A Web-Based Educational Environment for Teaching the Computer Cache Memory

Maria Grigoriadou, Evangelos Kanidis, and Agoritsa Gogoulou

***Abstract*—In this paper, a Web-based educational setting for teaching the computer cache memory is presented, aiming at supporting and enhancing the learning process and promoting the active and constructive involvement of students. The educational setting includes Web text-based educational material, a Web-based cache memory simulation program, and educational activities that enable the students to participate actively in the learning process and to collaborate in groups. The design of the educational setting was based on the principles derived from the conceptual change approach regarding the students' theory framework, the students learning difficulties/misconceptions, the text comprehension theory, and contemporary teaching approaches. The results obtained from the application/evaluation of a set of educational activities have been encouraging, indicating that the simulation program and the context of the activities can effectively support and enhance the learning process.**

***Index Terms*—Cache memory, conceptual change, educational activities, educational material, simulation program.**

I. Introduction

RECENTLY, there has been a growing interest in improving the teaching and learning of concepts related to internal computer operations. This issue is important, not only because of the repercussions it may have on the teaching and learning of these concepts but also because, in this way, students can build up effectively their own knowledge of computer operation when they have internalized the lower level information [1]. The present paper focuses on the concepts related to the organization and operation of a computer cache memory.

Teaching the basic concepts relevant to the cache memory operation is not an easy task. Their operation is not directly observable by the students, because it is presented through static figures or means of a manual simulation of the cache followed by oral or written description. In addition, the ability to perform relevant experiments is limited [2], [3] partially because of a lack of appropriate computer-based teaching tools [4], [5].

In order to enhance understanding of the cache memory operation, the students need to perform several experiments with different cache memory organizations, using many traces of actual program execution. A cache memory simulator is a valuable tool since it can support/facilitate such experiments [2], [3], [6]. In addition, research suggests that learning is a social affair; therefore, collaboration and mutual decision making can have an important impact in achieving the expected learning outcomes [7].

In this paper, a Web-based educational setting for teaching the computer cache memory is presented, aiming at supporting and enhancing the learning process and promoting the active and constructive involvement of students. This educational setting includes the following:

- Web text-based educational material;
- a Web-based cache memory simulation program;
- educational activities that enable students to participate actively in the learning process and to collaborate in groups.

The paper is organized as follows. Section II presents the results of research that has been conducted to investigate the students' understanding as expressed by their learning difficulties and misconceptions with respect to 1) the cache memory internal organization and operation and 2) the role of the cache memory in the communication between the CPU and the main memory. Moreover, this section discusses the students' framework theory concerning the main memory and the cache memory concepts and the need for conceptual changes. Section III describes the framework of the Web-based educational material that was developed according to the principles of Web-based distance education and Denhiere & Baudet's text comprehension theory [8]. Section III also presents the simulation program that was developed to help students obtain a better understanding of the cache memory organization and operation. Furthermore, this section discusses design issues concerning the development of educational activities in the context of exploiting the cache memory simulation program and adopting characteristics from contemporary teaching approaches. Next, Section IV presents the main points of the evaluation results regarding the effectiveness of the simulation program and the educational activities in enhancing the learning process. Finally, in Section V, the paper ends with concluding remarks and future plans.

II. Considering Conceptual Change on the Basis of Students' Learning Difficulties and Misconceptions About Cache Memory

Efficient teaching requires that the instructors are aware of the students' learning difficulties and misconceptions, taking into account these difficulties in the instruction process and in the design of appropriate teaching tools and techniques [9]. Experimental studies reveal that students face substantial difficulties in understanding the organization and operation of computer

Manuscript received February 15, 2005; revised October 16, 2005.

M. Grigoriadou and A. Gogoulou are with the Department of Informatics and Telecommunications, University of Athens, GR-15784 Athens, Greece (e-mail: gregor@di.uoa.gr; rgog@di.uoa.gr).

