - True  - False

Figure 1: Screen dump showing ILP-O questions.
56. I learn new concepts by expressing them in my own words.  
57. I daydream about things I've studied.  
58. When I study for something I devise a system for recalling it later.  
59. I know my words  
60. I know my items.  
61. I do not try to convert facts into "rules of thumb".  
62. While learning new concepts their practical applications don't usually come to my mind.

<table>
<thead>
<tr>
<th>56. I learn new concepts by expressing them in my own words.</th>
<th>True</th>
<th>False</th>
</tr>
</thead>
<tbody>
<tr>
<td>57. I daydream about things I've studied.</td>
<td>True</td>
<td>False</td>
</tr>
<tr>
<td>58. When I study for something I devise a system for recalling it later.</td>
<td>True</td>
<td>False</td>
</tr>
<tr>
<td>59. I know my words</td>
<td>True</td>
<td>False</td>
</tr>
<tr>
<td>60. I know my items.</td>
<td>True</td>
<td>False</td>
</tr>
<tr>
<td>61. I do not try to convert facts into &quot;rules of thumb&quot;.</td>
<td>True</td>
<td>False</td>
</tr>
<tr>
<td>62. While learning new concepts their practical applications don't usually come to my mind.</td>
<td>True</td>
<td>False</td>
</tr>
</tbody>
</table>

Calculate Scores  Reset

Figure 2: Screen dump showing JavaScript confirm of ILP-O.
Results

<table>
<thead>
<tr>
<th>Scale</th>
<th>Raw Score</th>
<th>Standardized Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Deep Processing</td>
<td>0</td>
<td>-2.69</td>
</tr>
<tr>
<td>Methodical Study</td>
<td>0</td>
<td>-2.36</td>
</tr>
<tr>
<td>Fact Retention</td>
<td>0</td>
<td>-2.61</td>
</tr>
<tr>
<td>Elaborative Processing</td>
<td>0</td>
<td>-3.64</td>
</tr>
</tbody>
</table>

Interpretation

You’re probably most interested in the column labelled "Standardized Scores", which represents the z–scores of the values in the "Raw Scores" column.

You should be able to figure out your approximate percentile rank with the following information:

- A z-score of 0 corresponds to a percentile rank of 50.

Figure 3: Screen dump showing ILP-O results.
with the latter containing a ScriptAliased reference (in srm.conf -- one of the Apache server configuration files). A ScriptAlias allows one to use a shortcut (or alias) for an arbitrarily lengthy URL. Any full URL less the http:// part may be substituted for some.machine....html.

To understand how ELSA works, select one of the links on the page and click on it several times. Then select the other link on the page and click on it several times. For simplicity, both links refer to the same page (i.e. they are links to the page on which they are themselves found). The HTML source for this page is shown in Figure 4.

To help understand what is happening to the agent, Table 1 shows a trace of the user’s profile as she clicks on the links on the page.

**Table 1: Trace of user’s profile accessing ELSA demonstration page**

<table>
<thead>
<tr>
<th>T</th>
<th>userid</th>
<th>ILP I</th>
<th>ILP II</th>
<th>ILP III</th>
<th>ILP IV</th>
<th>comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>test1</td>
<td>-2.69</td>
<td>-2.36</td>
<td>-2.61</td>
<td>-3.64</td>
<td>Initial profile</td>
</tr>
<tr>
<td>2</td>
<td>test1</td>
<td>-1.095</td>
<td>-0.43</td>
<td>-1.305</td>
<td>-1.07</td>
<td>after 1st link selected</td>
</tr>
<tr>
<td>3</td>
<td>test1</td>
<td>-0.2975</td>
<td>0.535</td>
<td>-0.6525</td>
<td>0.215</td>
<td>and again</td>
</tr>
<tr>
<td>4</td>
<td>test1</td>
<td>0.10125</td>
<td>1.0175</td>
<td>-0.32625</td>
<td>0.8575</td>
<td>and again</td>
</tr>
<tr>
<td>5</td>
<td>test1</td>
<td>-0.199375</td>
<td>0.63375</td>
<td>0.586875</td>
<td>1.42875</td>
<td>after 2nd link selected</td>
</tr>
<tr>
<td>6</td>
<td>test1</td>
<td>-0.3496875</td>
<td>0.441875</td>
<td>1.0434375</td>
<td>1.714375</td>
<td>and again</td>
</tr>
</tbody>
</table>

The above scenario may be recreated by submitting a blank ilp.cgi form (acknowledging that unanswered responses may result in inaccurate representation of the
This is some instructional text. It should appeal to a general audience.

This is some other text... it might appeal to one type of audience.

This is yet some other text. This target audience is quite different from the one below. This link is likely to appeal to this learning style.

