Design and Evaluation of Multimedia to Teach Java and Object-Oriented Software Engineering*

Glenn D. Blank, William M. Pottenger, G. Drew Kessler, Soma Roy, David R. Gevry, Jeffery J. Heigl, Shreeram A. Sahasrabudhe and Qiang Wang

Department of Computer Science and Engineering
Lehigh University, Bethlehem, PA 18015

Abstract

CIMEL is a multimedia framework for Constructive and collaborative, Inquiry-based E-Learning supplementing computer science courses. *Constructive* learning goes beyond learning by receiving knowledge, to learning by building systems, with immediate, visual feedback. *Collaborative* learning encourages students to interact with instructors and librarians via live links and remote-controlled “show me” sessions and by reviewing multimedia FAQs of recorded “show me” sessions. *Inquiry-based* learning guides the student into pursuing exploratory research in a community of students and scholars. A text mining and visualization tool enables students to identify and explore emerging technology trends in computer science as part of our inquiry-based framework. Our project documents, evaluation materials and a prototype are available at www.cse.lehigh.edu/~cimel.

1.0 Introduction

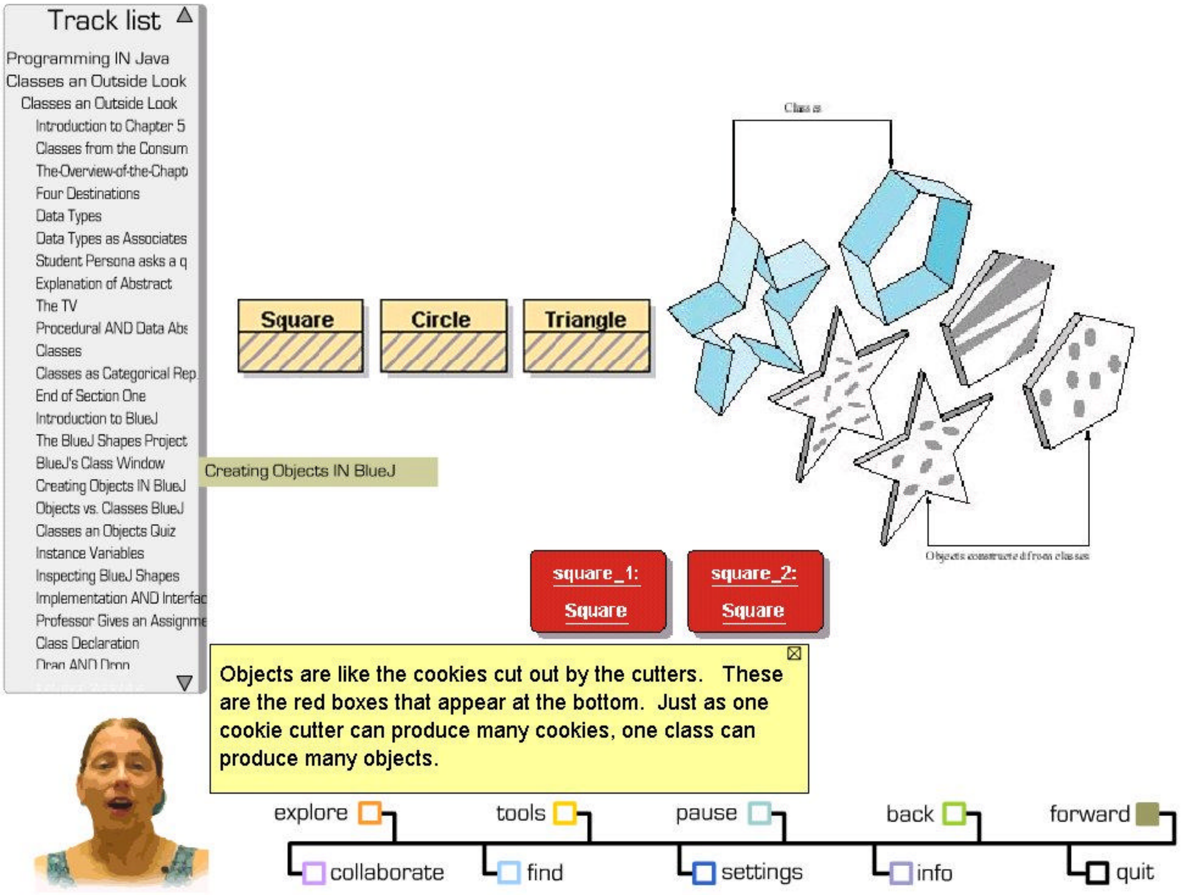
CIMEL is a multimedia framework for Constructive and collaborative, Inquiry-based E-Learning supplementing computer science courses. Within the CIMEL multimedia framework, we have developed alpha versions of new materials for two courses in computer science at quite different levels: a graduate level course in Object-Oriented Software Engineering (OOSE) and a first semester course in computer science (CS0/1) (CS0 for non-majors and CS1 for potential minors and majors). New multimedia modules, implemented in Macromedia Flash and played through high-speed connections on the web, feature audio narration, animation, simulations, quizzes, and constructive exercises.

For the OOSE course, we developed a multimedia unit on Abstract Data Types (ADTs), as a way to formalize the meaning of classes in connection with object-oriented design. In the past, graduate students have found it difficult to master this material from lecture and textbooks alone.

*This project is funded by National Science Foundation (Grant Number EIA-0087977)

Our premise is that multimedia will help students understand the material better, objectively, and also improve their design of actual ADTs to solve a problem.

For CS0/1, we began revising both the manuscript and multimedia of *The Universal Machine: A Multimedia Introduction to Computing* (McGraw-Hill 1998), tentatively re-titled *The Universal Computer: A Multimedia Introduction to Computer Science and Problem Solving in Java* (Blank, Barnes and Kay 2001). As the new title suggests, the new material introduces Java (instead of C++). A first draft of the new textbook was completed during the summer of 2001 and an alpha version of the multimedia was completed in time for use in the fall of 2001. During the fall semester, students in the Introduction to Computing course at Lehigh University began using the new material, with five of the six proposed chapters introducing Java programming and all ten other chapters on problem solving, software engineering and the breadth of computer science.



Objects are like the cookies cut out by the cutters. These are the red boxes that appear at the bottom. Just as one cookie cutter can produce many cookies, one class can produce many objects.

Figure 1: Screen Capture from *The Universal Computer*

Figure 1 illustrates several features of the CIMEL multimedia framework:

- Multimedia personae model a community of learners and instructors. The personae include a professor, teaching assistant (shown here), a reference librarian, and two students. In addition to graphical images, they speak in audio and/or text boxes. These personae model students and teachers studying material together, working through interactive and constructive exercises, and suggest exploratory research on relevant topics using online information.

- A tracks list displays the content of a lesson as a sequence of screens. Course designers, instructors and students will be able to tailor different tracks through the content.
- A user interface gives the user access to various tools, including the "explore" (emerging trends) and "collaborate" tools under development for this project, as well as the BlueJ and JavaEdit programming environments developed elsewhere. An ActiveX control lets the Flash page access these tools securely.

For both courses, we conducted experiments, surveys and focus groups to determine whether the alpha version of our multimedia actually improves learning, and to determine what improvements should be made for our beta version. We found significant learning effects in the more focused study in the OOSE course and marginal results in the broader study in the CS0/CS1 course.

In addition to our studies on the multimedia course material, we are developing an inquiry-based learning framework. In the CIMEL system, we are integrating a text mining and visualization framework for the detection of incipient emerging trends in order to facilitate inquiry-based learning. An emerging trend is a topic area for which one can trace the growth of interest and utility over time. An example of such a trend is XML, a technology that emerged in the mid 1990's. This framework allows the exploration of technology fields closely related to the course work. The detection of emerging trends in the course domain stimulates inquiry-based learning by providing an avenue of research into key developments in these related fields.

Our initial research into methods for detecting incipient emerging trends is presented in this article. For the OOSE course the students participated in a learning exercise involving emerging trend detection. The students were split into two groups of roughly equal numbers for this exercise. Both groups attempted an exercise that involved identification of three emerging trends in the area of Design Patterns of Object Oriented Software Engineering. However, only Group B received a methodology for the algorithmic identification of an emerging trend. For this approach we demonstrate with a confidence level of 99% that this approach results in a significant improvement in the precision of trend detection.

The CIMEL system will include additional collaborative tools that will provide a seamless network connection between students, instructors, teaching assistants and librarians. Through both live links, remote-controlled "show me" sessions, and through a multimedia FAQ of recorded "show me" sessions, this collaborative interface will encourage students to interact with instructors and TA's to obtain help, and allows students to more readily work with others in a project group. Our hypothesis is that collaborative tools can provide a technology infrastructure for students to increase and simplify their contact to the people that are their resources for the course material, and with other students for after-class projects, such that they improve their performance in these classes.

In our design, the collaborative tools include an instant messenger, a conference chat system, a shared white board, a "show me" remote demonstration system, two-way video and shared audio. We present, in what follows, our initial implementation, and an evaluation of that implementation and our overall design.

2.0 Motivation

A major goal of the CIMEL project is to provide opportunities for inquiry-based learning, i.e., learning by doing research in a community of students and scholars. For example, after the multimedia has introduced the String and StringBuffer classes, it asks the student to find out what methods can remove one or more characters. This exercise challenges the student to begin exploring the JDK documentation to find solutions to problems. On this page, the "explore" button opens a browser pointing at a local copy of the JDK documentation. If the student isn't sure what to do next, the "collaborate" button can take the student to the people behind the personae, such as a professor or teaching assistant associated with the course, in order to chat or open a remote-controlled "show me" session. Or the student could just look in the "collaborate" archives for an item in which the professor has already explained how to search for classes and methods in the JDK. For a more ambitious example, after learning about design patterns, the professor persona may challenge the student to investigate emerging trends in the literature on this topic, identifying at least two, with a short annotated bibliography. On this page, the "explore" button starts up the "emerging trends" tool, which guides the student through a method of identifying trends from online textual databases. Again, if the student isn't sure what to do next, the "collaborate" button lets the student ask the reference librarian how to proceed, or find a relevant "show me" in the archives.

