DOCUMENT RESUME

ED 388 258 IR 017 407

AUTHOR Holweg, G.; Sammer, P.
TITLE HyperM: A Hypermedia System with Extended Question/Answer Dialogs.
PUB DATE 94
NOTE 7p.; In: Educational Multimedia and Hypermedia, 1994. Proceedings of ED-MEDIA 94--World Conference on Educational Multimedia and Hypermedia (Vancouver, British Columbia, Canada, June 25-30, 1994); see IR 017 359. Figures contain very small print.
PUB TYPE Reports - Descriptive (141) -- Speeches/Conference Papers (150)

EDRS PRICE MF01/PC01 Plus Postage.
DESCRIPTORS *Computer Assisted Instruction; Courseware; Foreign Countries; *Hypermedia; Input Output Analysis
IDENTIFIERS Object Oriented Programming; *Question and Answer Exercises

ABSTRACT

Most educational software lacks either sophisticated computer assisted instruction (CAI) facilities or modern hypermedia functionality. Combining these two aspects not only increases interactivity between computer and learning persons by means of new functions such as moving dragbuttons across the screen and giving multimedia feedback, but also enhances the educational effect. This paper describes how to embed a system for question/answer dialogs into a hypermedia system and points out the resulting advantages. HyperM--a Windows-based hypermedia system, is a powerful PC-based hypermedia system that, among other things, can handle huge amounts of data without loss of performance. Based on ideas of object-oriented structures, the Q/A system consists of a number of objects, each representing a type of input as well as rules on how to judge the student's input. These objects are called interactors because they have the ability to receive user input, judge it, and act accordingly to the results of judging. A Q/A dialog consists of several interactors which are supervised by the control unit. Some points considered important for the development of the Q/A system are: flexible judging algorithms--user input must be examined for correctness by comparing the given answer with a predefined model answer; a need for several learning strategies; the usefulness of question asking algorithms, algorithms that repeatedly ask the same type of question but use different parameters and values; and the use of databases for storing Q/A dialogs, images, sounds, and films. (Contains 20 references.) (Author/MAS)

HyperM: A Hypermedia System with Extended Question/Answer Dialogs

G. HOLWEG
Institute for Information Processing and Computer Supported New Media (IICM), Graz University of Technology, Schieszstattgasse 4a, A-8010 Graz, Austria

P. SAMMER
Institute for HyperMedia Systems (IHM), JOANNEUM RESEARCH, Schieszstattgasse 4a, A-8010 Graz, Austria

Abstract: Most educational software lack either sophisticated CAI-facilities or modern hypermedia functionality. By combining these two aspects we not only increase interactivity between computer and learning persons by means of new functions such as moving dragbuttons across the screen and giving multimedia feedback, but also enhance the educational effect. This paper describes how to embed a system for question/answer dialogs into a hypermedia system and points out the resulting advantages.

During the last decade learning has shifted from rigid system driven courseware to user controlled systems allowing the learning person (in this paper simply called student) to explore the learning material by browsing through and interactively navigating in large databases of information material. Multimedia provides functions to present data in a motivating way stimulating our senses by using different kinds of information material such as films, high resolution digitized photographs and computer graphics, any kind of sound (digitized, MIDI, synthesized speech), or simple text. Hypermedia functions help to organize chunks of information stored in databases by setting links and references. But where is the learning component, someone might ask. With good educational software students should be involved actively, not just simply watching. Learning environments should ask questions and judge the given answers. Many students faithfully believe that they understand what they have just read or heard. But when it comes to applying this knowledge it soon turns out that there is a gap between understanding something and actually applying it. Only when you exercise an operation you can be sure that you have actually understood it. Many popular hypermedia and multimedia systems have been propagated as learning environments, as e.g. HyperCard (Ambron S. & Hooper K., 1990), HyperTies (Shneiderman B. 1989), or Authorware Professional (Authorware 1990) but they cannot often really combine traditional and approved CAI-techniques (CAI stands for Computer Assisted Instruction) with the new features of hypermedia systems such as different kinds of links. The subject of this paper is to illustrate how to build a highly interactive system by merging HyperM - a Windows based hypermedia system - with a powerful system for non intelligent question/answer dialogs (Q/A-dialogs).