E. Kanidis is with the Inter-University Program of Basic and Applied Cognitive Science, University of Athens, 15784 Athens, Greece (e-mail: vkanidis@di.uoa.gr).

Digital Object Identifier 10.1109/TE.2005.863431

TABLE I
STUDENTS' DIFFICULTIES/MISCONCEPTIONS CONCERNING THE CACHE MEMORY ORGANIZATION AND OPERATION

Students ...	Specifically, the students ...
have partial or fault knowledge about cache memory operations	describe a cache memory operation as *(a)* only a part of this operation (e.g., the identification operation as the address bit partitioning) (b) another operation (e.g., the placement operation as the identification operation) (c) a part of another operation (e.g., the placement operation as the address bit partitioning)
describe cache memory operations with common-sense concepts, not scientific ones	believe that the placement operation is performed only when the cache memory is empty and/or that the replacement operation is performed when the cache memory is full
use a simplified model of cache memory	use the simple model of a direct mapped cache memory instead of the most complicated set-associative cache memory required
have fault understanding of the n-way set-associative cache memory organization	believe that (a) the number n corresponds to the number of sets, and not to the number of blocks existing in a cache memory set (b) the offset indicates the place of the block in the set
perform the cache memory operations in a wrong sequence and/ or have fault knowledge about their causal relationships	believe that (a) the write operation takes place without a prior need to perform the identification operation, followed by a placement or replacement operation in case of a cache miss (b) the placement operation, not the replacement operation has to be performed, when the result of the identification operation is a cache miss caused by the block tag field while the valid bit is set
design insufficiently the new state of cache memory after an operation takes place	change the content of the block but not the block tag field after a replacement operation

cache memory [5]. In Table I, a succinct presentation of their learning difficulties/misconceptions is given. The students' difficulties are a result of a) the text-based material form used, b) the limited ability of performing experiments, and c) the absence of the required students' theory framework [5].

The students' preconceptions form a framework theory, where the term "theory" is used to denote a relational explanatory structure [7]. As far as the organization and the operations of the computer main memory are concerned, the students' framework theory includes the following characteristics.

1) The role of the main memory is to store the program executed by the CPU and the relevant data.
2) Special instructions transfer data (bytes or words) between the main memory and the processor unit.
3) The assembly language programmer uses either the read instruction, in which he specifies the main memory address to be accessed, or the write instruction, in which he specifies the main memory address along with the data that will be stored in it.
4) The operations of the main memory are the read/write ones. The read operation is used either to fetch instructions or data from the main memory, while the write one is used to store data in the main memory.

The students use this framework theory to interpret the cache memory organization and operation and to generate scientific explanations and predictions in a reasonably consistent fashion. However, some cache memory characteristics are not compatible with the explanatory structure of the main memory, as explained more specifically in the following.

1) The role of the cache memory is to increase the computer performance by holding the most recently accessed code or data, as it is a small and fast memory.
2) No special instructions transfer data between the processor and the cache memory. The cache memory is activated after a read or write command of the processor. Cache memory operations transfer bytes or words between the cache memory and the processor, while they transfer blocks between the main and the cache memory, taking advantage of the principle of locality of reference.
3) The assembly language programmer does not make any reference to cache memory addresses. The main memory addresses are mapped automatically to cache memory addresses depending on the cache memory organization.
4) The operations of the cache memory are the identification, the read or write in case of a cache hit, and the placement or the replacement in case of a cache miss.

Accounts of the knowledge acquisition process have customarily assumed that knowledge acquisition proceeds in a continuous manner, enriching initially fragmented conceptual structures and making them increasingly more systematic and more coherent. However, when main memory framework theories come into contact with cache memory organization and operation instruction, fragmentation, incoherencies, and misconceptions are often the result. Cache memory instruction proceeds by fragmenting initial main memory explanatory framework without succeeding in building an alternative cohesive scientific explanatory framework. In this context, to

help students acquire new knowledge and achieve conceptual change, radical reorganization of what is already known is required [10].