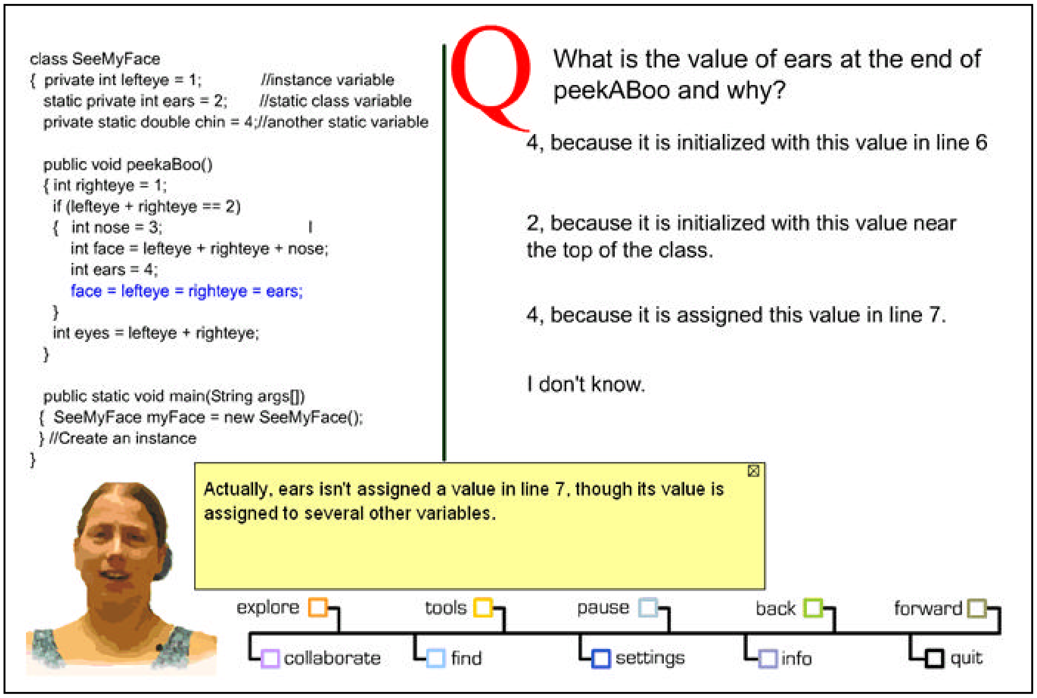
That's our vision. To get there from here, we first created a prototype, then an alpha version of a user interface, plus the multimedia content that is prerequisite to inquiry-based exercises. We have evaluated the alpha version in two different classes at Lehigh. Experimental evaluation, including pretests, posttests, demographics surveys, user surveys and focus group meetings, drives our development process. Our results have shown us what improvements we need to make for the beta version, which we will outline in the future work section of this paper.

In addition, we have developed use case scenarios that flesh out our initial sketch at how the emerging trends and collaborate tools will work and be integrated into our multimedia framework.

3.0 Approach

Interactive and Constructive Exercises

Interactivity is a key aspect of the CIMEL content. Interactive quizzes and constructive exercises help students learn by doing. Personae provide feedback guiding a student through each exercise. The following two figures illustrate a typical multiple-choice question, designed to reinforce important subject matter. In figure 2, upon selection of the wrong choice, the face of the Teaching Assistant persona changes from talking to puzzled, and supplies feedback in both text and voice media. The student gets another chance to answer correctly.



```
class SeeMyFace
{ private int lefteye = 1;          //instance variable
  static private int ears = 2;      //static class variable
  private static double chin = 4;//another static variable

  public void peekaBoo()
  { int righteye = 1;
    if (lefteye + righteye == 2)
    {  int nose = 3;
       int face = lefteye + righteye + nose;
       int ears = 4;
       face = lefteye = righteye = ears;
    }
    int eyes = lefteye + righteye;
  }

  public static void main(String args[])
 {  SeeMyFace myFace = new SeeMyFace();
 } //Create an instance
}
```

Q What is the value of ears at the end of peekABoo and why?

4, because it is initialized with this value in line 6

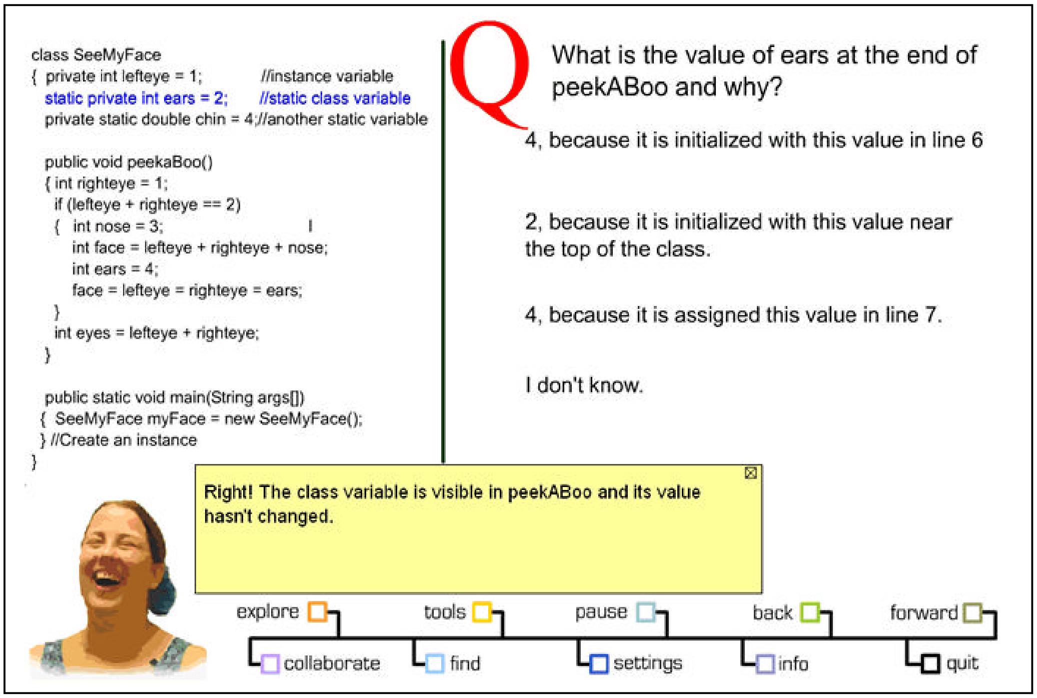
2, because it is initialized with this value near the top of the class.

4, because it is assigned this value in line 7.

I don't know.

Actually, ears isn't assigned a value in line 7, though its value is assigned to several other variables.

Figure 2: TA persona responds to a wrong choice in a multiple-choice question



```
class SeeMyFace
{ private int lefteye = 1;          //instance variable
  static private int ears = 2;      //static class variable
  private static double chin = 4;//another static variable

  public void peekaBoo()
  { int righteye = 1;
    if (lefteye + righteye == 2)
    {  int nose = 3;
       int face = lefteye + righteye + nose;
       int ears = 4;
       face = lefteye = righteye = ears;
    }
    int eyes = lefteye + righteye;
  }

  public static void main(String args[])
 {  SeeMyFace myFace = new SeeMyFace();
 } //Create an instance
}
```

Q What is the value of ears at the end of peekABoo and why?

4, because it is initialized with this value in line 6

2, because it is initialized with this value near the top of the class.

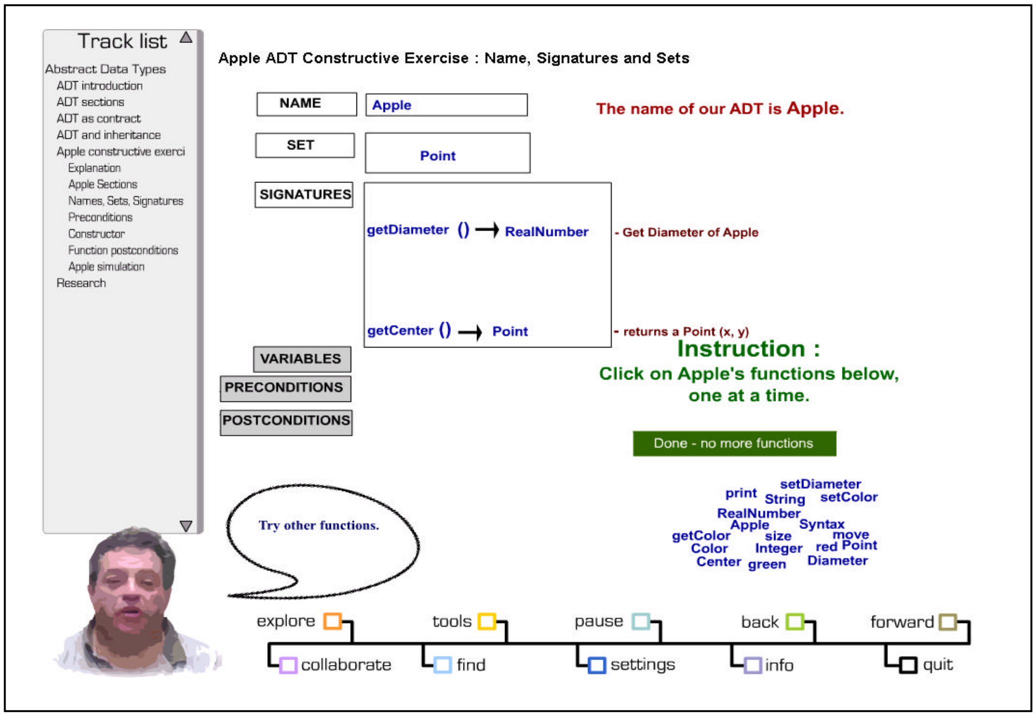
4, because it is assigned this value in line 7.

I don't know.

Right! The class variable is visible in peekABoo and its value hasn't changed.

Figure 3: TA persona responds to a correct choice in a multiple-choice question

When the user chooses a correct answer (figure 3), the persona provides positive reinforcement and feedback explaining the correct answer. Constructive exercises, on the other hand, are much more complex, challenging the learner to build solutions to problems by manipulating structures.



Apple ADT Constructive Exercise : Name, Signatures and Sets

NAME: Apple

The name of our ADT is Apple.