HyperM

For several hypermedia projects we needed a powerful PC-based hypermedia system that among other demands should be able to handle huge amounts of data without loss of performance. It should also enable us to add new features.

To this aid, HyperM (Sammer P., 1991) was developed at the IHM. "Images of Austria" is the most famous project running under HyperM. This project was developed in order to present Austria at the EXPO '92 in Seville, Spain, and at the EXPO '93 in Taejon, South Korea in an extended version including Korean text (Maurer H., Sammer P., & Schneider A., 1994). "Images of Austria" consists of some 3000 digitized photographs in high quality which are displayed at remarkably high speed, cartographic material, and digitized film-clips which can be played without additional hardware. Some 18000 textual supplements are available. The

user can select one of the languages English, French, German, Italian, Korean, and Spanish in order to get all the language-sensitive information including the user interface in the desired language.

The user interface can easily be redesigned for specific application to fit needs such as self explanatory and intuitive navigation support (e.g. for going back to the previous scene, scrolling large images, or branching to different parts of the presentation).We did not use all of the numerous features available in HyperM for the EXPO projects in order to obtain the necessary "lack of complication". The database management system (DBMS) for example allows complex database manipulations using SQL (structured query language). As a result of the flexibility of our hypermedia system we could not only incorporate peripheral devices (e.g. starting the presentation triggered by the beam of a photoelectric barrier) but also complex modules such as sophisticated answer judging.

The question/answer system

The Q/A-system has been designed as an independent module which allows the creation and execution of topic independent, highly interactive Q/A-dialogs. This section gives a general view of the Q/A-system and explains how it has been embedded into HyperM. A detailed description of all features can be found in Holweg (1994), implementational details can be looked up in Spinka (1993) and Dick (1994).

A Q/A-dialog is an interactive process where the system presents problems by asking questions and the student solves the problems by answering the questions. Problem solving should be done as intuitively as possible by clicking on buttons, moving graphical objects, entering text, or any other kind of user input you can think of (as for example input from a MIDI-keyboard, data gloves or touch screens). Our system can currently handle input from keyboards and from pointing devices. Each type of user input has its typical characteristics which can be checked (e.g. text input can be compared with a predefined model answer, the movement of graphical objects can be compared with predefined moving paths). Based on ideas of object-oriented structures our Q/A-system consists of a number of objects, each representing a type of input as well as rules how to judge the student's input. These objects are called **interactors** (IAs for short) because they have the ability to receive user input, judge it, and react according to the results of judging. In this way interactivity between system and student is guaranteed. The Q/A-system can be easily extended by adding new interactor types, as for instance an interactor that handles the input of a MIDI-keyboard. However, a single interactor is too restricted in its features to construct complex Q/A-dialogs. Therefore a Q/A-dialog consists of several interactors which are supervised by the **control unit**. The control unit contains a predefined flow structure of a Q/A-dialog coordinating when an interactor can be used by the student and when it is locked (Figure 1 schematically shows how interactors, judging algorithms, reaction parts and control units work together. Figure 3 gives an example of a Q/A-dialog).

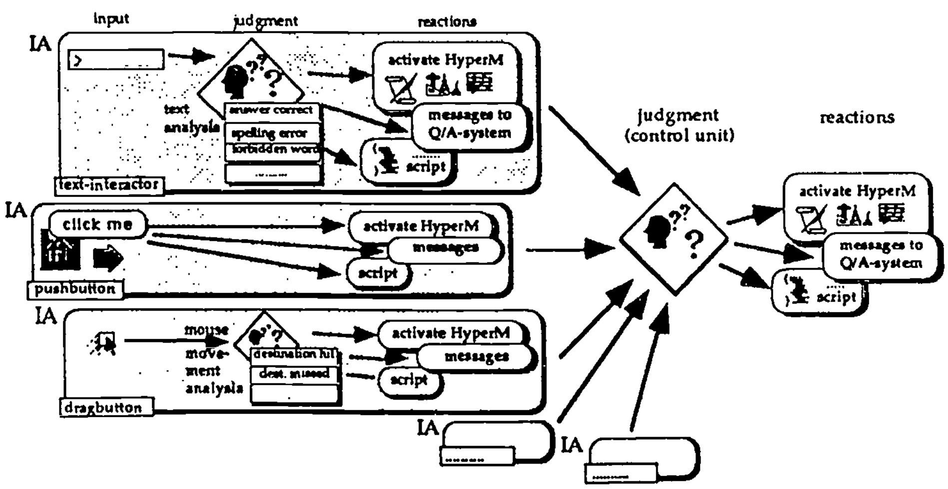

Figure 1: Q/A-dialogs - a cooperation of interactors, judging algorithms, reaction parts and control units.