The appropriate learning environment should be based on instruction design principles derived from the conceptual change approach [10], such as taking into account students' prior knowledge, providing adequate sources, facilitating students' metaconceptual awareness and metacognition with educational activities. In this context, an educational setting has been defined that consists of appropriate text-based educational material, a cache memory simulation program, and a set of educational activities, which take into account students' learning difficulties. The whole educational setting is Web-based since it is considered important to enable students to experiment in their own personal time and to support processes like self-assessment, self-regulation, and reflection.

III. Educational Setting

A. Web-Based Educational Material

The text-based educational material is a ubiquitous part of the education. The text-based educational material developed is based on Hennessy and Patterson's books [11], [12] and follows the principles and specifications of developing text material for Web-based distance education. In particular, each chapter of this educational material comprises a general overview, learning outcomes, development/presentation of the underlying concepts/subject, self-assessment exercises, and summary [13].

As stated above, two main reasons for the students' difficulties/misconceptions are the material form and the absence of the students' required theory framework. Taking these factors into consideration to facilitate students to reorganize their framework theory and resolve the inconsistencies, the chapter on cache memory was developed [14] on the basis of the students' difficulties, their framework theory [5], and Denhiere and Baudet's text comprehension theory [8].

Denhiere and Baudet [8] express the opinion that in order to examine the representation constructed by students during the comprehension process of a text, the primary role should be attributed to the understanding of the cognitive categories *state*, *event*, and *action*. The term *state* is static and describes a situation in which no change occurs in the course of time. The term *event* refers to an action that causes changes but is not originated from a human. The *event* can be coincidental or caused by a nonhuman action, e.g., by a machine. An *action* is an effect that causes changes but is originated from a human. Furthermore, Denhiere and Baudet say that the text analysis in relation to the conceptual categories of *state, event*, and *action* is not sufficient. Therefore, the organization and structure of cognitive representation should be examined in micro- and macro-levels.

The person who reads a text gradually constructs the *microstructure* of the text representation, i.e., the *states*, *events*, and compound *actions* of the world described in the text, and the time and causal relationships that interlock those structures. Denhiere and Baudet hold that a person, to be able to explain the operation of a technical system in micro-level scale, has to construct a representation of "natural flow of things," in which every new *event* should be causally explained by the conditions of *events* that have already occurred.

The creation of a text that allows a precise description of a technical system and facilitates the reader in constructing its *microstructure representation* must involve the following:

- the description of the units that constitute the system, based on the causal relationship that unites them as a group;
- the description of events sequence taking place in these units with respect to the cause affecting each event, and the changes the events bring to the state of the system.

In macro-level scale, the development of the *macrostructure* by the reader is achieved through the reconstruction of the microstructure and the establishment of a hierarchical structure with goals and subgoals.

In the case of a technical system, the creation of a text that facilitates the reader in constructing its *macrostructure representation* must involve the following:

- the teleological hierarchical structure of goals and subgoals of the various operations and their implications.

On the basis of the above principles, the Web text-based material was organized as follows.

1) Micro-Level Structure:

- The units that constitute the system and the causal relationships among them are described. The system includes three units: the processor, the cache memory, and the main memory. The execution of a read or write command by the processor is the cause that activates the cache memory and the main memory operations. From the cache memory operations, the cognitive category of action is absent, since no human-made changes (user or programmer) are involved.
- When the cache memory is activated, a sequence of events takes place. Each event is a cache memory operation. The sequence of these operations is described as well as the causal and temporal relationships among them according to the processor command (read or write).
- The execution steps of each operation (subevents) and their causal and temporal relationships are clearly described.
- The state of the cache memory system after each operation is declared.
- Moreover, the structure of the text has been clarified with paragraphs, headlines, bold, and other paralinguistic elements.

2) Macro-Level Structure:

- At this level, the main goal of using a cache memory is discussed (i.e., to provide an economical solution to the desire of programmers to possess available unlimited amounts of fast memory). The cache memory is a small and fast memory that takes advantage of the principle of locality of reference and the cost/performance ratio of memory technologies.