SET: Point

SIGNATURES:

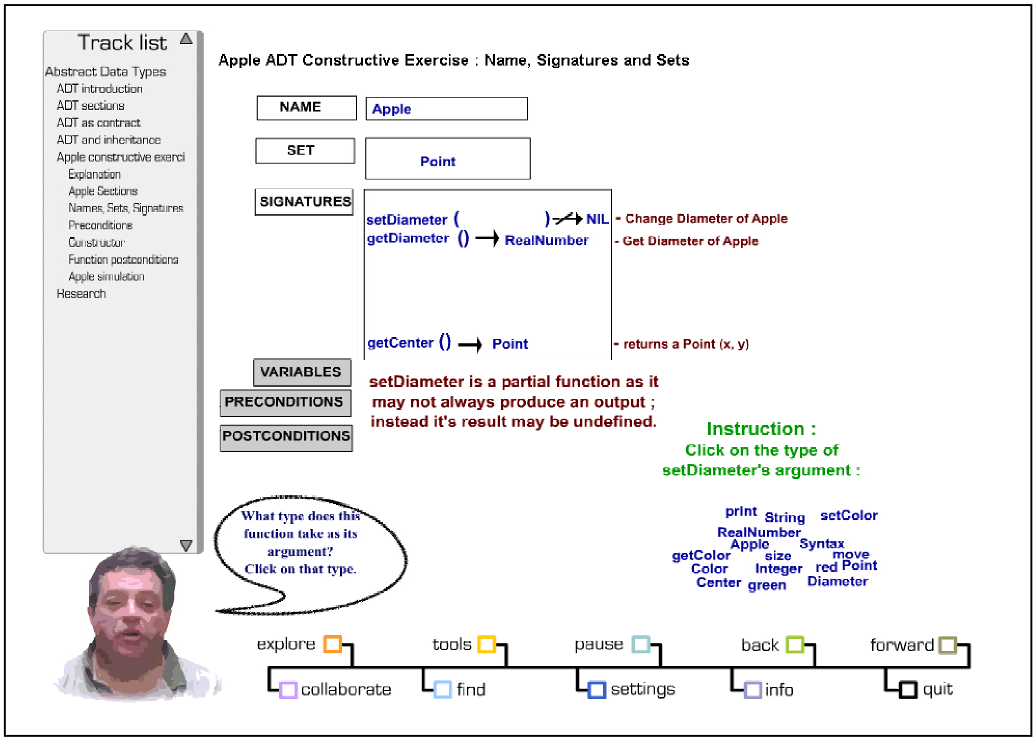
getDiameter () → RealNumber - Get Diameter of Apple

getCenter () → Point - returns a Point (x, y)

Instruction :
Click on Apple's functions below, one at a time.

Try other functions.

Figure 4: Excerpt from a constructive exercise. The learner chooses a function for Apple ADT.



Apple ADT Constructive Exercise : Name, Signatures and Sets

NAME: Apple

SET: Point

SIGNATURES:

setDiameter ( ) ↛ NIL - Change Diameter of Apple

getDiameter () → RealNumber - Get Diameter of Apple

getCenter () → Point - returns a Point (x, y)

setDiameter is a partial function as it may not always produce an output ; instead it's result may be undefined.

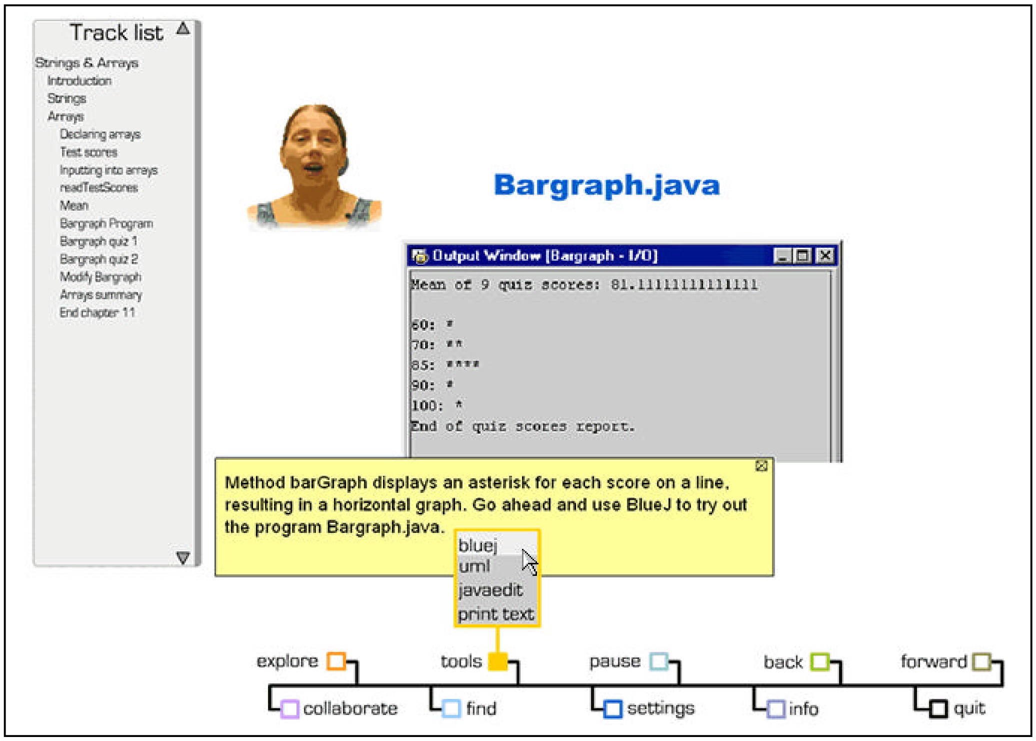
Instruction :
Click on the type of setDiameter's argument :

What type does this function take as its argument? Click on that type.

Figure 5: Now the learner must choose the function's argument for Apple's ADT signatures.

The figures above show an excerpt from a constructive exercise, requiring a high level of user interaction. Previous screens have presented the components of ADTs, interspersed with simpler interactive questions, and culminating with the simulation of an ADT for Employee. Finally, in this constructive exercise, the learner goes through a series of screens, incrementally building up an ADT specification for class Apple. In figure 4, the learner constructs the signatures section of Apple ADT by first choosing a member function, then in figure 5 selecting that function's arguments. Later on in the exercise, the learner builds the preconditions and postconditions for Apple ADT (as a drag and drop exercise). Finally, the learner runs simulations running the functions, preconditions and postconditions, testing them for completeness. At each step, feedback helps the learner learn from mistakes as well as correct actions.

Another type of constructive exercise involves the review, modification and development of actual programs in a programming environment. With the help of a specially designed ActiveX control, the user interface lets the user invoke either of two different available programming environments: BlueJ (a Java programming environment designed for beginners and freely available from www.bluej.org) or JavaEdit (Dick Chase's free, relatively small and easy to use tool). In figure 6, after presenting material on using arrays to generate bar graphs, the TA asks the learner to try out a program. On this screen, the BlueJ and JavaEdit buttons have been overloaded to specifically load BarGraph.java into the corresponding IDEs. After the learner has experimented inside a programming environment, subsequent screens may ask questions about the material, then challenge the learner to make changes to the program or create a new program similar to this one. The CIMEL alpha version so far incorporates over forty sample Java programs grouped in six different chapters.



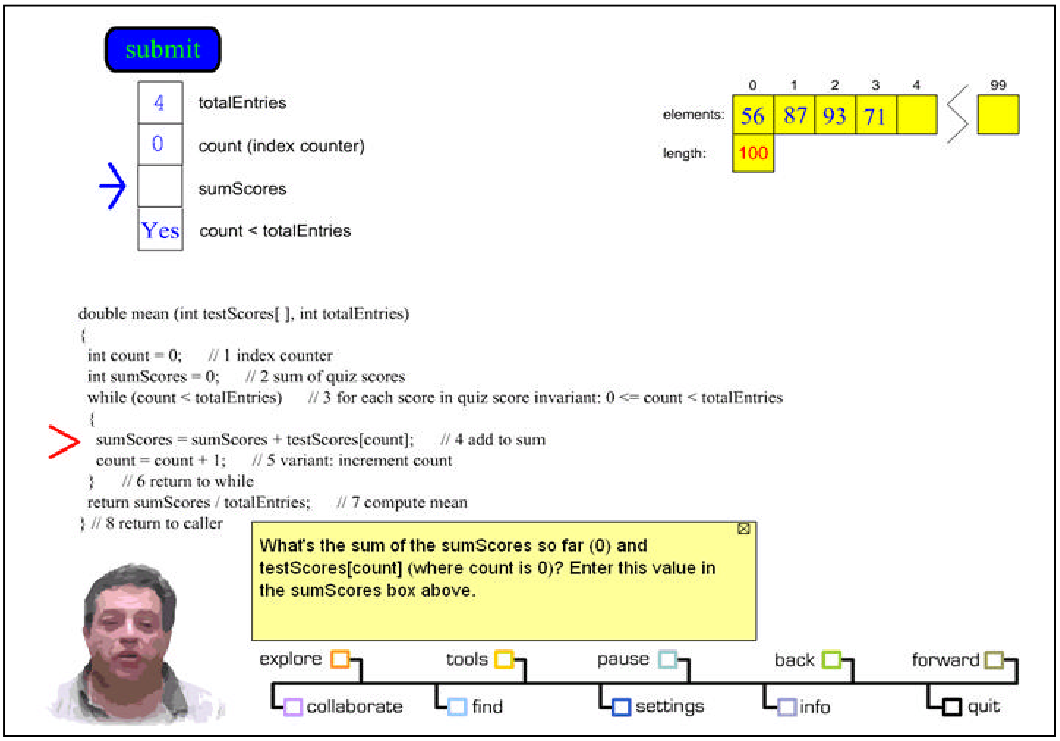
```
Mean of 9 quiz scores: 81.11111111111111

60: *
70: **
85: ****
90: *
100: *
End of quiz scores report.
```

Method barGraph displays an asterisk for each score on a line, resulting in a horizontal graph. Go ahead and use BlueJ to try out the program Bargraph.java.

Figure 6: Using the use of the tools menu to access a Java program

Beneath the multimedia content lies the CIMEL tracking system, which records nearly every interactive action in a database. In Flash Actionscript, getURL actions, associated with each user interface button as well as each exercise response, send data to the CIMEL tracking datab ase. Detailed tracking information helps our outside evaluators study how learners with different backgrounds and learning styles actually use the system.

The following example illustrates this important feature of the CIMEL multimedia environment. Figure 7 shows a constructive exercise in which the student traces array values in a loop, step by step. In the following iterations the student is asked to enter the new values of the variables, requiring him or her to pay close attention to the lesson.