As already mentioned the activity of an interactor deals with three essential points: user input -> judgment -> reaction. The Q/A-system is designed as a platform independent module that can be easily embedded into different hypermedia systems. In contrast to the judgment section which is implemented as a totally independent module there are parts of the user input and reaction section that are tightly connected to the superior hypermedia system and its user interface. Hence processes such as entering a text-answer or clicking on buttons, and reactions as e.g. displaying a picture on the screen, activating audio signals, or giving a textual response have been removed to the superior system. Figure 2 schematically shows how the Q/A-module is embedded into HyperM. Although as a consequence, this method has some disadvantages (HyperM has to know about at least the input features of the Q/A-module) the advantages are prevailing. The Q/A-system can react in any way by just requesting HyperM to act. The command "*activate subscene name_of_subscene*" directs HyperM to execute a subscene with the specified name. A subscene is similar to a subprogram in programming languages which has access to all features of the language. Thus, the Q/A-system has access to all features of HyperM including all "Hyper-functions". In the following we have listed some of the points which result from a combination of the two systems:

- The judging of a student answer causes HyperM to branch to specified locations and can thus be used to navigate in the learning material.
- Different languages are supported. Text messages and responses can be edited in several languages. HyperM automatically displays the text in the selected language.
- Hypertext features can be used. Words in text messages and responses can be clicked on and links to dictionaries, encyclopedias or other parts of HyperM can be executed. However, authors of Q/A-dialogs have to be careful when formulating questions while this feature is allowed. A question that asks for the birthday of Napoleon might for example encourage the students to look up the answer in an encyclopedia just by clicking on the word Napoleon instead of answering the question themselves.
- Keywords attached to Q/A-dialogs make those dialogs accessible through queries. Thus libraries of Q/A-dialogs of different topics can be collected.

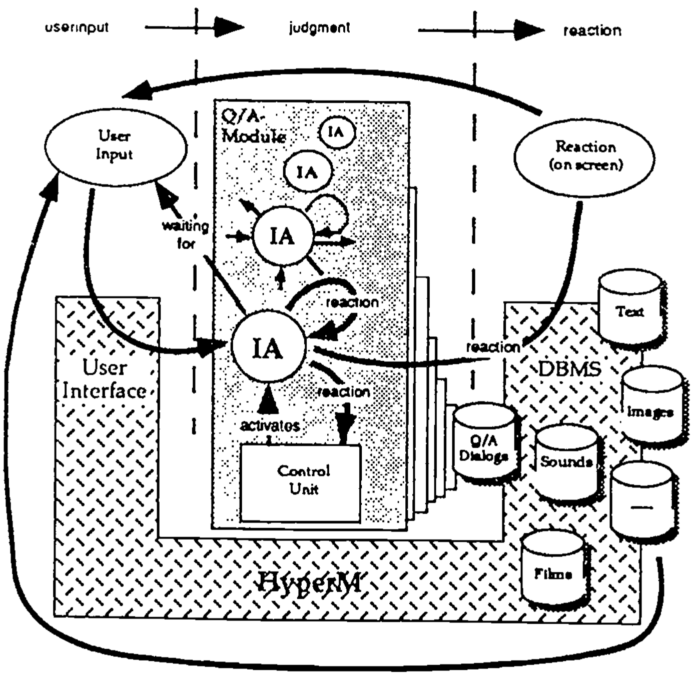

Figure 2:
The Q/A-module is responsible for controlling a Q/A-dialog and judging the user input. User input itself and reactions such as displaying a picture, playing sound or giving text responses are done by the superior hyper-media system.