The main goal above is divided into three subgoals.

a) Describe ways to overcome the restrictions from the cache memory size and determine where a main memory

TABLE II
IOWA STATE UNIVERSITY CACHE MEMORY SIMULATION PROGRAMS EVALUATION COMMENTS

Program operation	Evaluation comments
1. The mapping techniques for a cache memory and a main memory of fixed size are applied.	The first program supports only fixed sizes of the cache memory and the main memory (8 and 32 blocks, respectively), which are not used in real computers. The program is used in the presentation of the placement operation instead of the right one that is the identification operation.
2. The implementation of main memory address bit partitioning.	The second program would be more useful if the student could select a specific main memory address.
3. The cache memory block is mapped to each one of a series of main memory addresses; the result of the identification operation (hit/miss) is determined for each address. Finally, the program calculates the number of hits and misses.	A strong point of the third simulation program is that the student can give a series of main memory addresses. A better approach is to have the program visualize the steps of the identification operation (e.g., address bit partitioning) and the steps of the placement operation (e.g., updating of the cache block's tag field and valid bit). Finally, useful to the student is the opportunity to select the command (read/write) that will be executed by the CPU and to perform an experiment with a set-associative cache.
4. The communication between the CPU, the cache memory, and the main memory is presented through graphics, text, and animation.	The fourth program enables the student to observe the communication between the CPU, the cache memory and the main memory, but not in detail. Useful would be the ability to have the simulation program visualize the internal structure of the cache memory (e.g., blocks, sets, identification circuits) and the main memory (e.g., address, data), the communication buses between units, and the way in which the operations' steps are represented at each unit.

block can be placed. *This subgoal leads to the three different cache memory organizations along with the corresponding mapping techniques.*

b) Identify whether or not a cache block contains the desired information. *The most common procedure to achieve this subgoal is to add an address tag on each block frame, which gives the block address, and a valid bit to the tag, which denotes whether this block contains data from the program executed by the computer. These fields are used in the identification, placement, and replacement operations.*

c) Discuss/use techniques to take advantage of the principle of locality of reference. *The techniques concern 1) management of the cache memory block size and 2) placement of data from main memory to cache memory (e.g., in case of a cache miss the fixed size block of data containing the requested word and its neighboring words are placed into the cache memory).*

B. Design and Implementation of a Web-Based Cache Memory Simulation Program

In order to specify the design principles of a simulation program for the cache memory operation, a preliminary evaluation of relevant programs was conducted. The simulation programs that were examined and evaluated satisfy the following criteria: 1) they are Web-based programs and 2) they are suitable for university students that take the course of Computer Architecture I, addressing the concept of memory hierarchy.

The following sections briefly present the cache memory simulation programs developed in the Iowa State University and in the University of Massachusetts. These programs meet a number of the evaluation criteria defined (i.e., whether the program supports a) realistic sizes of main memory and cache memory, b) the selection of main memory address, c) the selection of the CPU command to be executed (read/write), d) the visualization of the operations, and e) different replacement strategies and/or write policies). The presentation is based on the main points of the evaluation results.

1) Iowa State University Simulation Programs: The Computer Science Department of the Iowa State University has developed four cache memory simulation programs [15]. In Table II, the program operations and the evaluation comments are presented.

2) Massachusetts University Simulation Programs: The Electrical and Computer Engineering Department of the Massachusetts University has developed three cache memory simulation programs.[1] In Table III, the program operations and the evaluation comments are presented.

Web-Based Cache Memory Simulation Program: To help students have a better understanding of the cache memory operation, the evaluation results of the existing simulation programs and the students' learning difficulties were taken into consideration in specifying the design principles of a cache memory simulation program. These design principles include the following.