This section is designed for users with a style quite different from the one above. This other link is likely to be of interest to them.

Figure 4: Contents of testelsa.html.
user's learning style), which will reset the user's profile to the 50th percentile. Thereafter, the user can access testelsa.html and follow the links as indicated. The ILP ranges of links 1 and 2 are summarized in Table 2.

**Table 2: ILP minima, maxima, and averages of links in ELSA demonstration page**

<table>
<thead>
<tr>
<th>Link</th>
<th>ILP I</th>
<th></th>
<th></th>
<th></th>
<th>ILP II</th>
<th></th>
<th></th>
<th></th>
<th>ILP III</th>
<th></th>
<th></th>
<th></th>
<th>ILP IV</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Min</td>
<td>Max</td>
<td>Avg</td>
<td></td>
<td>Min</td>
<td>Max</td>
<td>Avg</td>
<td>Min</td>
<td>Max</td>
<td>Avg</td>
<td>Min</td>
<td>Max</td>
<td>Avg</td>
<td>Min</td>
<td>Max</td>
<td>Avg</td>
</tr>
<tr>
<td>One</td>
<td>0</td>
<td>1</td>
<td>0.5</td>
<td>1</td>
<td>2</td>
<td>1.5</td>
<td>-1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>1.5</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Two</td>
<td>-1</td>
<td>0</td>
<td>-0.5</td>
<td>0</td>
<td>0.5</td>
<td>0.25</td>
<td>1</td>
<td>2</td>
<td>1.5</td>
<td>1</td>
<td>3</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

At time $T=1$, both of the "link" blocks are shown in muted colour, which indicates that the user's profile is outside the ranges of both ELSA blocks (Figure 5). As the user selects the first link, her profile becomes more like that of a user for whom the first block is designed until (at time $T=4$) the user's profile is sufficiently like that of a user for whom the first block is designed that the block is no longer shown in muted colour (Figure 6). Changes to the user's profile are calculated by averaging the average ILP scores of the ELSA block with the ILP scores in the user's profile. In the demonstration sequence, that same user then selects the second link, and her profile is updated accordingly. After selecting the second link twice, her profile has changed such that the second block is shown as the normal foreground colour, whereas the first block is shown in muted colour (Figure 7). Whereas this example is self-referencing for the purposes of demonstrating the process of changing user profiles, it is envisioned that links will normally be to other pages.
ELS A Test Document

This is some instructional text. It should appeal to a general audience.

This is some other text. It might appeal to one type of audience.

This is yet some other text. This target audience is quite different from the one below. This link is likely to appeal to this learning style

This section is designed for users with a style quite different from the one above. This other link is likely to be of interest to them.

**ELS A Version 1.2**

---

Figure 5: testelsa.html rendered by ELSA for user whose profile is outside both specified ILP ranges.
Discussion

ELSA Test Document

This is some instructional text. It should appeal to a general audience.

This is some other text... it might appeal to one type of audience.

This is yet some other text. This target audience is quite different from the one below. This link is likely to appeal to this learning style

This section is designed for users with a style quite different from the one above. This other link is likely to be of interest to them.

ELSA Version 1.2

Figure 6: testelsa.html rendered by ELSA for user whose profile is within first specified ILP range.
Discussion

ELS A Test Document

This is some instructional text. It should appeal to a general audience.

This is some other text... it might appeal to one type of audience.

This is yet some other text. This target audience is quite different from the one below. This link is likely to appeal to this learning style.

This section is designed for users with a style quite different from the one above. This other link is likely to be of interest to them.

ELS A Version 1.2

Figure 7: testelsa.html rendered by ELSA for user whose profile is within second specified ILP range.
6. Discussion

6.1. The ILP Online

The Inventory of Learning Processes - Online (ILP-O) provides a WWW-based mechanism by which the learning styles of users can be determined. Since the scoring of this instrument is relatively straightforward, and because there is little room for subjective interpretation of the responses, it is likely that the computer-administered form of the questionnaire produces results that are as reliable as the face-to-face version of the same. In fact, other studies (Ford et al., 1996; Peterson et al., 1996) have shown that individuals are sometimes more likely to answer truthfully when presented with a computer-based questionnaire as opposed to one administered by another person.

Data collected via the ILP-O can also be used in a series of ongoing validity and reliability studies of the instrument at considerable savings compared with the costs of administering a conventional paper-and-pencil form of the instrument.