The Q/A-system currently supports the following interactors: *TextInput*-fields, *PushButtons*, *RadioButtons*, *TouchButtons*, *DragButtons*, *CheckBoxes* and others which do not get their input from the student (*Timers*,

Counters, *Conditionals*). These "other" interactors are important for controlling the internal flow of Q/A-dialogs by performing reactions under special conditions (such as time-outs or counting the number of wrong answers). Interactors can be arranged in any desired order. Figure 3 shows a typical example that extensively makes use of *DragButtons* which can be clicked on and moved across the screen. The left part of this figure shows a sketch of a bicycle brake. After the brake has been displayed to the student, it is erased from the screen and the student has to reassemble the brake by dragging each single part with the mouse to its corresponding position (right part of the figure). All parts are defined as interactors of type *DragButton* and have one or more destinations where they can be moved to (a bolt can for example be used either on the left or the right side of the brake, therefore the bolt has defined more than one destination). The order of assembling is not arbitrary, a nut for example cannot be set before the bolt has been fixed. Thus, the student has to take care of special rules (as they occur in the daily work of a mechanic). These rules have to be specified in the judging section of the *DragButtons* and in the control unit performing the task of supervising the learning person. The following list shows some examples of such rules:

- DRAG aluminum part 1 TO position aluminum 1
- DRAG bolt1 TO (position bolt left OR position bolt right)
- start with aluminum part 1 or aluminum part 2 in any order
- bolt1 must not be moved before aluminum part 1 or aluminum part 2 has been set

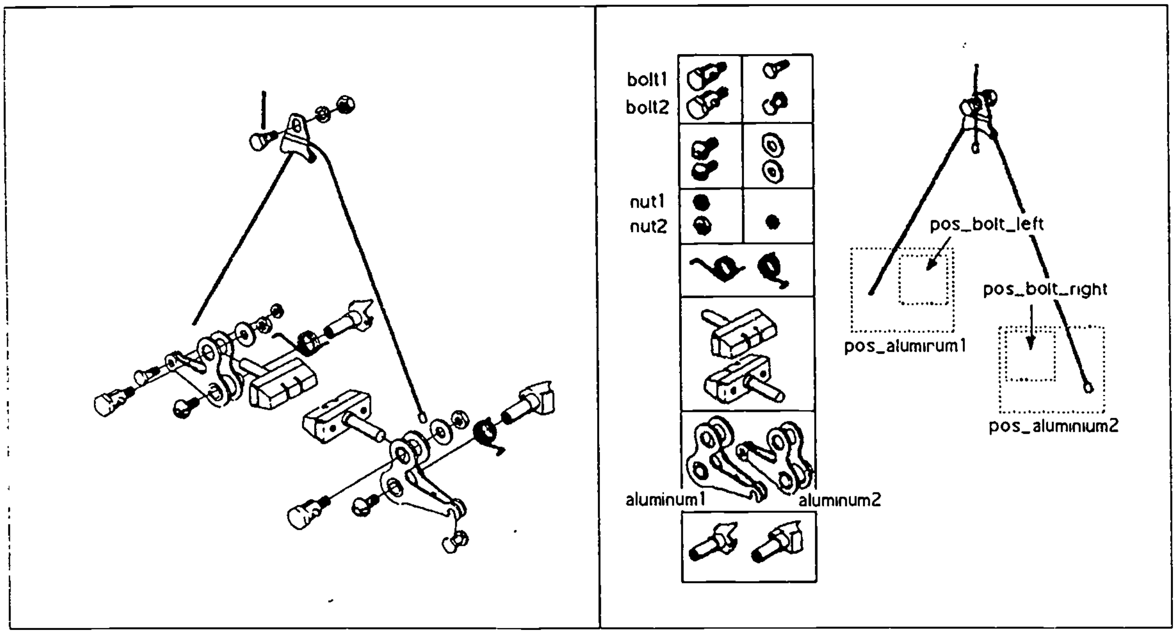

Figure 3: This is an example of a bicycle brake that has to be assembled by snapping each part with the mouse and dragging it to its corresponding position on the right side of this figure.