4 totalEntries

0 count (index counter)

sumScores

Yes count < totalEntries

```
double mean (int testScores[ ], int totalEntries)
{
 int count = 0;     // 1 index counter
 int sumScores = 0;     // 2 sum of quiz scores
 while (count < totalEntries)     // 3 for each score in quiz score invariant: 0 <= count < totalEntries
  {
   sumScores = sumScores + testScores[count];     // 4 add to sum
   count = count + 1;     // 5 variant: increment count
  }     // 6 return to while
 return sumScores / totalEntries;     // 7 compute mean
} // 8 return to caller
```

What's the sum of the sumScores so far (0) and testScores[count] (where count is 0)? Enter this value in the sumScores box above.

Figure 7: A constructive exercise reviewing how array values are used in a loop

Whenever the student enters a wrong answer, he or she is given another chance, but alongside, the CIMEL tracking system records that event. An instructor can zero in on a student's weakness by looking in the database for the number of attempts before the student answered correctly. CIMEL developers use the database to zero in on potential usability problems with the system.

The tracking system also records interactions with the user interface buttons. This provides evaluators and developers with information about settings preferences (sound on/off, text display on/off, etc.), what tools they have accessed and even how long a user took to view each lesson. Each user's preferences are maintained between CIMEL sessions, allowing the user to customize the multimedia environment to their own personal taste.

The exercises sampled above highlight the range and depth of CIMEL material. Since we are designing material for a students ranging from first year to graduate students, we are creating a dynamic menu interface, shown in Figure 8, that will let instructors and students manipulate what gets presented.

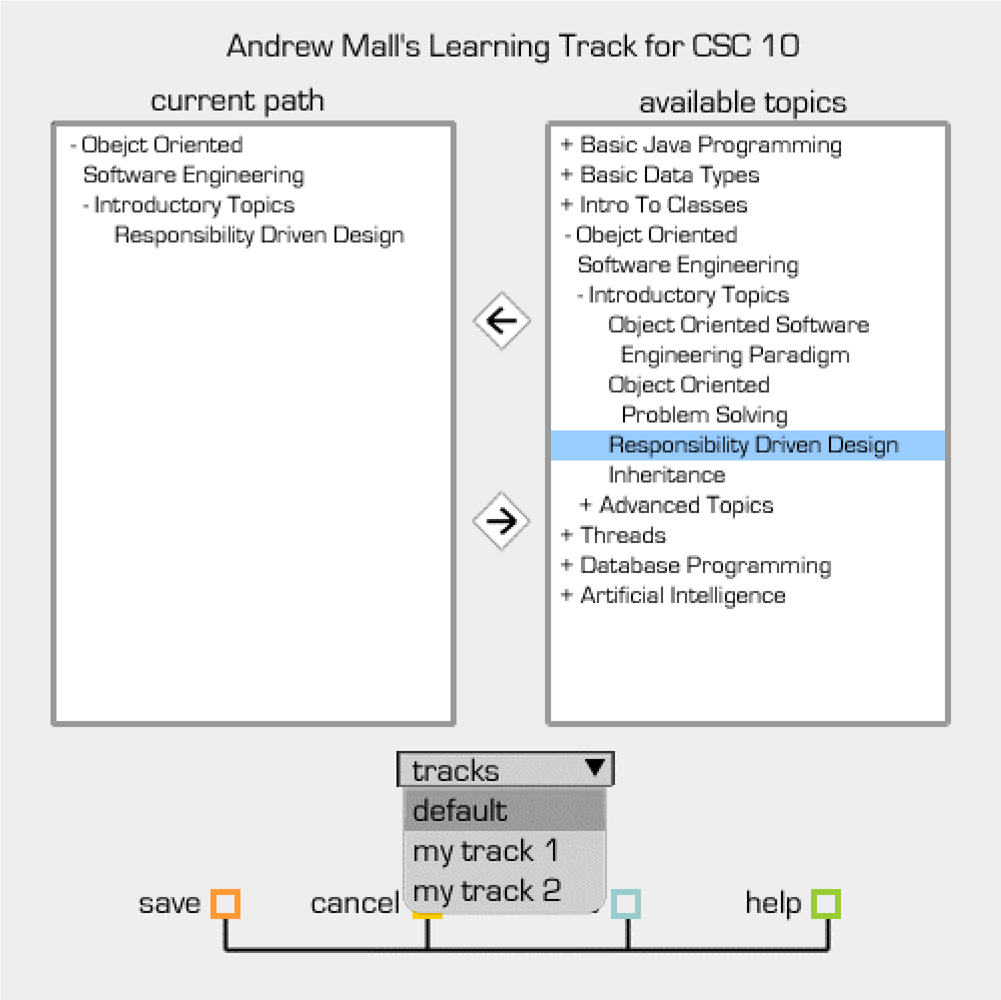


Figure 8: Selecting menu and modifying their content

With the dynamic tracks interface, instructors and students can customize different tracks or paths through the available material. The multimedia material is organized hierarchically, divided up into chapters, which are then divided into sections, and further divided into screens. CIMEL designers, instructors and students can customize default tracks by adding and removing the desired chapters, sections and screens. CIMEL will provide at least three different sample tracks, for use in CS0, CS1 and an upper level OOSE course. An instructor can then tailor material for a particular class, by deleting or adding screens, sections or whole chapters. A student can modify these defaults, supplied by the CIMEL designers or other instructors, for their own needs. For example, a student in CS0 may choose to go into more depth than deemed necessary by a designer or instructor, or a student in an upper level course may select additional introductory material for review.

Inquiry Based Learning through the Detection of Incipient Emerging Trends

One of the goals of the CIMEL framework is to offer students ways to go beyond the knowledge presented in the course work, by exploring current research trends. We believe that providing students with opportunities to explore the research literature related to a course will give them a better appreciation for the fundamentals presented in the courses, as well as a perspective of where technology is headed. Learning how to do research is itself a valuable skill for lifelong learning. We are developing an inquiry-based learning module that guides the student through the process of detecting incipient emerging trends in key topic areas related to course material. Like the constructive exercises, this module will provide the students an opportunity to pursue course related topics in a more hands-on manner. Through the detection of incipient emerging trends the students will see the role that current topics play in course related research areas.

We have completed the first steps in this effort with the development and evaluation of a methodology for detecting incipient emerging trends. Detection of emerging trends starts with the selection of a main topic area, such as inheritance in object oriented programming. The use of domain knowledge at various stages of identification of emerging trends is key. The method begins by searching recent conference and workshop proceedings for discussion of the main topic area, giving special attention to workshop websites and technical papers for possible emerging trends (e.g., topics within the domain of the main topic area "inheritance"). A web search engine (such as Google or Yahoo) is used to find additional trends and further evidence of references to the candidate trends. There are two possible scenarios: 1) If the conference and workshop search identifies any candidate emerging trends, then the student performs a web search with the combined query of the candidate trend and a helper term such as recent research, novel, emerging trend, etc. Otherwise, 2) the web search uses a combined query of the main topic area and any of the predefined helper terms. After choosing one of these queries, one follows the algorithm in Figure 9 to identify candidate emerging trends[†]. This algorithm identifies trends as either incipient or non-emerging. For further verification, the method performs a database search of research abstracts, using the combined query of the main topic area and a newly found candidate emerging trend, from the year of origin of the main topic area to the current year. If the frequency of documents referencing the search terms increases over the years, the candidate emerging trend is confirmed as a bona fide trend with respect to the main topic. In the next section, we will provide a case study to flesh out this methodology.

In our next release, a tutorial will introduce students to this methodology. This tutorial will guide the students through the process of trend detection in the main topic area of "inheritance" and explain the usage of the trend detection interface as well as the algorithmic approach.

Our goal is to provide the students with an easy to use interface that will step them through the trend detection methodology as well as an implementation that will perform the database search and create a summary of key statistics for each candidate emerging trend. The module will search a repository generated from online sources of research abstracts in the area of

[†] Note that if the first scenario applies, then "main topic area" should be read as "candidate emerging trend" in the algorithm depicted in Figure 9.

```
While (there are links to follow from the search engine retrieved pages <or> the desired number of trends has not been found) {
    Make an empty list L2. // use this to store all candidate emerging trends.
    Click on link = 1 // first link of interest in the search results
    If (year page last modified == current year or (current year -1) or (current year –2)) {
        Do while a page of interest is found {
            Make a list L1:
                number of occurrences of  the term "main topic area" in the page : "main topic area" : m
                number of occurrences of the helper words used to do the search in the   page : <helper term> : n
                If (m, n > 1) {
                    The page is of interest.

                    Add "main topic area" to L2 if it is a candidate emerging trend.
                    List the frequency of occurrences of all the phrases in the page.
                    Look for the phrases (except general phrases) with the highest frequency of occurrence.
                    Add them to L2 if they qualify as candidate emerging trends (use domain knowledge).
                    Give special attention to the line (or paragraph) containing the word set : main topic area <and> helper  term.
                    Add to the list L2, phrases appearing in that paragraph (or sentence) that are judged to be candidate emerging
                        trends (use domain knowledge).
                }
                Else {
                    If  (m > 1 && (find any other helper term in the page)) {
                        The page is of interest.
                        Add "main topic area" to L2 if it is a candidate emerging trend.
                        List the frequency of occurrences of all the phrases in the page.
                        Look for the phrases (except general phrases) with the highest frequency of occurrence.
                        Add them to L2 if they qualify as candidate emerging trends (use domain knowledge).
                        Give special attention to the line (or paragraph) containing the word set : main topic area <and> helper  term.
                        Add to the list L2, phrases appearing in that paragraph (or sentence) that are judged to be candidate emerging
                            trends (use domain knowledge).
                    }
                    Else {
                        Add "main topic area" to L2 if it is a candidate emerging trend.
                        List the frequency of occurrences of all the phrases in the page.
                        Look for the phrases (except general phrases) with the highest frequency of occurrence.
                        Add them to L2 if they qualify as candidate emerging trends (use domain knowledge).
                        Give special attention to the line (or paragraph) containing the word set : main topic area <and> helper  term.
                        Add to the list L2, phrases appearing in that paragraph (or sentence) that are judged to be as candidate emerging
                            trends (use domain knowledge).

                        If (found candidate emerging trend)
                            The page is of interest
                        Else Reject page.
                    }
                }
        }  // Close Do While loop
    } // Close If year
    Else {
        Perform an INSPEC database search to confirm that term is not emerging. Reject page.
        // An INSPEC database search should show an increasing number of documents referencing the term over the years
        //   if the candidate is truly an emerging trend. If not, it is not emerging.
    }
    Click on link++
    // click on next link of interest
} // Close outermost While loop
```

Figure 9: Algorithm for use in detecting emerging trends

“inheritance,” and later be expanded to other topic areas. Currently we are determining metrics to use as visualization cues to guide students in their detection of emerging trends. An example is a bar chart showing the increasing spread of authorship across time for a given trend. These metrics will be displayed after the module performs the database search. The module will run as a tool accessible from the “explore” button of the CIMEL interface and provide students with links to course related conference sites, search engines, and helpful hints and explanations of the trend detection process. Additionally, the module will provide a query mechanism to access a repository of research abstracts on various main topic areas. We plan to conduct an evaluation of the incipient emerging trend detection inquiry-based learning module in a Programming Languages course this spring using the beta version of the CIMEL multimedia framework.