6.2. ELSA as an Agent

ELSA, in contrast to most agents, is not designed around a particular learning module. Instead, ELSA is based on principles and specific parameters about the module content. The, the "intelligence" is provided by the individual who codes the ELSA tags, and ELSA merely acts as an agent on that individual's behalf.
6.3. **ELSA as a Test-Bed for ATI Research**

Much of the research on ATIs is plagued by noise in the data set (Shute, 1993a). ELSA provides a viable test-bed for future ATI research that eliminates many of the sources of noise, such as variation in classroom environment and teacher. By using ELSA, ATI researchers should be able to replicate experiments easily, particularly in light of recent gains in the popularity of the Internet.

ELSA is delivered using a widely available and commonly used web server (or http daemon), which has built into it a highly configurable event logging system. The built-in event logs provide a rich source of data, including the user identification, time of access, and URL accessed. These data may be used for further analysis of individuals’ behaviours without having to use more expensive and cumbersome technologies such as video recording.

6.4. **Balancing Micro- and Macroadaptation**

ELSA is capable of both micro- and macroadaptation.

ELSA macroadapts the presentation of instructional modules based on information derived from the Inventory of Learning Processes Online. Based on a particular user’s ILP-O profile, ELSA will render appropriately coded HTML fragments in such a way as to reflect the original author’s intent. This process is similar to the face-to-face teacher’s administration and analysis of a learning style instrument (such as the ILP) and subsequent tailoring of instruction based on those results.
ELSA accomplishes microadaptation by updating a user's profile when the user selects a hyperlink from within an ELSA-encoded block of HTML. For example, consider a user who was identified via the ILP-O as being of learning style X. That user is then presented a page from an instructional module and she selects a hyperlink from a section of text that has been identified by the author as being most appropriate for a user whose learning style has been identified (again via the ILP-O) as learning style Y. At that point, the user's profile is updated to reflect her tendency to be "Y-like" via a simple averaging algorithm. This is in keeping with Schmeck's (1983) concept of modifiable learning styles.

The weighting of the user's current selection relative to her identified learning style represents the degree to which ELSA favours microadaptation over macroadaptation. If ELSA is tuned to do simple averaging (i.e. for each ILP score, average the value of the user's score with the midpoint of the ELSA-encoded HTML fragment), then microadaptation will be prominent. Conversely, if ELSA is tuned to do weighted averaging (e.g. for each ILP score, compute the weighted average of the user's score with the midpoint of the ELSA-encoded HTML fragment, weighting the user's score by an arbitrarily large proportion), then macroadaptation will remain the dominant form of adaptation displayed by the system.

The user's learning style is represented as a single point in four-dimensional space (the four dimensions correspond to the four factors of the ILP). This is conceptually difficult to envision, and it is conceivable that authors may inadvertently create documents that contain ELSA tags that do not adequately cover the typical ILP score ranges found in the user population. In this first implementation of ELSA, the author is
responsible for ensuring that such ranges are adequately covered. In the future, I hope to
develop an automated validity checker similar to the Unix lint utility that checks C
programs for common errors.

The representation of a user profile as a single point in multidimensional space is,
nevertheless, a simplification. I think the notion of a "probability cloud" that describes
the location of a user's learning style in 4-D space as a probability density function is a
more realistic representation. Obviously, representing a user's profile as such would
require substantially more sophisticated data storage and manipulation than is currently
available with ELSA.

Another issue that should be examined is the relative importance of the four ILP
factors. Since the factors were determined using a factor analysis, subsequent factors
likely explain less and less variance. Authors of ELSA documents should keep this fact
in mind when coding documents, and they may want to focus on factors I and II (Deep
Processing and Methodical Study) rather than factors III and IV (Fact Retention and
Elaborative Processing).

Because ELSA can access documents via HTTP from anywhere on the Internet,
users who wish to use the system need not implement ELSA on their own server,
although they may wish to do so in order to improve response time and minimize their
reliance on network connections to remote sites. All components of ELSA are available
at no cost to educational institutions should individuals desire to implement ELSA at
their own institution.
ELSA is designed to parse almost any HTML document retrievable via HTTP. If the page that ELSA retrieves contains Java tags or JavaScript, the page should still pass through ELSA without problems (although in the current implementation these retrieved objects will also pass through the ELSA filter, which may cause problems if the objects inadvertently contain ELSA tags).

In its current implementation, ELSA cannot retrieve documents via ftp, gopher, or local file access. This limitation can be overcome by employing additional modules that support network connections using these protocols (in the case of ftp and gopher), or by using simple file access (for local file access).

When ELSA is called, the server process is tied up (i.e., it cannot serve documents to other users) while ELSA retrieves remote documents. Most new servers support multiple forked child processes to handle incoming requests (via http) for documents, so a brute-force approach to addressing this problem is to increase the number of "spare" server processes to cope with the demand placed by ELSA on the http server. This inelegant solution does not scale well. A better solution is to use a caching proxy http server, which stores copies of retrieved documents locally, thereby minimizing the time required to retrieve remote documents. The most recent version of the Apache server (currently version 1.1.1) can act as a caching proxy server, and the current implementation of the agent employs this feature.