1) *The use of a relevant real size for the cache memory and main memory*: The main memory size was set equal to 1 MB, the cache memory size was set equal to 8 KB,

[1]Cache Main Page, Department of Electrical and Computer Engineering, University of Massachusetts, [online] available: http://www.ecs.umass.edu/ece/koren/ece668/cache/frame0.htm

TABLE III
MASSACHUSETTS UNIVERSITY CACHE MEMORY SIMULATION PROGRAMS EVALUATION COMMENTS

Program operation	Evaluation comments
1. The main memory address bit partitioning.	The first simulation program is similar to the second one of the Iowa State University, with the difference that it can be used for more realistic sizes of the main memory (32MB) and the cache memory block (2B and 4B). However, useful also would be the student's ability to select a specific main memory address.
2. The main memory address bit partitioning and the main memory block are mapped to a specific cache memory address.	At the second program, the student can select a specific main memory address but the main memory size is fixed (64K). Useful also, for a specific main memory address, would be the student's ability to observe in detail the steps of the operations and select the command (read/write) to be executed by the CPU.
3. The cache memory block is mapped to each one of a series of main memory addresses and is followed by the result of the identification operation (hit/miss) for each address.	In the third simulation program, the cache organization (direct mapping or full, associative) is selected for a specific cache memory size by the number of cache memory sets. This simulation program is appropriate for more experienced students. The cache memory characterization is not correct given that the number of ways is not equal to the number of sets. Finally, the student would find useful the ability to select the command (read/write) to be executed by the CPU and observe in detail the steps of the identification operation (e.g., address bit partitioning) and the steps of the placement operation (e.g., updating of the cache block's tag field and valid bit).

and the size of the cache memory block was set equal to either 16 B or 32 B.

2) *The provision of facilities to students to enable them to set the desired parameters*:
   a) the command (read/write) to be executed by the CPU;
   b) the block size;
   c) the cache memory organization;
   d) the replacement strategy;
   e) the write policy;
   f) a specific main memory address.
3) *The visualization of the internal structure of the CPU (e.g., registers), the cache memory (e.g., blocks, sets, identification circuits), the main memory (e.g., address, data), visualization of the communication buses between units, and the animation of the operations' steps in detail.*
4) *The detailed visualization of operation steps*: These include identification (e.g., address bit partitioning, circuits), placement (e.g., updating of the cache block tag field and valid bit) and replacement, by using multiple representations.

The simulation program is available from the Department of Informatics and Telecommunications, University of Athens, at http://hermes.di.uoa.gr/simulation. By activating the "Start of Program" button, the student can attempt experimentation on all cache memory operations by setting the following desired parameters (Fig. 1):

- the organization type of the cache memory of 8-KB size (direct mapping, two-way, or four-way set-associative);
- the block size of the cache memory (16 or 32 B);
- the replacement strategy (random, least recently used (LRU) or first-in/first-out (FIFO));
- the write policy in case of a hit (write through or write back);
- the write policy in case of a miss (write allocate or no write allocate).

Then, the student can select (Fig. 2) the following:

- the command to be executed by the CPU (i.e., read command in case of instruction fetch or read data or write command in case of write data);
- the main memory address within a main memory size of 1 MB.

The student can either start a new program or trace the operation steps (i.e., button "Next Step" or "Pause"). Figs. 3 and 4 present indicative operation steps (i.e., Fig. 3 the placement operation and Fig. 4 the identification operation).

C. Designing Educational Activities

A number of educational activities were designed aiming at helping students with the conceptual change regarding the concept of cache memory. The activities utilize the simulation program and their design exploits characteristics from contemporary teaching approaches, and it is based on the following principles:

- address learning outcomes, which are based on the students' learning difficulties;
- enable students to "learn by doing" by engaging them in exploratory and collaborative activities;
- encourage the metaconceptual awareness and the development of intentional learning.

