

Extending the Learning Environment: Virtual Professors in Education

By Mary Harrsch
Network & Management Information Systems
College of Education, University of Oregon

Six years ago, my sister was telling me about a fascinating [History Alive](#) Chautauqua event she had attended near Hutchinson, Kansas. The program brings a reenactor portraying an historical figure into schools and communities for an educational presentation and question and answer session. I thought to myself, “It’s too bad more people can’t take advantage of such a unique learning experience.” Then, the technologist within me began to wonder if there was a way to create a virtual Chautauqua experience online. As I pondered this possibility, I realized that if I could find software that could be used to create a “virtual” person online, I could not only recreate the experience of the Chautauqua, but provide a tool faculty could use to answer course-specific questions. It could even be used to provide information about the professor’s personal interests and research to enhance the sense of community within the learning environment.

My quest led me to a [website](#) that included links to a number of different software agent projects. I learned that the type of agent I needed was commonly called a “chatterbot”. The first “chatterbot” was actually developed long before the personal computer. In the early 1960s, [Joseph Weizenbaum](#) created “Eliza”, a virtual psychoanalyst. In his efforts to create a natural language agent, Weizenbaum pointed out that he had to address the technical issues of:

- the identification of key words,
- the discovery of minimal context,
- generation of responses in the absence of keywords

As I began to explore different agent implementations, I found that, in addition to these issues, the application needed to be able to prioritize keywords to discern the most appropriate response. Several agents I evaluated, including [Sylvie](#), a desktop assistant, developed by Dr. Michael (“Fuzzy”) Mauldin, Artificial Life’s [Web Guide](#), [Carabot 500](#) developed by U.K. company, Colorzone, and Kiwilogic’s [Linguibot](#), used slightly different methods to set the priority of subject keywords to select the most appropriate responses. The response with matching keywords under the subject with the highest level setting was “fired” – displayed to the user. However, when editing their script files, I found keeping track of subject priorities was challenging.

Another problem with many script-driven agents I evaluated was the use of left-to-right parsing sequences that did not compensate for a variance in the order of keywords in a question. Each query had to be evaluated for subject and for matching character strings, based on left-to-right word order with the use of various “wildcard” characters to indicate

placement of keywords within the overall question. Therefore, you often had to have multiple script entries to compensate for different word order. For example, if a student asks “How do I change my password in e-mail?” you would need one script entry. If the student asks “How do I change my e-mail password?” a different script entry would be required:

- * email * * password * as well as
- * password * * email * to trap for either wording.

Although this attention to script design resulted in improved response accuracy the scripting knowledge required for these agents was not something I would expect a faculty member to have the time or desire to learn.

A third problem with several of the agent applications I used was the necessity to unload and reload the agent each time the script file was edited. If students were actively querying the agent, you could not save any script changes until the script file was no longer in use.

When I invested in the Enterprise Edition of Artificial Life’s WebGuide software, I also realized the importance of a logging feature that I could use to study and improve my guide’s responses. I recognized the importance in a virtual tutoring environment of having the ability for a student to print out a transcript of their tutoring session for future study. Not only was this feature absent in the agents I evaluated, but the responses produced using Javascript or Flash would not allow the user to highlight and copy responses to the clipboard either.

One day, I explored [UltraHal Representative](#), developed by Zabaware, Inc. I liked the ability Ultrahal provided to program the agent through a web interface. It had the capability to include links to related information. It could be customized with personalized graphics. It logged interactions. Best of all, it had a straightforward approach to editing - no scripting – just type your question three different ways then type your intended response. [Figure 1]

But, I soon discovered that without the ability to identify keyword priority, I found that the results using whatever algorithm was built into the agent engine were too inaccurate for a virtual tutoring application.

I needed a product that could be programmed to be “omniscient”.

“Effective ITS require virtual omniscience -- a relatively complete mastery of the subject area they are to tutor, including an understanding of likely student misconceptions.”
(McArthur, Lewis, and Bishay, 1993)

I needed a virtual professor that could be “programmed” by real professors, the individuals who would have a mastery of the subject and an understanding of student misconceptions. But all of the chatterbots I had encountered, so far (with the exception of

Ultra Hal), required knowledge of scripting that most faculty members do not have the time to learn. I would not have the time to provide one-on-one time with faculty developers and paying a programmer to work with a faculty member is also too expensive. (I noticed most developers of commercial agents actually relied on the scripting needs of their clients for their primary revenue stream.) So, I decided to attempt a radically different approach to agent design.