ELSA cannot retrieve documents from authenticated realms. This is a limitation imposed by the design of the agent in general and the use of LWP::UserAgent in particular, as user authentication is used to identify the user's profile. This limitation
could be overcome by storing in the mSQL database the userid and password for authenticated realms. However, doing so may compromise the security of such web pages.

Currently, there is no mechanism to check for the validity of an ELSA block. If there are unmatched ELSA tags, the agent will not inform the user of any problems, but rather it will treat the remainder of the document as part of the same ELSA block. Nested ELSA blocks are not permitted. The format of the ELSA tag is very inflexible in this prototype version.

Currently, the ELSA agent does not respect the material in /robots.txt on a server. Although ELSA is technically a robot, it performs its actions only when actively used by a user (in contrast to various web-crawling robots that "surf" the WWW). Therefore, it could be argued that ELSA should be exempt from the usual rule governing the acceptable behaviour of WWW robots.

6.5. Extensions to ELSA

ELSA currently employs the Inventory of Learning Processes as the psychometric instrument of choice. Whereas this instrument is certainly appropriate (if somewhat dated) for the target audience of undergraduate university-level students, it may be inappropriate for other user populations. ELSA could be generalized to employ multiple, different instruments (and their associated psychometric scales). This generalization could also provide research opportunities for individuals who are interested in examining differences between the various psychometric instruments.
A mechanism by which users can actively override their ILP profiles is required, and is partially implemented. Overriding the user profile is accomplished by appending \texttt{?overrideilp=A;B;C;D} to any URL passed to ELSA.

There is room for improvement in the representation of ILP scores. Currently, ELSA authors must describe parts of their documents using z-score ranges. This is unintuitive, and can be misleading due to the nonlinearity of the normal distribution. A relatively simple enhancement would be the ability to use alternative units (say percentiles) to describe the range of users to which a particular part of an ELSA-encoded document would be shown. A more sophisticated enhancement would be the derivation of common combinations of ILP z-score ranges and the assignment of descriptive names to these combinations (e.g. "basic","concrete-random","left-hemispheric").

Currently, document parts enclosed by ELSA tags that specify ILP score ranges that lie outside the user’s ILP scores are de-emphasized (shown in a "muted" colour if the user is using a graphical browser or as an "aside" if the user is using a less sophisticated browser). An interesting extension to the agent would be allowing the author to indicate what action should be taken if the user’s score is outside the document-specified ILP range. For example, the author could indicate that part of a document, if the user’s ILP score is outside the range, should not be shown at all (although this precludes dynamic updating of the user’s profile).

There is currently no editor that can assist users in the creation of ELSA tags. The most promising technology seems to be JavaScript (in which HTML editors have been written), although adopting it would restrict users to JavaScript-enabled browsers.
Further research needs to be conducted to determine the adequacy of the ILP user profile averaging algorithm. Currently, a user's selection is averaged with her existing profile to calculate the new profile. This may introduce too much variance into the user's profile, so alternative averaging algorithms (e.g. moving averages, median averaging, etc.) should be investigated.

On a broader scale, the central concept of ELSA (i.e. different renditions of the same WWW page for different users based on some arbitrary profile) has the potential to shift the current paradigm of static vs. dynamic web pages. We tend to think of dynamic web pages as those that respond to a user based on conscious choice on the part of the user. ELSA changes to a paradigm in which "decisions" about what a user might want to see are the responsibility of some agent. Whether that agent is capable of making the "correct" choice about what a user might want to see is an exciting field of potential research. Some research about "correctness" of software agents is currently being undertaken by the MIT Autonomous Agents Group (Maes, 1989).
7. Conclusions

I believe that the World Wide Web's potential for use has been greatly underestimated due to the perception that static web pages are the *de facto* standard. Tim Berners-Lee's concept of a simple yet powerful markup language (HTML) has allowed millions of users to provide material online, but these information providers seem to be constrained by the paradigm of static representation of information. It is the area of dynamic web documents that holds great promise for the future of the WWW, and in particular in the field of education via the World Wide Web. I hope that by creating ELSA, and by offering this crude implementation of it, other educators will reconsider how they teach with the World Wide Web. There is much research to be done here. Let us capitalize on the potential for new and interesting research opportunities.

The ELSA system is an Internet agent that is capable of retrieving documents via HTTP and rendering those documents according to information about the learning style of specific users. The ELSA system provides a functional test bed for research into aptitude-treatment interactions, as it controls for a variety of confounding factors that have plagued previous ATI research.
8. References

http://www.sn.no/libwww-perl/


Station, TX: ASHE-ERIC.