Use Case Study of the Incipient Emerging Trend Detection Methodology

In this section, we present a use case illustrating the above methodology. For this example, the main topic area is chosen to be “inheritance” in object oriented programming.

i. Selection and validation of main topic area.

First following step 1 of the methodology, a main topic area is chosen which in this case is *inheritance*. Following this selection an INSPEC database search is performed to verify the selected topic area for its potential to contain emerging trends.

1981	**1982**	**1983**	**1984**	**1985**	**1986**	**1987**
1	1	2	4	5	33	50
1988	**1989**	**1990**	**1991**	**1992**	**1993**	**1994**
83	123	173	179	227	272	269
1995	**1996**	**1997**	**1998**	**1999**	**2000**	**2001**
253	267	217	265	176	136	75

INSPEC database search: “Inheritance” and “Object-Oriented”

The above table shows the document counts by year from an INSPEC query on Inheritance and Object-Oriented. The coverage of this topic over recent years suggests that new innovative enhancements to *inheritance* are being researched. Therefore *inheritance* is a valid main topic area for the identification of emerging trends.

ii. Search for candidate emerging trends.

The next step is to explore workshops and conferences related to the discussion of this main topic area. For this example we chose to show the second scenario of the methodology, where candidate trends are discovered in a main topic area through queries to a web search engine.

iii. Candidate emerging trend verification.

A search engine (e.g., Google or Yahoo) is used to find trends. Using the algorithm in Figure 9 various web searches were performed to identify candidate emerging trends. The first step in this process was a Google search was performed using the helper term “a new paradigm” <and> “inheritance”, which resulted in the URL:

www.cs.uni-essen.de/…/shaUnl_ConcerningAOPandInheritance_2001.pdf

Pertinent excerpts from this link included:

> *Aspect-Oriented Programming (AOP) has recently been proposed as a* ***new paradigm*** *for software development.*

A candidate emerging trend has been discovered in *Aspect Oriented Programming*, so now we will proceed with the first scenario of the methodology and perform a Google search using "inheritance" <and> "aspect oriented programming". Two of the resulting hits were:

http://ecoop2000.unice.fr/Program/Technical/Posters/p8.html
http://www.sdmagazine.com/documents/s=1123/sdm0109e/0109e.htm

Pertinent excerpts from these two links included:

> *Revisiting Separation of Concerns with an Aspect-Oriented Framework*
> Presenter: Constantinides Constantinos (Department of Computer Science, Illinois Institute of Technology, USA) …
> Further advantages include the avoidance of inheritance anomalies …
> These cross cutting properties have been termed aspects …
> Aspect-Oriented Programming (AOP) is an emerging paradigm that addresses components and aspects …
>
> "Aspect-oriented programming (AOP) is a new programming methodology that enables the modularization of crosscutting concerns." …
> Like inheritance, to be interesting, an aspect has to involve at least three system components: a caller, a receiver, and another, less obvious, participant. In inheritance, a call (or access to a field) may be handled by an "ancestor" object. In aspect-oriented programming, a call or access can be intercepted by an aspect. …

The above excerpts are outputs from the algorithm in Figure 9. Upon completion, *Aspect Oriented Programming* is verified as a candidate emerging trend in the area of inheritance.

iv. Verification of Algorithmic Results

An INSPEC Database search using “Aspect Oriented Programming” and “Inheritance” was performed to verify the algorithmic results.

1981	**1982**	**1983**	**1984**	**1985**	**1986**	**1987**
0	0	0	0	0	0	0
1988	**1989**	**1990**	**1991**	**1992**	**1993**	**1994**
0	0	0	0	0	0	0
1995	**1996**	**1997**	**1998**	**1999**	**2000**	**2001**
0	0	0	1	1	0	1

INSPEC database search: “Aspect Oriented Programming” and “Inheritance”

The frequency of documents referencing the search term has recently increased. As a result, we conclude that *Aspect Oriented Programming* is indeed an incipient emerging trend with respect to Inheritance. We will present the results of a formal, statistical evaluation of the above methodology for emerging trends detection in the results section later in this article.

Collaborative Networking

The software architecture of collaborative networking in CIMEL[1] is depicted in Figure 10.

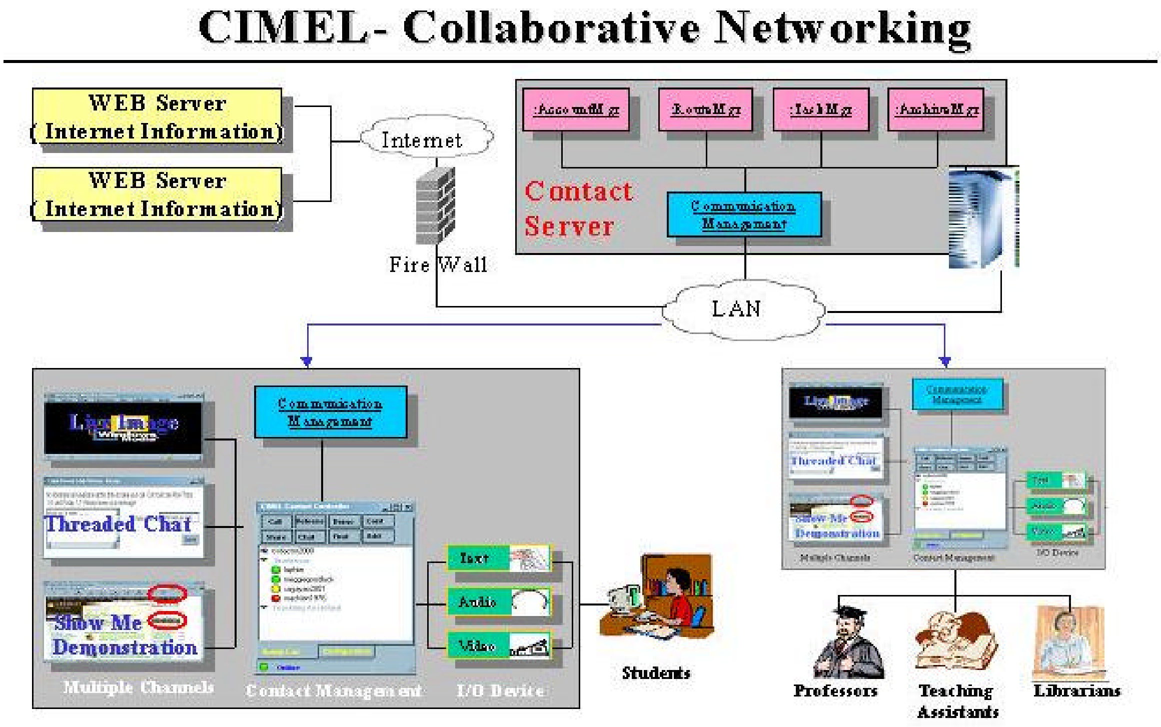
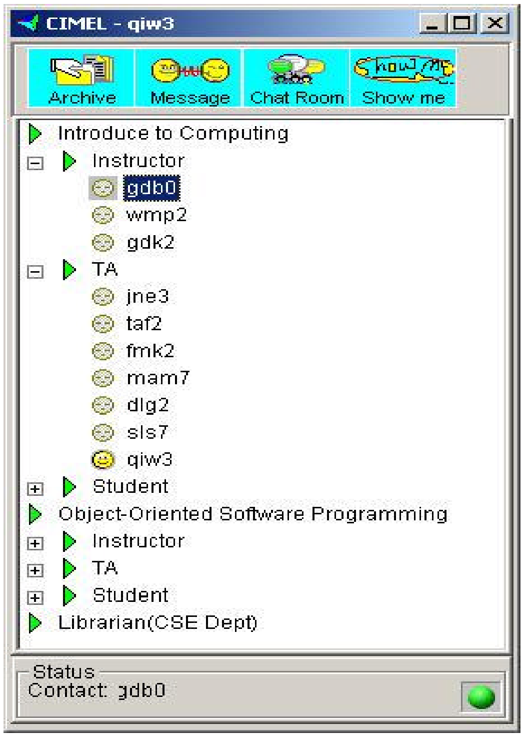


Figure 10: Software Architecture of CIMEL Collaborative Networking

For the students in the OOSE course, we developed instant messenger and collaborative chat tools. In order to support high bandwidth information exchanges between participants, the framework for our collaborative tools uses a peer-to-peer communication scheme instead of the traditional client-server model. A server is still used, but it only maintains the global state information (e.g., availability) of participants. The communication between participants is handled by direct, peer-to-peer connections over IP. Each peer in this configuration has a communication component and a set of upper-level collaboration tools. The communication component communicates with the collaboration tools through shared memory, and acts as a communication broker. This design provides a simple interface to collaboration tools for communication management, maintaining the IP connection list among peers, and the First-In-First-Out queue of messages from other peers.

The first step in using the collaboration tools is identifying the parties with which one wishes to communicate. The system provides a list of users that may be contacted, organized by their roles. Particular roles are assigned to users when they open an account. One user could be an instructor, student, teaching assistant, librarian, or some combination of these roles, e.g., one user can be a teaching assistant in course A and a student in course B. Users are managed in an organization tree according to their roles. This organization tree is similar to a "buddy list" in standard instant messenger systems (e.g., Yahoo Messenger, MSN Messenger); however, these roles are pre-defined to reflect a given user's participation and skill. We plan to use roles to let users to contact group members without seeking a particular person, e.g., to find an available librarian.