More specifically, the design of the educational activities may take advantage of the characteristics of the exploratory

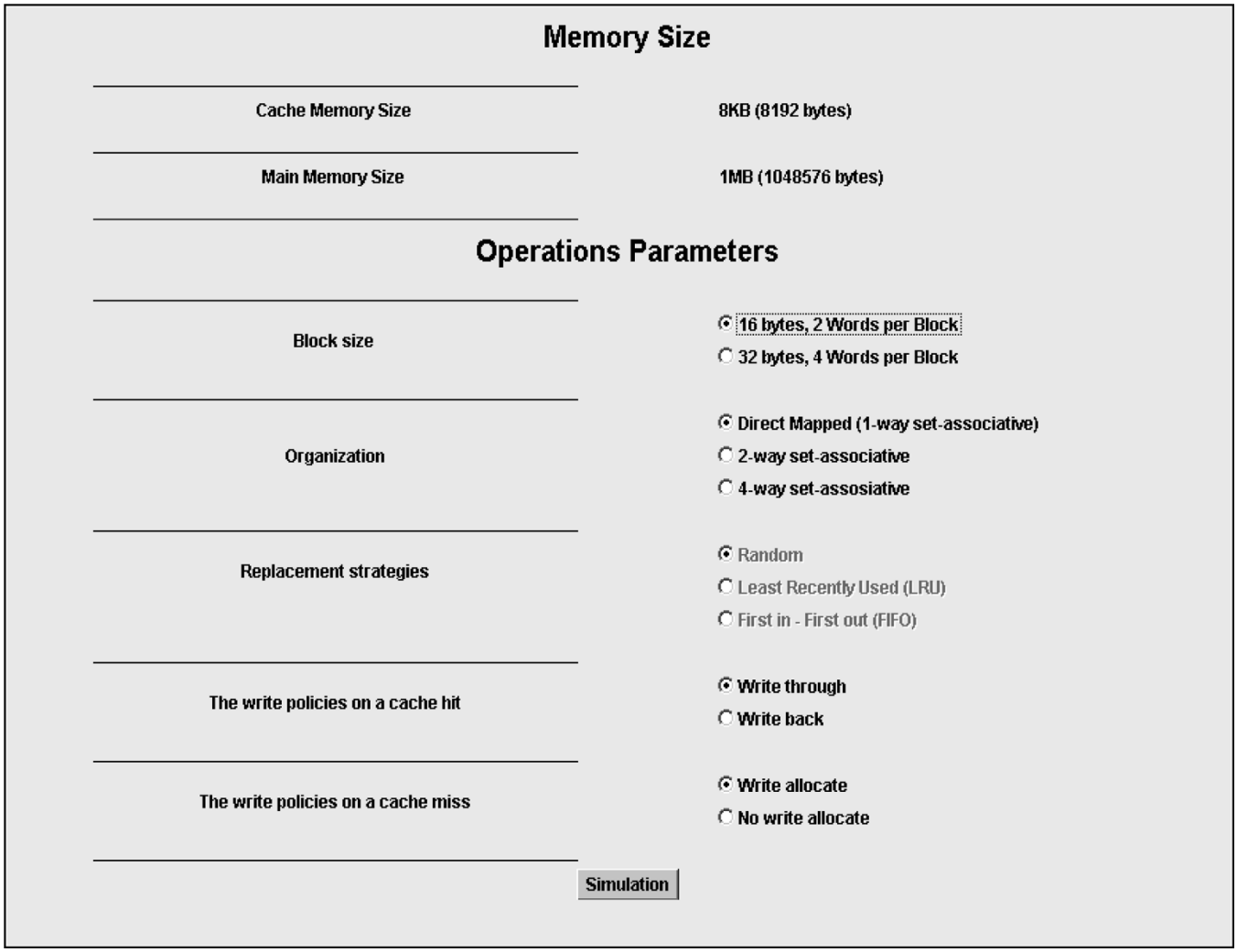

Fig. 1. Set of the desired parameters.

learning [16] in order to support the students in exploring on their own the particular functional properties of the cache memory and also their own ideas and beliefs about it. Moreover, to support knowledge reconstruction and refinement, one finds essential the incorporation of characteristics from collaboration [17]. The collaboration may take the form of group activities in which the students discuss and exchange ideas, or the form of a model of collaboration where the students play specific roles. The engagement of students in exploratory and collaborative activities promotes metacognitive processes, such as reflection, self-explanation, and self-regulation [17], [18].