Shute, V.J. (1993a). A comparison of learning environments: all that glitters.... In S.P. Lajoie & S.J. Derry (Eds.) *Computers as cognitive tools* (pp. 47-74). Hillsdale, NJ:
Erlbaum.


9. Appendix 1: The Inventory of Learning Processes Online (ILP-O)

#!/usr/local/bin/perl

# Perl CGI script to generate and process ILP form and then
# process it.

# Questions are from
# report inventory for assessing individual differences in learning processes.

# Perl script (C) 1996 by Chris Teplovs (chris.teplovs@utoronto.ca)
# no part of this code may be reproduced in any way without the express
# written consent of the author

use CGI qw(:standard); # import names from CGI
use Msql; # and use Msql stuff

print header; # print the HTTP header in case errors are coming

# Here I open the data file and read each line (lines beginning with a "#"
# are comment lines and are discarded), which is split into the text of the
# question, the loading, the factor number, and the code (the latter was of
# my own devising). I also construct two associative arrays for later use:
# %aqfactor, which allows me to look up the factor number associated
with a particular question, and $aqloading, which allows me 
# to look up the loading associated with a particular question.

open(QUESTIONS, "/home/chris/public_html/1551s96/questions.txt") ||
  print("Couldn’t open data file.") && exit(1);
$nquestions = 0;
# loop around the questions, reading each line and splitting it into 
# four fields. Note that split assumes $._
while (<QUESTIONS>) {
  next if (/^#/);
  chomp;
  ($qtext[$nquestions],$qloading[$nquestions],$qfactor[$nquestions],$qcode[$nquestions]) =
    split(""');
  # now create some useful associative arrays
  $aqfactor{$qcode[$nquestions]}=$qfactor[$nquestions];
  $aqloading{$qcode[$nquestions]}=$qloading[$nquestions];
  $nquestions++;
}

$nquestions--;

# Here I assign to a variable ($javascript) the 
# entire JavaScript function that will
# be associated with the submit button. The function is called 
# validate and takes the form as an argument. Two variables are 
# used: one is a boolean which is used to keep 
# track of any unanswered questions, and the other is a string that 
# contains the message displayed to the user if the confirm window is 
# opened.

# Since I know that 
# the checkboxes of the radio buttons form consecutive 
# elements within the form, processing is simplified because I can 
# use numeric references to the elements rather than the names 
# of the elements.