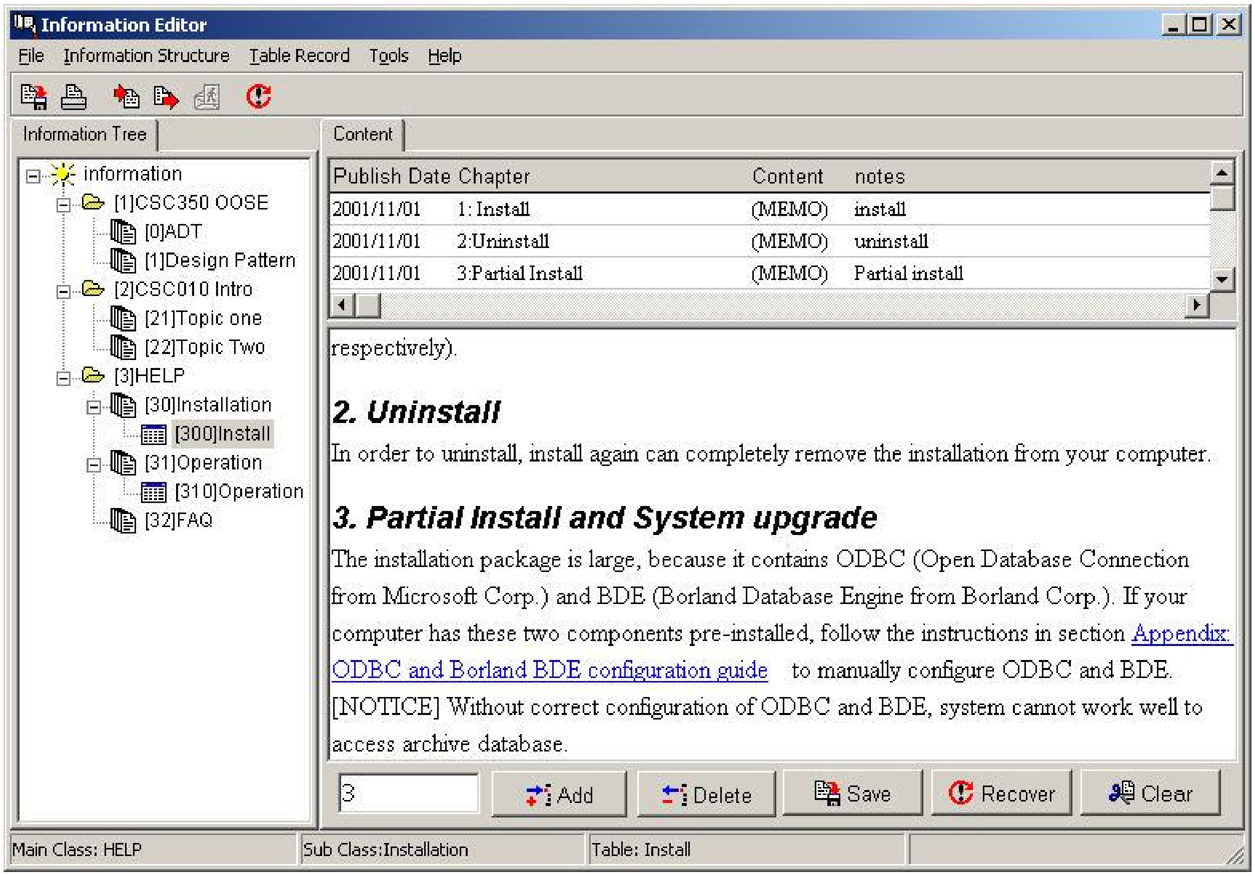


Status
Contact: gdb0

Figure 11: CIMEL client user interface

The instant messaging system is particularly useful in situations in which students need a prompt response. In our system, an instant message can be sent to a single person or multicast to a group. Different fonts and "emotion" characters can be selected to enhance the visual effect in the chat session. Chat conference rooms are used for group discussion. There are two kinds of conference rooms: system pre-defined and user-defined. User-defined conference rooms can be assigned one of three access privileges: public, which is a room open for all users, protected, which is a room open for users listed by the creator, and private, which is a room for private communication that doesn't appear in the conference room list.

In addition to implementing these common features of instant messaging and chat, our framework supports the recording of communication sessions that can be added to a Frequently Asked Questions (FAQ) database, thereby supporting much of what is currently done with communication in an educational setting. Our system uses a flexible database scheme to manage the information. It uses an information tree (or information dictionary) and an information table (including the schema of the table) that are built and maintained dynamically by an authorized user. Chat archives can be stored and displayed in Rich Text Format, which is particularly useful for the storage of non-textual elements. The system also allows the user to sort and search archived FAQ information.



Publish Date	Chapter	Content	notes
2001/11/01	1: Install	(MEMO)	install
2001/11/01	2:Uninstall	(MEMO)	uninstall
2001/11/01	3:Partial Install	(MEMO)	Partial install

respectively).

2. Uninstall

In order to uninstall, install again can completely remove the installation from your computer.

3. Partial Install and System upgrade

The installation package is large, because it contains ODBC (Open Database Connection from Microsoft Corp.) and BDE (Borland Database Engine from Borland Corp.). If your computer has these two components pre-installed, follow the instructions in section Appendix: ODBC and Borland BDE configuration guide to manually configure ODBC and BDE. [NOTICE] Without correct configuration of ODBC and BDE, system cannot work well to access archive database.

Figure 12: FAQ information archive interface

In order to record the communication history of participants for inclusion in the FAQ and for future analysis, our system implements a tracking mechanism. The user name, IP address, time stamp (message entry time, send time and receive time), and chat messages are stored in a database. In addition to facilitating the creation of FAQ elements, collected data will help with future analysis of the use and effectiveness of real-time collaboration tools.

4.0 Evaluation Methodology and Results

Multimedia

For the OOSE course, the alpha version of the multimedia covered abstract data types (ADTs). We designed an experiment to determine whether multimedia actually improves learning, both in terms of objective knowledge and in terms of students' ability to perform a task designing ADTs for a sample problem involving several classes and inheritance. We divided the 20 students into two groups, A and B. Both groups took the same pretest of 20 questions, with questions and choices presented in random order. Then group A studied the multimedia. Following this both groups took a mid-test, consisting of 20 questions, each covering similar topics as the original set, again with questions and choices presented in random order. (Both groups took the mid-test so that the practice effect could be accounted for in our analysis.) Both groups then attended a lecture on ADTs and were assigned reading from relevant chapters in a textbook on reserve at the library. Both groups then were given an assignment, creating ADTs to solve a problem involving several classes, inheritance and dynamic binding. (The problem description is given in www.cse.lehigh.edu/~glennb/oose/afruitADT.htm.) After handing in their assignments, Group B studied the CIMEL multimedia on ADTs. Then both groups were given the opportunity to resubmit the assignment. Our hypothesis was that Group B would be more likely to resubmit the assignment with improvements. Both groups then took a post-test of another 20 questions. Our hypothesis was that both groups would show an equivalent learning effect from the pre-test to the post-test, regardless of whether they studied the multimedia or the lecture first.

For the Introduction to Computing course, the alpha version of the CIMEL multimedia covered Java programming. For our study, we divided participants from this course into two groups. An experimental group (80% of about 50 students participating) got access to all materials (the complete new manuscripts, the old multimedia associated with *The Universal Machine*, and CIMEL multimedia for five chapters on Java programming associated with *The Universal Computer*), while a randomly selected control group (20%) got everything but the CIMEL multimedia. We compared performance of the two groups, with respect to performance on actual Java programming assignments and examinations, as well as a pre-test (administered at the beginning of the course) and a post-test (during finals), consisting of the same 30 multiple-choice questions, with questions and choices presented in random order for each participant. In addition, a survey solicited feedback from the students with respect to the user interface and content of the new multimedia, and demographic data was gathered in order to track performance and feedback of different subgroups, such as women and minorities.

For the OOSE course, twenty students were divided into two equal groups, randomly. Group A, who saw the multimedia first, performed significantly better than group B on the mid-test. For Group A, mean scores improved from 9.2 on the pre-test to 15.6 on the mid-test, while group B showed no change. The difference between groups was highly significant, $F(1,20)=42.23$, $p<0.0001$. This result suggests that the multimedia does indeed contribute to objective learning of this content.

On the post-test, by which time both groups had been to lecture and completed a task as homework, and group B had completed the multimedia, group B also showed significant improvement from the mid-test, while Group A's performance remained the same, as measured by a test for the group-x-test interaction, F(2,60)=7.77, p<.001. This result suggests that the multimedia contributes to objective learning regardless of whether it is presented before or after the lecture or homework task. This result also suggests that students can learn the objective material from the multimedia independent of any lecture material.

After a lecture on ADTs, both groups were given an assignment, creating ADTs to design a solution to a problem. After handing in their assignments, Group B saw the multimedia. Then both groups were given the opportunity to resubmit the assignment. Our hypothesis was that Group B would be more likely to resubmit the assignment with improvements. Several students did so, but the overall results were not statistically significant, F(1,40)=3.25, p>.05. Thus the results for task learning are less clear than the results for objective knowledge. In part this may be due to the finer nature of the data collected for task learning, which all took place after presenting the content in lecture.

For the Introduction to Computing course, 25 students completed the study. These were divided into two groups, randomly, with 18 students in an experimental group (using the CIMEL multimedia) and 7 in a control group (not using the CIMEL multimedia). Both groups took a pre-test at the start of the course and a post-test at the end of the course, consisting of the same set of 20 multiple-choice questions. The experimental group gained 13.5 points, while the control group gained 9 points between the pre-test and post-test. Though a difference in the change in scores occurred, it was determined to be non-significant with p = 0.08.

Demographics appear to imply that the CIMEL multimedia instruction of *The Universal Computer* can have a significant impact on the achievement of students when learning programming. While the groups were similar in their demographic makeup, the students' motivation for taking the course may have impacted the results of the assessments. In the experimental group, 78% reported taking the course to fulfill the requirement of their major or minor, while in the control group, 71% indicated they were taking the course out of personal interest. Thus there is a possibility that differences in motivation could have impacted the results of each assessment.

Incipient Emerging Trend Detection

An experiment was conducted to test our methodology for detecting emerging trends. 21 students participated in this experimental evaluation. The subjects were all students of the graduate course in OOSE. They were asked to identify three emerging trends in the area of Design Patterns. This main topic area was given to them to simplify the experimental evaluation and also to make it more relevant to their coursework. A survey was conducted before starting the experiment to find out how many students knew the definition of the term 'literature search' and how many of them had used web-based resources before to search for information. 19 students responded to the survey. While all of them said that they use the web to search for information, 57.8% said that they didn't know what a literature search was.