Components of the Q/A-system

From the technical point of view Q/A-dialogs consist of interactors and control units. But there are other properties that are responsible for a flexible and powerful use of Q/A-dialogs. This section presents some points that were considered to be important for the development of our Q/A-system.

- The power of every Q/A-system stands and falls with the availability of flexible *Judging Algorithms*. The user input must be examined for correctness by comparing the given answer with a predefined model answer. Most interactors typically offer judging algorithms as part of their features. A *TextInput*-interactor for example contains a judging algorithm that checks student answers for spelling errors, synonyms, forbidden words, repeated words, word groups and intervals of numbers, date or time. To find spelling errors we have used the

popular algorithm for the *Weighted Levenshtein Distance* (a short description can be found in Nesbit (1986)) because this algorithm is easy to use and returns a measure for the dissimilarity of the compared words. *DragButtons* use a judging algorithm that watches over the student's movements with the mouse. The learning person has to move the *DragButton* to predefined destinations, touch areas on the screen, follow given paths or stay within a given area. The control unit itself is nothing else than a judging algorithm that supervises the flow of a Q/A-dialog. Because of its modularity the Q/A-module can be easily enhanced by new judging algorithms and existing algorithms can be exchanged by more powerful ones. In this way judging text answers could be managed by language understanding systems based on semantic analysis (which in contrast to the existing method requires language dependent knowledge based databases) (see e.g. Crocker M.W. & Lewin I., 1992). Methods of artificial intelligence as those used in game playing theory (Winston P.H., 1984) could be incorporated into the control unit and thus help to decide if students' actions have to be judged right or wrong.

- When we consider Q/A-dialogs from the pedagogical point of view, we see a need for several *Learning Strategies*. There exists a number of standard types of learning strategies such as multiple choice tests, vocabulary tests, drill&practice and form filling which are well known features in computer assisted learning (see e.g. Merill P.F. & Salisbury D., 1984; Huber F., Maurer H., & Makedon F., 1988; and Nesbit J.C. & Yamamato N., 1991). They are still interesting nowadays especially when they are mixed with graphical features. A multiple choice question for example could be answered by simply clicking on the correct answers. However, they are often regarded as old-fashioned because of their rigid behaviour. In addition to these well known learning strategies it is necessary to offer strategies that have shifted the control from the system to the learning person. The student can experiment with a problem by trying different solutions thus figuring out which one is correct. Most times there is not only one correct solution to a problem, there can be many ways to finish a Q/A-dialog in the right way. The example with the bicycle brakes is typical for that kind of strategy which can be placed somewhere between *Learning by Doing* and *Learning by Trial and Error* (see e.g. the articles of Schank (1993) who presents an interactive multimedia system for learning and Vivet (1991) who deals with those strategies from a pedagogical point of view). New strategies can easily be created by use of the control unit, approved stategies can be stored in databases and accessed whenever needed (see also the following paragraph about databases).

- Sometimes it is useful to have algorithms that repeatedly ask the same type of question but use different parameters and values. We call them *Question Asking Agorithms*. Questions and answers are not exactly defined by the author, but they are established at execution time. This type of Q/A-dialog is known as *adaptive CAI* in literature (see Uhr L., 1969; or Park S. & Tennyson R., 1983 for a survey of early systems). A vocabulary test for example gets words and translations out of a database. The selection of vocabularies depends on special learning strategies such as the *n-piles method* (Stubenrauch R., 1989) which repeats incorrectly answered questions. Linear equations and definite solutions can be calculated at runtime which guarantees alternation since the same question can contain different variables each time it is presented (see Maurer H., Stone M.G., Stubenrauch R., & Gillard P., 1991 for a discussion on the pros ans cons of such algorithms).

- HyperM extensively uses *Databases* for storing all kinds of information such as images, sounds, films and last but not least Q/A-dialogs. This guarantees easy access to the data and makes enlargements simple. Questions can be collected in databases and can then be accessed by HyperM through keywords and queries. Thus it becomes possible to ask HypeM to post all questions about a specific topic, as for example sorting algorithms. The Q/A-system itself uses da.abases. Vocabulary tests for instance get words and translations out of a database. The order of questions depends on question asking algorithms as we mentioned above. Furthermore, databases can be useful for editing Q/A-dialogs. Standard types of learning strategies, control units, predefined interactors and model answers are best qualified for being stored in databases and being accessed when needed.