# A for loop iterates through the question numbers and checks that 
# at least one of the two checkboxes is filled in.

# If that's not the case, 
# the variable filledin is set to false. Once the loop is completed, 
# a confirm window is created if filledin is false.

```javascript
var confirmStr = "At least one question remains unanswered." + 
"You may continue, but your scores will be inaccurate.";

filledin = true
```

```
for (i=0;i<\$nquestions;i++) {
    if (!((form.elements[i*2].checked) || (form.elements[(i*2)+1].checked)))
        filledin = false
}
if (!filledin) return(confirm(confirmStr))
}

EOJS

unless (param()) {

    # in case we need to see STDERRish stuff
    print start_html(-title=>'ILP',-script=>\$javascript); # print the HTTP header,
    print h1('Inventory of Learning Processes (ILP)'),# an H1 heading
    Applied Psychological Measurement 1(3): 413-431.".
    "The following series of questions was developed by Schmeck \textit{et al.} as a measure of behavioural learning style. Feel free to
    complete this form and then press
    out your scores. The scores are valid only if you answer all questions.
    The data you provide are permanently stored on the host
    computer. You can also \texttt{<A HREF=}
    \texttt{well as \texttt{<A HREF=}
    \texttt; } # semi-colon to indicate end of print
# now print the list of questions

# the documentation seems to be wrong -- it suggests startform:

print start_form(-onSubmit=>"return validate(this)"):

# Note that the JavaScript validate

# function is called as return validate(this) rather than just

# validate(this) which allows the user to abort the submission of

# the form. Printing the questions is accomplished by using a for loop.

print "<TABLE BORDER=1">

for ($i=0;$i<=$nquestions;$i++) {
    print "<TR><TD>".$i+1.".".$qtext[$i]."</TD><TD>".
        radio_group(-name=>$qcode[$i],
            -values=>['True','False'],-default=>'Neither');
    "</TD></TR>
"
}

print "</TABLE>

print submit(-name=>'Calculate Scores')," .reset:

print end_form ;

print end_html;

print "\n": # just to make output look nicer

} else {

# got some form input, let's try to calculate some scores

# need to get some form of associative array here to look up

# values from form’s field names
print start_html(-title=>"ILP Results");

for ($i=0;$i<4;$i++) { $score[$i]=0; }

# score if loading < 0 and False, or if loading > 0 and True

foreach $key (param()) {
  if ( (($aqloading{$key} > 0.0) && (param($key) eq 'True') ) ||
      (($aqloading{$key} < 0.0) && (param($key) eq 'False')) )
    {$score[$aqfactor{$key}-1]++;}
}

# of foreach

# okay, now we have the scores in @score, let's generate some
# meaningful output

# these are the published scores (p. 420)

@pubscores = (11.6, 10.4, 4.7, 10.2):
@pubsd = (4.3, 4.4, 1.8, 2.8):
@scale = ("Deep Processing","Methodical Study","Fact Retention","Elaborative Processing"): # I should round the values, but it's less than straight-forward because
# I also have -ve values

for ($i=0;$i<4;$i++) {
  $z[$i]=int(((score[$i]-$pubscores[$i])/$pubsd[$i]*100))/100;
}

print h1(Results).

"<TABLE BORDER=1><TR><TH>Scale</TH><TH>Raw Score</TH>
<TH>Standardized Score</TH>

for ($i=0;$i<4;$i++) {
    print "<TR><TD>".$scale[$i]."<TD>".$score[$i]."<TD>".$z[$i]."</TD><TR>\n";
}
print "</TABLE>

# now write the scores to the Msql database
$dbh = Msql->connect("localhost","chris");

$ssth =
$dbh->query("update ilp set ilp1=$z[0],ilp2=$z[1],ilp3=$z[2],ilp4=$z[3] where userid='test1'") if $dbh;
print h1(Interpretation);
print <<EOM;
You’re probably most interested in the column labelled
"Standardized Scores", which represents the z-scores of the
values in the "Raw Scores" column.

<P>
You should be able to figure out your approximate
percentile rank with the following information:

<UL>
  <LI>a z-score of 0 corresponds to a percentile rank of 50
  <LI>a z-score of 0.26 (-0.26) corresponds to a percentile rank of 60 (40)
  <LI>a z-score of 0.53 (-0.53) corresponds to a percentile rank of 70 (30)
  <LI>a z-score of 0.84 (-0.84) corresponds to a percentile rank of 80 (20)
Appendix I

A z-score of 1.28 (-1.28) corresponds to a percentile rank of 90 (10)

A z-score of 2.33 (-2.33) corresponds to a percentile rank of 99 (1)

The Deep Processing score indicates your ability "to glean organization from a unit of material as well as [your] ability to reorganize it" (Schmeck et al., 1977 p.427). A high score suggests that you "evaluate content more carefully before making choices on a recognition test" (Schmeck et al., 1977 p.427)

Methodical Study refers to the degree to which you adhere to traditional study techniques and behaviours (Schmeck et al., 1977). High scores on this scale correspond to one who would be labelled as a "good student" in the traditional sense (Schmeck et al., 1977).

Fact Retention is a measure of your "ability to store or retain detailed factual information" (Schmeck et al., 1977, p.428) and is considered to be an assessment of memory capacity.

Elaborative Processing "seems to assess the lengths to which a student will go to encode new information" (Schmeck et al., 1977, p.428).