The elaboration of exploratory activities is supported by the simulation program in which the students have the possibility to experiment with various parameters and observe the execution of commands by the CPU in multiple visual representations. The students may collaborate in the context of the above activities by using the Adaptive Communication Tool (ACT) [19]. ACT is a synchronous communication tool, which adapts the provided set of the scaffolding sentence templates (i.e., sentence openers or communication acts) with respect to the learning outcomes (i.e., cognitive skills) addressed by the activity, the educational tool, if any, used for the elaboration of the activity, and/or the model of collaboration followed by the group members.

1) Design Framework of an Indicative Educational Activity: The activity under consideration was designed according to the principles and the design framework of the teaching approaches of "explorations" and "pair-programming." The "explorations" arise from the constructivists' view of learning, which aims at helping students with the development of effective mental models and with the refinement of existing mental models regarding the way the computer works [20]. These processes have been used and evaluated in the context of an introductory programming course. An "exploration" is a structured dialogue with the student. The student must read a short program, answer questions about that program, make predictions about the program's behavior, and then run the program to check his or her own predictions. Follow-up questions may guide the student to give a plausible explanation in case of wrong predictions. The activities, designed on the basis of "explorations," support the elicitation of the students' prior knowledge and beliefs and enable them to participate actively in the learning process. The scaffolding questions may help the students reconstruct/refine their knowledge toward the desired conceptual change. In addition, the students' engagement in a collaborative setting enables them to examine and discuss their ideas with others, and/or evaluate others' opinions/solutions,