Conclusion

Most of the components listed above have already been implemented and we are working on the completion. We have tried to combine the advantages of a hypermedia system with the features of a Q/A-system. Information and learning material can be presented in a modern way by displaying all kinds of multimedia data. The user can browse through the presented material by following links and exploring the contents of the presentation. The Q/A-system offers both traditional learning strategies such as drill&practice and form filling, and Q/A-dialogs where the control is partly given to the learning person and the system's job is to watch over the student's activities. The learner should be able to focus on the contents and not the operation of the program, problems should be solvable intuitively, according to the motto: "you don't explain the solution to a problem, you just show it".

References

Ambron S. & Hooper K., (1990). *Learning with Interactive Multimedia*, Microsoft Press.

Authorware (1990). *Authorware Professional Reference Manual*, Authorware Inc., Minneapolis MN.

Crocker M.W. & Lewin I., (1992). *Parsing as Deduction: Rules versus Principles*, Proc. of the ECAI'92, Vienna, pp. 508-512.

Dick H. (1994). *Integration eines Frage/Antwort-Systems in HyperM*, Diplomarbeit, Graz Univ. of Technology, is to appear.

Holweg G. (1994). *Aspekte von Frage/Antwort-Dialogen in Multimedia- und Hypermedia-Systemen*, Dissertation, Graz Univ. of Technology.

Huber F., Maurer H., & Makedon F. (1988). *HyperCOSTOC: A Comprehensive Computer-based Teaching Support System*, IIG Report 258, Graz Univ. of Technology.

Maurer H., Sammer P., & Schneider A. (1994). *Multimedia Systems for the General Public: Experiences at World Expositions and Lessons Learned*, Proc. of IIMS'94, Perth, Australia, pp.333-341.

Maurer H., Stone M.G., Stubenrauch R., & Gillard P.(1991). *Question/Answer Specification in CAL Tutorials (Automatic Problem Generation Does Not Work)*, Didaktik der Menschheit 21, Hölder-Pichler-Tempsky, Wien, B.G.Teubner Stuttgart, pp.191-197.

Merill P.F. & Salisbury D. (1984). *Research on Drill and Practice Strategies*, Journal of Computer-Based Instructions, 11(1), pp. 19-21.

Nesbit J.C. & Yamamato N. (1991). *Sequencing Confusable Items in Paired-Associate Drill*, Journal of Computer-Based Instruction, 18(1), pp. 7-13.

Nesbit J.C. (1986). *The Accuracy of Approximate String Matching Algorithms*, Journal of Computer-Based Instruction, 13(3), pp. 80-83.

Park S.I. & Tennyson R.D. (1983). *Computer Based Instructional Systems for Adaptive Education: a Review*, Contemporary Education Review, 2(2); pp. 77-98.

Sammer P. (1991). *Hypermedia Basis System*, Proc. of 6th Austrian Hungarian Conference on Intelligent Systems CIS`91, pp.281-289.

Schank R.C. (1993). *Learning via Multimedia Computers*, Communications of the ACM, 36(5), pp. 54-56.

Shneiderman B. (1989). *Reflections on Authoring, Editing, and Managing Hypertext*, The Society of Text: Hypertext, Hypermedia, and the Social Construction of Information, Cambridge MA: The MIT Press; pp.115-131.

Spinka O. (1993). *Antwortanalyse in HyperMedia-Systemen*, Diplomarbeit, IICM, Graz Univ. of Technology.

Stubenrauch R. (1989). *Advanced "Non-Intelligent" Question-Answer Dialogs*, IIG Report 268, Graz Univ. of Technology.

Uhr L. (1969). *Teaching Machine Programs that Generate Problems as a Function of Interaction with Students*, Proc. of the 24th National Conference; pp.125-134.

Vivet M. (1991). *Learning Science & Engineering with Open Knowledge Systems*, proceedings CALISCE91, Lausanne Switzerland, pp. 53-62.

Winston P.H., (1984). *Artificial Intelligence*, Addison-Wesley.