

Integration of an Expert Teaching Assistant With Distance Learning Software

Steven P. Fonseca and Nancy E. Reed

Computer Science Department
University of California, Davis, CA 95616-8562
fonseca@ece.ucdavis.edu, nereed@ucdavis.edu

The Remote Teaching Assistant (RTA) software currently under development at UC Davis allows students and Teaching Assistants (TA's) to interact through multimedia communication via the Internet. To resolve the problem of TA unavailability and limited knowledge, an Expert Teaching Assistant (ETA) module is being developed. When TA's are not on-line, students in need of help consult ETA. The focus of this research is the development and integration of ETA with RTA, the establishment of an architecture suitable for use with education (the domain) in any sub-domain (course), and the creation of a mechanism usable by non-technical personnel to maintain knowledge bases.

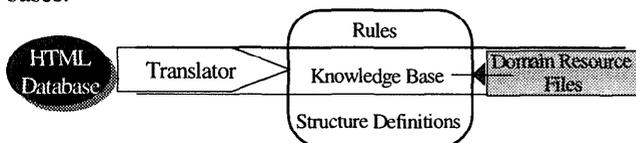


Figure 1: Basic System Architecture for the Expert TA.

The primary goal of ETA is to construct a set of Uniform Resource Locators (URL's) that index the most useful available web pages in the HTML database. To establish the relationship between information within the HTML documents, a small set of commands have been created that logically connect the information in the database. Pre-defined dependency types structure and chain the data objects. The translator reads two segments of an ETA header contained within comments in the HTML files and produces facts and rules that are compatible with the CLIPS inference engine. The first segment describes the nature of the document including the context and several other attributes. The second segment defines the data objects that inherit the attributes of the document and contain additional fields describing properties of the information. The HTML documents are structured for accuracy and ease of maintenance.

Periodically, the translator is invoked to generate or update the knowledge bases for all sub-domains. The translator scans all HTML files, produces CLIPS compatible code, and stores this for use by the inference engine. The translator organizes the CLIPS code, separating information by document and the scope it should

have when the inference engine runs. Additional information is obtained from the domain resource files where rules that are sub-domain dependent and temporary relationships are kept.

In addition to the rules and facts generated by the translator, the knowledge base also contains knowledge applicable to all sub-domains. The inference engine executes two preliminary stages to prepare for a solution search and an iterative refinement stage to pinpoint possible solutions. Because students may have trouble describing their problem, ETA is designed to help students find a solution.

ETA is called by the RTA server when requested by a student using the student client. The server creates an instance of the ETA module dedicated to this student and acts as an intermediary for all future communication. During the first stage, ETA receives the sub-domain and a series of keywords entered by the student. These words are examined to ensure correct form and to eliminate any inadvertent errors. The goal of stage two is to generate additional pattern matches that are logically connected to the keywords provided by the student. Many different types of relationships may hold between pieces of information. Weakly related material is eliminated from the solution. The solution set is narrowed down by interactively prompting the user with a series of questions based on previous solution states, the initial keywords and stage two pattern matches, and the responses received to previous questions. The student client has a graphical user interface with dialog boxes, radio buttons, and other widgets available to obtain the user's responses. Upon reaching a solution set, ETA sends a ranked list of the URL's to the user. Each URL is paired with a description string that briefly describes the contents of the associated HTML document. Double clicking on this description invokes a browser to display the HTML document.

A prototype implementation of ETA was successfully integrated with RTA during the Fall of 1995 and tested in an introductory computer class at UC Davis. The knowledge base was composed of manually written rules and facts that linked a small database containing question and answer files, terms, and lecture notes. Development is ongoing for this course and others, including non-CS ones.