The class was divided into two groups, groups A and B, each group having an approximately equal number of students. Students from both groups A and B were expected to have attended the lectures of the class. They were also expected to have introductory knowledge in the main topic area of Design Patterns before participating in this experiment. This was necessary as at different steps of the experiment they needed to apply their domain knowledge to justify their choices of emerging versus non-emerging trends. Also, all the students had access to their textbooks, reference books and handouts given in the class. Both groups A and B attempted an exercise that involved identification of three emerging trends in the area of Design Patterns. In addition, group B was provided with the methodology. Group B was also provided with a practical case study that demonstrated the process of detecting emerging trends as outlined in the methodology. After completing the task, students in group A were given the methodology and case study and required to resubmit their solutions using the methodology.

The standard metrics of evaluation in text mining are precision and recall and the trend detection task was evaluated using precision. Precision is defined as the number of truly emerging trends over the number of trends the students identified. Recall is defined as the number of truly emerging trends identified verses the number of emerging trends in the main topic area. For this exercise, however, we did not use recall as a metric. We did not have the resources to obtain a complete list of emerging trends at the time of this experimental evaluation, nor was it our pedagogical goal to have students retrieve all trends.

Hypotheses:

1. Group B (R5: followed methodology including resubmissions) will perform significantly better than group A in terms of precision of their results.
2. Group B (R4: followed methodology excluding resubmissions) will perform significantly better than group A in terms of precision of their results.

	Group A (R1)	Group B (R2)	Group B (R3)	Group B (R4)	Group B (R5)
Mean	54.54	60	69.23	100	100
Standard Error	11.26	12.96	10.98	0	0
Median	66.67	66.67	100	100	100
Std Dev	37.34	40.98	39.58	0	0
Sample Variance	1394.01	1679.06	1566.99	0	0
Range	100	100	100	0	0
Minimum	0	0	0	100	100
Maximum	100	100	100	100	100
Sum	600	600	900	400	600
Count	11	10	13	4	6
95% Confidence	25.08	29.31	23.92	0	0

Table 1: Analysis of Group A and Group B precision results (R2) entire group B, (R3) entire group B including resubmissions, (R4) actually followed methodology, and (R5) actually followed methodology including resubmissions

The sample data collected is based on precision of the results of student performance on the assignment. As noted, precision of the results is calculated based on the number of correct trends identified by each student versus number of total trends identified.

Using a lower tail t-test we found, with a confidence level of 99%, that the mean precision of sample 2 (actually followed the methodology including resubmissions from Group A) was significantly greater than the mean precision of sample 1 (without the methodology). Also, with a confidence level of 95%, mean precision of sample 2 (actually followed the methodology excluding resubmissions from Group A) was significantly greater than sample 1 (without the methodology). This means that students who used the methodology performed significantly better on the task. These initial results are promising.

In our experiment initially we did not find a significant difference in precision results between Group A (R1: without methodology) and Group B (R2: entire group B with methodology). However, after a critical study of the results and the experimental methodology, we found that there are some variables that we were unable to account for in this first set of experiments. For example, some students, even with the methodology, did not actually finish the assignment and reported only one or two out of three required emerging trends.

To address this concern we held a focus group discussion with the students and determined that at least some of them decided to stop after putting in seven hours of time on the assignment. This gave us some confidence to use partial results because we had explicitly directed the students to stop after seven hours.

A second variable that we were unable to control was whether the students in group B actually followed the methodology. Based on the focus group discussion with the students we learned that despite the fact that they were required to follow the methodology, several of them had difficulty understanding it and *did not follow the methodology at all*. As a result, we performed a critical study of group B results, and were able to determine which students had actually followed the methodology because the assignment required students to log the steps they took to detect the emerging trends. In future experiments we will modify our methods to take into account the usability of the methodology.

Finally, our survey result showed that almost half of the students did not know the definition of 'literature search'. The overall result of Group B (R2) may be at least partially attributed to this fact as well. An obvious solution could be to give training on the use of the methodology before conducting an experiment.

Collaborative networking

After providing students with an instant messenger and chat tool to install and use, we discovered that the existence of these tools was not enough to motivate the students to use them and, therefore, make use of a more immediate connection to instructors, TAs, and other students. The feedback from students through the online survey and focus group discussion provided some clues for why that is, and what we need to focus on to get students to use the tools.

The reasons why students did not use the tools varied. The tools were made available late in the semester, so many students did not feel they had the time to install the tools on their computer systems. Those that did had initial difficulties accessing the system (e.g., there was confusion about passwords), connecting to the server, and connecting at a time when the instructor was also connected. Another significant point made in the focus session was that the advantage of using this tool over other similar tools that they were already familiar with was not clear to them. On the other hand, of those that tried the tools, no one criticized the interfaces or their usability.

Though the instant messenger and chat tools were not widely used, focus group discussion did provide support for the collaborative tool design as a whole. A question regarding collaborative capability was posed. The participants thought that tools that allowed people to synchronously see and manipulate the same application from different computers would be particularly useful for getting help with debugging code because the professor and student would be able to see and edit the same code. The participants thought that this capability would also be useful in other situations, such as in getting clarification about topics presented in the multimedia.

On the other hand, in response to questions asking if adding audio and video would be useful, we learned that these features might not be helpful, as many people noted that they did not have microphones or cameras. Based on these responses, we have prioritized our development of collaborative tools, from highest (most desired) to lowest (least desired): interactive demonstration, multimedia FAQ, instant messaging, chat, audio and video connections.

6.0 Related Work

There is a rich and growing literature of multimedia-based educational material. The CIMEL framework seeks to provide a framework for curriculum development in computer science and to support multiple learning styles and tracks. In this section we compare our approach to a selection of comparable research efforts and systems.

An ongoing project at Massey University[2] is developing and evaluating an integrated system for web-based education. This system uses web-based delivery of course material including interactive multimedia presentations, problem solving and simulation environments in which students learn by doing. Like CIMEL, TILE (Technology Integrated Learning Environment) provides students with an interactive multimedia environment, and developers with a framework for managing, authoring, monitoring and evaluating multimedia. The most salient differences are that the CIMEL framework lets students go beyond the lessons through collaboration with experts (e.g., instructors, TAs, research librarians and other student) as well as through tools that allow the student to explore current research trends in course-related literature.

The Interactive Learning Modules (ILM)[3] presents web-based multimedia tutorials, created with the Director authoring environment. ILM provides a mechanism for the creation of supplementary material for lectures, and collaborative problem solving environments. The system is highly modular to encourage the usage of parts of the lesson material in different courses. Similarly, the CIMEL multimedia framework is being developed with modularity in mind, where each screen is a separate Flash movie, and screens are organized hierarchically into sections and chapters. (We chose Flash instead of Director or Authorware because its vector-based graphics lets us easily use the whole screen, regardless of resolution.) The CIMEL dynamic tracks interface will let instructors and students create and traverse their own learning track, corresponding to their unique requirements

Another system that provides course material in a student-based manner is MLE (Multimedia Learning Environment)[4], a networked educational application. MLE provides a virtual learning environment, through a client application, where multimedia educational material is structured in Adaptive Hypermedia from which sets of hypermedia pages are dynamically retrieved and presented to the student by tailoring both contents and presentation style to the student's needs. In addition to this dynamic content, the MLE system provides a mechanism for real-time audio communication between a student and an instructor. CIMEL also uses a client-server model to present multimedia through a high-speed network.

The ProgramLive application[5] provides a rich multimedia tutorial of the Java programming language. ProgramLive's interface represents a notebook, within a browser. There are tabs to the side of the notebook display that can be used for the navigation of the material, as well as pop-up explanation of key terms. The CIMEL user interface also plays through a browser, but avoids mixing interface metaphors by eliminating all of the usual buttons of a browser. The interface thus immerses the learner in an environment uniquely associated with the material[†]. Another difference between ProgramLive and CIMEL is in their approach to feedback. CIMEL provides an explanation of each of the wrong answers as the student makes these mistakes rather than just providing the correct answer. This helps the student to gain a better understanding of the thought process involved in solving problems.

To the best of our knowledge there are no other projects that aim at inquiry-based learning based on emerging trend detection. The emerging trend detection module currently under development is based on our previous work in the field of textual data mining[6,7,8]. In this previous work we examined the usage of various linguistic and statistical features to track trends across time. The HDDI™ system[9,10] is used to extract linguistic features from a repository of textual data and to generate clusters based on the semantic similarity of these features. The rate of change in the size of clusters and in the frequency and association of features is used as input to machine learning techniques to classify topics as emerging or non-emerging. However, a domain expert does not use linguistic features exclusively to detect an emerging trend. Our current research is motivated by the desire to better characterize a domain expert approach to the detection of emerging trends. Through this research we aim to identify features and methods to enhance the automatic detection of emerging trends.

Several research projects are exploring solutions to the detection of emerging trends in a non-classroom setting. ThemeRiver[11] enables users to visualize trends and detect emerging trends. It is a prototype (mock-up) that visualizes thematic variations over time across a collection of documents. As it flows through time, the river changes width to depict changes in the thematic

[†] One thing we have learned from our focus groups, however, is the need for a minimize function—though it won't necessarily need to look like the usual button.

strength of temporally collocated documents. The river is within the context of a timeline and a corresponding textual representation of external events.

The Envision system[12] lets users explore trends in digital library metadata (including publication dates) graphically to identify emerging concepts. It is basically a multimedia digital library of computer science literature, with full-text searching and full-content retrieval capabilities.

TimeMines[13] is an automated system that generates overview timelines for topics in free text news corpora. These timelines are used to indicate the key topics involved in the corpora and their coverage with a ranking function of how important a particular topic is within that area. In contrast, our research goal is not to identify all topics that are important but rather identify selected emerging trends that are incipient.

Another aspect of the CIMEL multimedia framework that we have not seen in many other projects is the collaborative interface. The ability to collaborate with experts and other students is a useful feature and we are developing a wide range of communication modes for the CIMEL multimedia framework. Below are some comparisons between existing collaborative environments for learning and our proposed interface.

MRAS (Microsoft Research Annotation System)[14] is a system is designed to support annotation of multimedia content on the web, a fine-grained access control structure, and close integration with email. This system allows a user to write and examine the annotation to multimedia content such as video in a HTML page. Through this process of commenting on different points in a video, users can collaborate with one in the discussion of the underlying multimedia. The CIMEL project however pursues a more direct collaborative environment where students will be able to collaborate through wide variety of methods (e.g., chat, voice, stored explanation or "show me" sessions). Additionally the goal of the CIMEL collaborative interface aims at providing a wider means of communication that is focused at enhancing the entire learning experience rather than focusing on single points in a lesson.