I am an experienced Filemaker Pro solutions developer and one day I was reviewing some of Filemaker's text functions and realized that the position function could be used to detect key words in a text string. The beauty of the position function is that the keyword can be identified anywhere within the target text. It is not dependent on a left to right orientation. Filemaker is also not case sensitive. Also, the new version 7 allows most scripting calls for text processing to be used with their Instant Web Publishing interface. I realized this would greatly simplify web integration.

So, reviewing my experiences with the agent applications I had used, I developed a list of features that I wanted to incorporate:

Web functionality:

- Multiple agents controlled by a single administration console
- Web-based query interface
- Web-based editing interface
- Multiple graphic format support
- Web accessible logging function for both agent editor and student user
- Ability to display related resources

Query processing functionality:

- Question context awareness (who, what, when, where, why, how, etc)
- Ability to weight keywords by importance without user scripting
- Ability to return an alternate response if a question is asked more than once
- Ability to use one response for different questions
- Ability to process synonyms, international spelling differences, and misspellings
- Independent of word order
- Not case sensitive

Structural Design:

- Modular design to enable importation of knowledge modules developed by others
- Agent specific attributes to customize the interface and responses such as a personal greeting, the opportunity to use the person's homepage as a default URL, information about area of expertise and research interests for alternative agent selection criteria, custom visual representations, etc.

I began by designing my response search criteria. I programmed the agent search routine to categorize responses by the first word of the query – usually What, Where, Why, How, Who, Did, Can, etc. to establish the question context. Then I used position formulas to

test for the presence of keywords. I then developed an algorithm that weighted the primary keyword or synonym and totaled the number of keywords found in each record.

I designed the search function so that when the visitor presses the button to ask their question, the database first finds all responses for the question category (who, what, when, etc.) containing the primary keyword (or its synonym). Responses are then sorted in descending order by the total sum of keywords present in each response. The first record – the one with the most keyword matches – is displayed as the answer.

If there are no category responses containing the primary keyword, then a second find will execute to look for all responses with the keyword regardless of category. In working with other agent products, I have found that if you return a response with at least some information about the keyword, even if it is not an exact answer to the question, the student assumes the agent recognized their question and may learn auxiliary information that is still helpful to them.

For example, if a visitor asks [my virtual Julius Caesar](#) if he really loved Cleopatra, he will answer “Cleopatra...ah, what an intriguing woman.” Not only is this more in character with Caesar (most of his female dalliances were for political reasons) but the answer could also be appropriate for a different question, “What did you think of Cleopatra?” My search routine would find it in either case because of the weighting of the primary keyword, Cleopatra.

If there are no responses containing the primary keyword, a third find looks for any generic category responses. For example, if a student asks who someone is and you have not programmed your agent with a specific answer for the keyword (the person they are asking about), the agent will reply with an appropriate “who” response such as “I’m afraid I’ve never made their acquaintance.”

If a student’s question does not begin with any words set as category words, the last find will return a generic “what” response such as “I may be a fountain of knowledge, but I can’t be expected to know everything.” Programming the agent with default generic responses, ensures that the agent always has something to say, even if it knows nothing about the subject. I developed a small database of generic responses for each question category that is imported into an agent database each time a new agent is created. The faculty member can go through the responses and edit them if they wish.

Next, I turned my attention to the faculty’s content editing interface. I wanted the faculty member to enter a proposed question, designate a primary keyword and synonym, supply any other keywords they thought were important to identify more precisely the desired response, and the desired response. Using the Ultrahal interface as my inspiration, I created the editing layout illustrated in Figure 2.

I also provided a button that enables a faculty member to quickly generate a different answer for the same question or a different question for the same response.

I created a field that is populated with a different random integer on each search. By subsorting responses by this random integer, it enables the agent to offer a different response to the same question if the question is asked more than once. This supports the illusion of the agent being a “real” person because it will not necessarily return the same identical response each time.

“Believable agents must be reactive and robust, and their behaviors must decay gracefully. They must also be variable, avoiding repetitive actions even when such actions would be appropriate for a rational agent. They must exhibit emotion and personality. Ultimately they must be able to interact with users over extended periods of time and in complex environments while rarely exhibiting implausible behaviors.” – Dr. Patrick Doyle, [*Believability through Context: Using “knowledge in the world” to create intelligent characters*](#)

With the “engine” of my agent developed, I turned my attention to the visual representation of the character. In their paper, [*The Relationship Between Visual Abstraction and the Effectiveness of a Pedagogical Character-Agent*](#), Hanadi Haddad and Jane Klobas of Curtin University of Technology, Perth, Western Australia, point out the divergent views of information systems designers outside the character-agent field with those developers within it.