Literature Cited

Schmeck, R.R., F. Ribich, and N. Ramaniah. 1977. Development of a self-
report inventory for assessing individual differences in learning processes.


<HR>

You might want to check out the

<A HREF="elsacatalog.html">Catalog of ELSA-enabled Instructional Modules</A> now that your profile is recorded in the database.

EOM

print end_html:

} # of else
10. Appendix 2: The Inventory of Learning Processes Online (ILP-O) Data File

I can easily handle questions requiring comparison of different concepts. 0.53  DC
I have trouble making inferences. -0.51  MI
I have trouble organizing the information that I remember. -0.47  OM
Even when I know that I have carefully learned the material I have trouble remembering it for an exam. -0.46  TR
I find it difficult to handle questions requiring critical evaluation. -0.46  CE
I do well on essay tests. 0.46  ET
I often have difficulty finding the right words for expressing my ideas. -0.43  RW
I have difficulty learning how to study for a course. -0.43  SC
I have difficulty planning work when confronted with a complex task. -0.41  CT
I get good grades on term papers. 0.41  TP
I often memorize material that I don’t understand. -0.37  NU
I have trouble seeing the difference between apparently similar ideas. -0.35  TS
I can usually state the underlying message of films and readings. 0.34  UM
I think fast. 0.32  TF
Most of my instructors lecture too fast. -0.31  LF
I can usually formulate a good guess even when I don’t know the answer. 0.26  GG
I try to resolve conflicts between the information obtained from different sources. 0.25  RC
I read critically. 0.25  RD
I cram for exams. -0.47  CR
I have regular weekly review periods. 0.46  WR
Getting myself to begin studying is usually difficult. -0.45  BS
I review course material periodically during the term. 0.45 PE
I maintain a daily schedule of study hours. 0.44 DS
I carefully complete all course assignments. 0.41 CA
I rarely write an outline of the material I read. -0.39 OU
I spend less time studying than most of my friends. -0.36 TI
I prepare a set of notes integrating the information from all sources in a course. 0.36 NO
I rarely read beyond what is assigned in class. -0.35 BY
I usually refer to several sources in order to understand a concept. 0.34 SS
Toward the end of a course I prepare an overview of all material covered. 0.33 EN
I increase my vocabulary by building lists of new terms. 0.32 VO
I rarely use a dictionary. -0.32 DI
Even when I feel that I've learned the material, I continue to study it. 0.32 CO
I make simple charts and diagrams to help me remember material. 0.30 CD
I always make a special effort to get all the details. 0.28 DE
I do not usually work through practice exercises and sample problems. -0.28 EX
I have a regular place to study. 0.27 PL
I have difficulty locating particular passages in a textbook when necessary. -0.25 TX
I would rather read a summary of an article than the original article. -0.25 OR
I rarely use the library. -0.25 LI
When studying for an exam I prepare a list of probable questions and answers. 0.25 ST
I do well on exams requiring much factual information. 0.57 FA
I am very good at learning formulas, names and dates. 0.53 FO
I do well on tests requiring definitions. 0.44 DF
I do poorly on completion items. -0.38 CI

I have trouble remembering definitions. -0.34 DR

My memory is actually pretty poor. -0.33 PM

For exams, I memorize the material as given in the text or class notes. 0.26 ME

I rarely look for reasons behind the facts. -0.51 LR

New concepts rarely make me think of many other similar concepts. -0.46 NC

While studying I attempt to find answers to questions I have in mind. 0.44 IM

I am rarely able to design procedures for solving problems. -0.41 PS

I rarely sit and think about a unit of material which I have just read. -0.38 SI

I learn new words or ideas by visualizing a situation in which they occur. 0.33 NW

When learning a unit of material I usually summarize it in my own words. 0.33 IS

I learn new concepts by expressing them in my own words. 0.32 MW

I daydream about things I've studied. 0.27 DD

When I study for something I devise a system for recalling it later. 0.26 SY

I learn new words and ideas by associating them with words and ideas I already know. 0.26 AS

I learn new words and ideas by relating them to similar items. 0.25 RL

I do not try to convert facts into "rules of thumb". -0.25 RT

While learning new concepts their practical applications don't usually come to my mind. -0.25 CM
11. Appendix 3: ELSA

#!/usr/local/bin/perl

#

# This is ELSA: Experimental Learning Styles Agent
#

# Copyright (C) 1996 Chris Teplovs

# No part of this code may be reproduced by any means without the
# express written consent of the author
#

# I import the CGI module names
# into my namespace (with qw(:all)) to simplify coding later on.
# For example, I can then code a level 1 heading as
# Ch1("Heading") rather than $query->h1("Heading").
# This short-cut provides for more readable, if somewhat more
# cryptic, perl code.

use CGI qw(:all);

# import other CGI Modules

use LWP::UserAgent;

use URI::Escape;

use Msql;

use Carp;