Another collaborative project[15], the VIENA Classroom, provides an interactive learning environment in which students with questions about the material can query a FAQ system with a natural language interface and highlighted context from the material. The student receives an answer from the FAQ knowledge base that is semantically similar to the material and the query. Otherwise the system forwards the query to an instructor, or in the case of multiple queries the system will group and form representatives for sets of queries on the same topic. An instructor can then collaborate either directly with a student through a shared window or through an off-line answering mode. The CIMEL multimedia framework aims at providing a wider variety of expert sources (instructors, TAs, research librarians, and peers) and different collaboration mechanisms (direct peer-to-peer communication, FAQ listings and "show me" explanations).

7.0 Conclusions and Future Work

Our experimental study with the unit on ADTs indicates that the alpha version of CIMEL is more successful at conveying objective information than at conveying the knowledge needed to complete a task. However, we maintain (and have confirmed in user surveys and a focus group meeting with the students) that improvements to the multimedia can enhance learning needed for task knowledge. Improvements planned include:

- Correcting bugs and unclear steps in the interactive and constructive exercises. Some students in the OOSE focus group reported that they experienced trouble with these exercises that caused them to skip over parts of the material.
- Adding a checklist at the end of the module, summarizing what one should consider in designing ADTs. Each item in the checklist will also include a link back to the relevant screen, for review. Students in the OOSE focus group strongly agreed that a checklist would give them a better idea how ADTs were designed as well as a good review.
- Supplying a less media intense mode of presentation of the content as an alternative to the Flash media-rich version of the content. Students in both OOSE and CS0/1 focus groups were roughly split about whether they liked media-rich techniques such as animated, graphical personae and audio narration in the alpha version. Those who liked the rich media included most of the women and non-majors, while those who found these media a bit annoying and wanted to get right to the nuts and bolts tended to be male and more experienced in computer science. Even students who prefer the media-rich presentation would like a way to review the material in a less media-rich form. This result suggests that it would be useful to provide a way to generate an additional set of HTML pages with the text and key graphics from Flash content pages, dropping the use of personae and audio. (Interactive exercises will remain as is.)
- Complete the integration of the incipient emerging trend detection module for inquiry-based learning.
- Completing the remote demonstration system, where people can synchronously see and manipulate the same application from different computers. Our proposed approach is to develop technology integrating the standard image based remote control as well as technology to determine the events and components to reconstruct an interactive demonstration.
- Integrating together a framework that supports remote demonstration, FAQ, chat, whiteboard, shared editing and two way audio.
- Adding logic time stamps to communication components such that communication between peers can be causally ordered. We plan to use this information combined with linguistic analysis to build models of both user interactions with other users and models of user interests. One goal is to automate the generation of FAQs from chat data.

We plan to make these improvements to a beta version of the ADT, inheritance and design patterns modules during the spring semester 2002. We also plan to conduct similar studies as those described earlier, in a Programming Languages class in order to test our hypotheses that multimedia can indeed enhance learning needed to solve a task. We hope to report these results during our presentation at the ASEE conference this summer. Over the summer we plan to develop a beta version of the modules on Java programming, which we will use and evaluate at Lehigh and perhaps other universities next fall.

Acknowledgements

This work is supported by NSF Grant Number EIA-0087977 and Lehigh University cost-sharing. The authors would like to acknowledge the National Science Foundation and Anita LaSalle, Mary Poats and Harry Hedges, Program Directors for the Combined Research-Curriculum Development (CRCD) Program. The authors would also like to gratefully acknowledge the students in CSc 432 and CSc 14, participants in the use and evaluation of CIMEL; Edwin J. Kay, co-author of *The Universal Computer* and contributor to the statistical evaluation; Morgan Jennings and Debra Dirksen, our expert educational evaluators; Chris Janneck, Mark Shulman, Pamela Wikkerink-Blank and David Goldfeder, the faces and voices of the personae; and Martin Herr, Harriet Jaffe, Sumit Jain, Adam Kinnear, Andrew Mall, Aaron Sherrick, and David Servas for their contributions to the design and implementation of the multimedia and database tracking system. Above all, co-authors Glenn D. Blank, William M. Pottenger and G. Drew Kessler give thanks to God for His inspiration and provision in this work.

References

1. G. D. Blank, William M. Pottenger, G. D. Kessler, M. Herr, H. Jaffe, S. Roy, D. Gevry, Q. Wang. CIMEL: Constructive, collaborative Inquiry-based Multimedia E-Learning. Presented at the *6th Annual Conference on Innovation and Technology in Computer Science Education* (ITiCSE), June 2001.

2. Professor Chris Jesshope, Dr Eva Heinrich, Dr Kinshuk. *On-line Education using Technology Integrated Learning Environments.* Massey University, New Zealand. www-tile.massey.ac.nz/publicns.html.

3. Don Lewis Millard. Interactive Learning Modules for Electrical Engineering Education and Training. In *the Proceedings of the American Society for Engineering Education.* Session 305, 1999.

4. Marco Roccetti and Paola Salomoni. A Web-based Synchronized Multimedia System for Distance Education. In *the Proceedings of the 16th ACM Symposium on Applied Computint,* pp. 94 – 98, 2001.

5. David Gries & Paul Gries. *ProgramLive - Master Java Programming in a Self-paced Learning Environment.* New York, John Wiley and Sons, Inc. 2002.

6. William M. Pottenger and Ting-hao Yang. Detecting Emerging Concepts in Textual Data Mining. Chapter in *Computational Information Retrieval*, Michael Berry, Ed., SIAM, Philadelphia, PA, August 2001.

7. T. Yang. *Detecting Emerging Contextual Concepts in Textual Collections.* M.S. Thesis, Department of Computer Science at the University of Illinois at Urbana-Champaign, 2000.

8. L. Zhou. *Machine Learning Classification For Detecting Trends In Textual Collections.* M.S. Thesis, Department of Computer Science at the University of Illinois at Urbana-Champaign, 2001.

9. Fabien Bouskila and William M. Pottenger. The Role of Semantic Locality in Hierarchical Distributed Dynamic Indexing. In the *Proceedings of the 2000 International Conference on Artificial Intelligence (IC-AI 2000),* Las Vegas, Nevada, June 2000.

10. W. Pottenger, Y. Kim, and D. Meling. HDDI™: Hierarchical Distributed Dynamic Indexing. Chapter in *Data Mining for Scientific and Engineering Applications,* R. Grossman, C. Kamath, V. Kumar and R. Namburu, Eds., Kluwer Academic Publishers, 2001.

11. Susan Havre, Beth Hetzler and Lucy Nowell. ThemeRiver™: In Search of Trends, Patterns, and Relationships. Battelle Pacific Northwest Division. Presented at *IEEE Symposium on Information Visualization,* InfoVis '99, San Francisco CA, October 25-26 1999.

12. L. T Nowell, R. K France, D. Hix, L. S Heath and E. A Fox. Visualizing Search Results: Some Alternatives to Query-Document Similarity. In *Proceedings of SIGIR'96*, Zurich, 1996.

13. Russel Swan and David Jensen. TimeMines: Constructing Timelines with Statistical Models of Word Usage. In the *Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2000.

14 Bargeron, D., Gupta, A., Grudin, J., Sanocki, E. & Li, F. Asynchronous collaboration around multimedia and its application to on-demand training. In the *Proceeding of the 34th Hawaii International Conference on System Science*, 2001.

15. W. Winiwarter et al., Virtual Distance Education in a Collaborative Hypermedia Environment – The VIENA Classroom. In the *Proceedings of the Joint Pacific Asian Conference on Expert Systems/ Singapore International Conference on Intelligent Systems*, 1997, pp. 575-582.

Author Biographical Sketches

GLENN D. BLANK

Dr. Glenn D. Blank (blank@cse.lehigh.edu) is an Associate Professor of Computer Science and Engineering at Lehigh University. His work as principal investigator for the CIMEL project builds on his experience as the lead author, designer and developer for *The Universal Machine: A Multimedia Introduction to Computing* (McGraw-Hill, 1998).

WILLIAM M. POTTENGER

Dr. William M. Pottenger (billp@cse.lehigh.edu) is P.C. Rossin Assistant Professor of Computer Science and Engineering at Lehigh University. His research as a co-principal investigator for the CIMEL project builds on his existing research program in textual data mining[6,7,8,9,10].

G. DREW KESSLER

Dr. G. Drew Kessler (dkessler@cse.lehigh.edu) is an Assistant Professor of Computer Science and Engineering at Lehigh University. His contributions to the CIMEL project as a co-principal investigator follow from his research interests in user interface systems, distributed collaborative systems, and multi-user virtual environments.

SOMA ROY

Soma Roy (sor4@lehigh.edu) is currently a Research Assistant in the Textual Data Mining Lab in the Department of Computer Science and Engineering at Lehigh University. Her work involves the development of the methodology and multimedia interface for detection of incipient emerging trends and development of constructive exercises for the ADT module.

DAVID R. GEVRY

David Gevry (drg2@lehigh.edu) is a graduate student at Lehigh University, pursuing a M.S. in Computer Science. He is currently focused on emerging trend detection research and the design of the incipient emerging trend detection module for the CIMEL project.

JEFFREY J. HEIGL

Jeffrey Heigl (jjh5@lehigh.edu) is a graduate student at Lehigh University in Bethlehem, Pennsylvania pursuing a M.S. degree in Computer Engineering. He received his B.S. degree in Computer Engineering at Lehigh University in 2001. Jeffrey is a research assistant working on the CIMEL Multimedia Team.

SHREERAM A. SAHASRABUDHE

Shreeram A. Sahasrabudhe (sas4@lehigh.edu) received his B.E. in Computer Technology from Nagpur University, Nagpur, India in 2001. He is a currently pursuing an M.S. degree in Computer Engineering at Lehigh University. He is a Research Assistant working under Professor Glenn D. Blank on the CIMEL Multimedia Team.

QIANG WANG

Qiang Wang (qiw3@lehigh.edu) is a Ph.D. student and Research Assistant in the GIVE Lab in the Computer Science and Engineering Department at Lehigh University. His research interests are CSCW, distributed systems, virtual environments, and software engineering. He was a systems analyst and product manager of INTESS®-Call Center at Huawei Technologies Co. Ltd. from 1996 to 2000. He received a B.S. degree in Computer Science from Nanjing University, China in 1996.