“Wilson (1997) suggests that more realistic character-agents may introduce distraction associated with the user’s curiosity about the personality of the character and overreading of unintended messages because of presentation complexity.”

Unlike detailed realistic drawings, sketches help focus the mind on what is important, leaving out or vaguely hinting at other aspects. Sketches promote the participation of the viewer. People give more, and more relevant, comments when they are presented a sketch than when they are given a detailed drawing. A realistic drawing or rendering looks too finished and draws attention to details rather than the conceptual whole (Stappers et al, 2000).

“On the other hand, research by psychologists suggests that people may put considerable cognitive effort into processing abstract representations of faces (Bruce et al. 1992; Hay & Young 1982). It is possible, therefore, that response to anthropomorphised character-agents, and especially their faces, may differ from responses to sketches. Gregory and his colleagues (1995) conducted studies on human response to faces at the physiological level. They demonstrated that humans are particularly receptive to faces. In terms of recognition, participants in their studies were more responsive to real faces than to abstracted line faces. They speculated, however, that people spend longer studying abstracted line faces and may find them more interesting (Gregory et al. 1995). If this is so, then contrary to theories of information design, an abstract face may introduce more distraction into the communication than a realistic face.”

Filemaker Pro 7 provides multimedia container fields that enable me to include still images, animations, or even video clips. However, not only is creating a unique graphic for each response time consuming, motion video files can be quite large and slow down

the delivery of response information over the web. Working with other agents, I had noticed that just the slight eye movement of a blink can be enough to reinforce the illusion of a sense of presence. This approach straddles the two opposing theories described above. I would utilize a real face to capitalize on the human receptivity to a

framework for questions, a focus for questions, and different levels of questions. Well-designed inquiry learning produces knowledge formation that can be widely applied.” - [Thirteen Ed Online](#).

References:

McArthur, David, Matthew Lewis, and Miriam Bishay. "The Roles of Artificial Intelligence in Education: Current Progress and Future Prospects". 1993. Rand. <<http://www.rand.org/education/mcarthur/Papers/role.html#anatomy> >.

Doyle, Patrick. "Believability through Context Using "Knowledge in the World" to Create Intelligent Characters." Trans. SIGART: ACM Special Interest Group on Artificial Intelligence. International Conference on Autonomous Agents. Session 2C ed. Bologna, Italy: ACM Press New York, NY, USA, 2002. 342 - 49 of Life-like and believable qualities.

Haddad, Hanadi, and Jane Klobas. "The Relationship between Visual Abstraction and the Effectiveness of a Pedagogical Character-Agent." The First International Joint Conference on Autonomous Agents & Multi-Agent Systems. Bologna, Italy, 2002.

Wilson, M. "Metaphor to Personality: The Role of Animation in Intelligent Interface Agents." Animated Interface Agents: Making them Intelligent in conjunction with International Joint Conference on Autonomous Agents. Nagoya, Japan, 1997.

Stappers, P., Keller, I. & Hoeben, A. 2000, 'Aesthetics, interaction, and usability in 'sketchy' design tools', *Exchange Online Journal*, issue 1, December, [Online], Available: http://www.media.uwe.ac.uk/exchange_online/exch1_article4.php3 [2004, August 3].

Bruce, V., Cowey, A., Ellis, A. W. & Perrett, D. L. 1992, *Processing the Facial Image*. Oxford, UK, Clarendon Press.

Hay, D.C., Young, A.W. 1982, 'The human face', in *Normality and Pathology in Cognitive Function*, Ellis, A.W. ed., London, Academic Press, pp. 173-202.

Gregory, R., Harris, J., Heard, P. & Rose, D. (eds) 1995, *The Artful Eye*, Oxford University Press, Oxford.

"Thirteen Ed Online: Concept to Classroom". 2004. Educational Broadcasting Corporation. 8/9/04 2004. <<http://www.thirteen.org/edonline/concept2class/> >.

Hodges, Dr. Andrew. "The Alan Turing Internet Scrapbook". Oxford, 2004. (3/15/2004): University of Oxford. 8/09/04 2004. <<http://www.turing.org.uk/turing/scrapbook/test.html> >.