$l=1; # unbuffer output so nothing strange happens

$elsaversion="1.2";

# if we can’t find an ilp set, use z-score=0 (=50th %ile)
@nullilp=(0,0,0,0);

$mutedcolour="999999"; # some rather grey colour... "grey" doesn’t work

$graphic_browsers="Mozilla":

#$user = remote_user();

$user='test 1';

$doc_to_be_parsed = path_info();

&update_user_profile(uri_unescape(param('elsailp')),$user) if

   (param('elsailp') && !param('overrideilp'));

#&htmlerror("User must be authenticated.") unless $user:

&htmlerror("No path name given.") unless ($doc_to_be_parsed && ($doc_to_be_parsed ne "/"));

$ua = new LWP::UserAgent;

$ua->agent("ELSA/0.1");

$ua->proxy("http", "http://localhost:80/");

$request = new HTTP::Request 'GET', "http://".$doc_to_be_parsed;

#$request = new HTTP::Request 'GET', path_info();

$response = $ua->request($request);

$document = $response->content;

# note the funny *? regexp... minimizes matching (otherwise matching is greedy)
while ($document =~ /<ELSA "(.*)"(.*?)<VELSA>/is) {
    $docilp = $1;
    $test = &checkilp($docilp,$user);
    # this is an evaluation of an expression is used for the
    # substitution
    if ($test) {
        $document =~ s/<ELSA (.*?)(.*?)<VELSA>/&process_hrefs($2,$docilp)/ies:
    } else {
        user_agent() =~ /$graphic_browsers/ ?
        $document =~ s/<ELSA (.*?)(.*?)<VELSA>/"<FONT COLOR="$mutedcolour">".
        &process_hrefs($2,$docilp)."<VFONT>"/ies :
    $document =~ s/<ELSA (.*?)(.*?)<VELSA>/"<DL><DT>Aside:<DD>".
        &process_hrefs($2,$docilp)."<VDL>"/ies:
    }
print $document,"\n";
    # and let the user know that ELSA's been here
    print hr.i(ELSA)," Version $elsaversion\n";
    exit(0);

    # Subroutines follow
# checkilp: tests user's ilp against ELSA tag contents
# calls getuserilp, returns true if user's ilp is within
# specified range, otherwise false.

sub checkilp {
  my($elsailp, $user) = @_; 
  local(@mins, @maxs);
  &splitilps($elsailp, *mins, *maxs);
  my(@userilp) = &getuserilp($user);
  $return = 1; # start off true
  for ($i=0; $i<4; $i++) {
    $return = 0 if ((@userilp[$i] < @mins[$i]) || (@userilp[$i] > @maxs[$i]));
  }
  return $return;
}

# getuserilp: returns array of user ilps
# This subroutine extracts the user's ILP from the mSQL database.
# using the value of @nullilp in the event of a problem
# with the user profile database. The subroutine returns a four-element
# list of the user's ILP standardized scores.

sub getuserilp {
  my($user) = @_; 
  my(@results);

my($dbh) = Msql->connect("localhost","chris");

my($sth) = $dbh->query("select ilp1,ilp2,ilp3,ilp4 from ilp where userid='$user'") if $dbh:

if ($sth) {
    @results = $sth->fetchrow;
} else {
    @results = @nullilp;
}

if(scalar(@results)) {
    return @results;
} else {
    return @nullilp;
}

# process_hrefs:

# This subroutine updates the HREFs in an ELSA block so that the
# user's profile can be updated when a link is selected. The ELSA
# block is passed to the local variable $data, which is
# then parsed for HREFs. When an HREF is found, the URI::Escape'd
# value of the ELSA block's ILP ranges is tacked onto the end of the
# URL (note that processing differs based on the presence of
# a "?" in the URL).

sub process_hrefs {
my($data,$ilp) = @_;  

$ilp = uri_escape($ilp,"\0-377");  

if ($data =~ /^A/?) { # special case if link already has parameters  
    $data =~ s/A HREF="\((\[\])\)+"*/A HREF="\$1\&elsailp=$eilp"/g;  
} else { # no parameters, so we can stuff a ? in the url  
    $data =~ s/A HREF="\((\[\])\)+"*/A HREF="\$1?elsailp=$eilp"/g;  
}  

return "$data";  

}  

# update_user_profile subroutine  
# This subroutine is called when a user selects an ELSA-encoded  
# URL. The arithmetic mean of each ILP scale’s range is averaged with  
# the user’s ILP and the resulting profile is written to the  
# mSQL database.  

sub update_user_profile {  
    local($ilp,$user) = @_;  
    local(@mins,@maxs,@newilp);  
    local(@userilp) = &getuserilp($user);  
    &splitilps($ilp,*mins,*maxs);  
    for ($i=0;$i<4;$i++) {  
        # check and update each user ilp  

push(@newilp,$userilp[$i]+(($mins[$i]+$maxs[$i])/2))/2)
;

my($dbh) = Msql->connect("localhost","chris");

my($sth) = $dbh->query("update ilp set ilp1=$newilp[0],ilp2=$newilp[1],
ilp3=$newilp[2],ilp4=$newilp[3] where userid='$user'") if $dbh;

# splitilps subroutine

# This is a utility subroutine that takes a document ILP and
# splits it into a list of maxima and a list of minima.
# Note that the arrays are passed by reference, which is a very
# useful perl technique.

sub splitilps {
  local($ilp,*mins,*maxs) = @_;;

  my(@elsailps) = split("\",$ilp);
  foreach $pair (@elsailps) {
    my($min,$max) = split("\",$pair);
    push(@mins,$min);
    push(@maxs,$max);
  }

  # htmlerror subroutine
# This is a simple error-printing routine that prints a client-friendly
# message to the user in the event that something goes wrong with the
# script, such as access by an unauthenticated used (in the final
# version of the agent) or the passing of a null string as the PATH_INFO.

sub htmlerror {
  my($error) = @_;

  print "<H1>ELSA Error:</H1>";$error;"\n";

  exit(1);
}