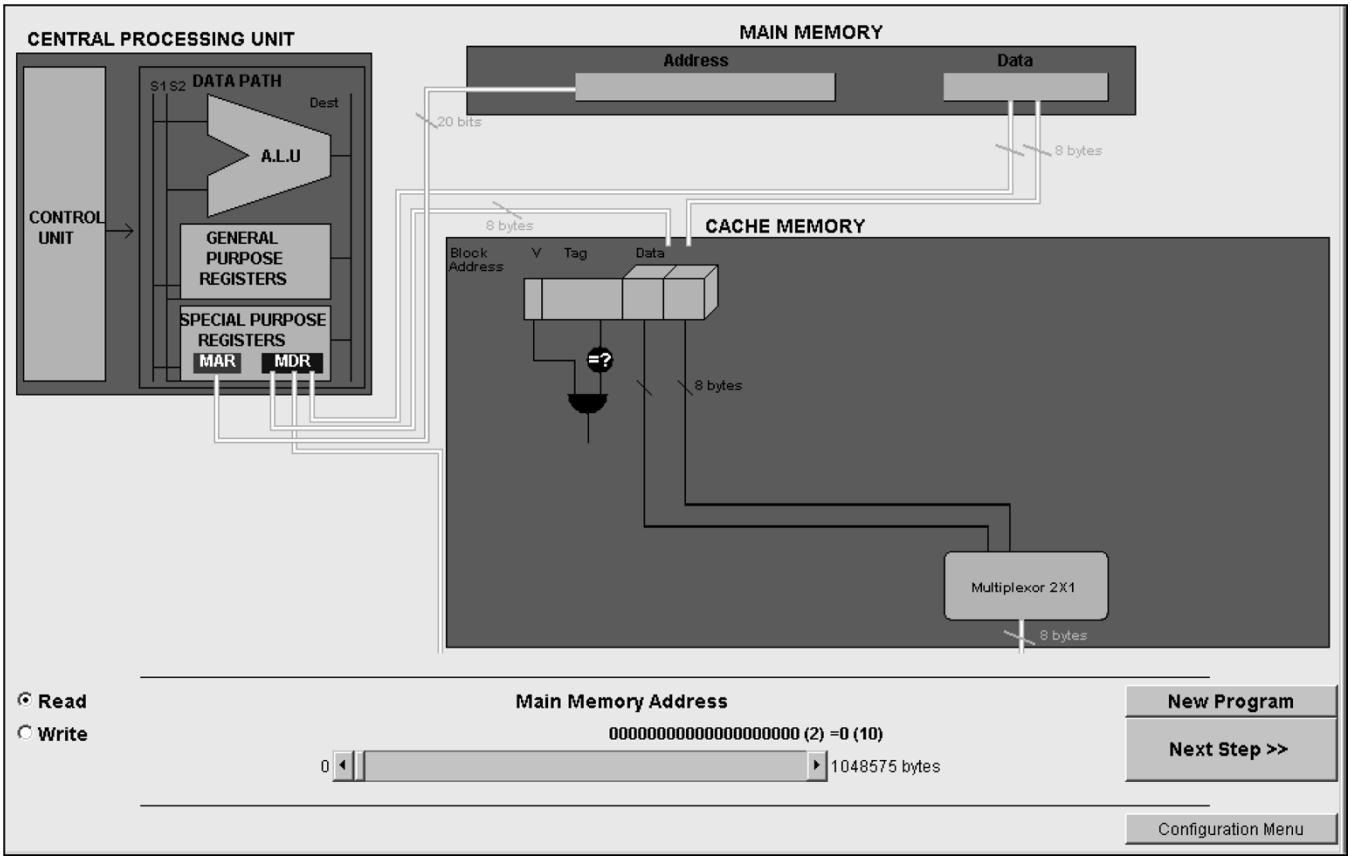

Fig. 2. Student has specified the direct-mapped cache memory and the block size of 16 B.

and supports the reflection process [17]. In the context of the specific activity, the "pair-programming" model [21] is used in which two programmers–students jointly produce one artifact (design, algorithm, code, etc.). The two students of the pair are responsible for every aspect of the artifact. One partner is the "driver" and has control of the pencil/mouse/keyboard and writes the design or code. The other partner, the "observer," continuously and actively observes the work of the driver—watching for defects, thinking of potential alternatives, and asking questions. The roles of "driver" and "observer" are deliberately and periodically switched between the two members of the pair.

The aim of this activity is to help students understand the sequence and the relationships between the operations performed in a set-associative cache memory. More specifically, the learning outcomes were defined taking into account the students' learning difficulties as follows.

- The students should be able to describe the structure of a two-way, set-associative cache memory (this concerns the learning difficulty about the structure of a set-associative cache memory).
- The student should be able to use the address-mapping technique of the main memory to a set-associative cache memory (this concerns the learning difficulty about the use of a simple model of a cache memory instead of a complex one required in the context of a problem).
- The student should be able to predict the sequence of the operations performed (this concerns the learning difficulty of specifying the sequence of the operations).
- The student should be able to specify the changes in the state of the cache memory as a result of performing a specific operation (this concerns the learning difficulty of specifying the required changes in the state of the cache memory).
- The student should be able to compare the structure of a set-associative cache memory with a direct-mapped cache memory.

In the context of the specific activity, the students have to collaborate in dyads, according to the "pair-programming" model, to design the state of the cache memory and to specify and describe the sequence of operations that will be performed for a series of read commands to be executed by the CPU. More specifically, the activity context sets the specifications of a two-way set-associative cache memory (i.e., memory size of 1 MB, block size of 32 B, word size of 8 B, replacement strategy LRU). Taking into account these specifications, the students have to describe the sequence of operations that will be performed and design the state of the cache memory for a series of read commands to specific addresses. Afterwards, the students interchange the two roles and use the simulation program to perform similar operations. They have to observe the real output, to compare their predictions with the output, and to proceed with plausible explanations regarding any differences. The students may use as a scaffolding framework a set of questions, which may guide them in the desired "productive" directions. In particular, the student who plays the role of the "observer" may pose to the "driver" questions such as 1) How many bits are required for the tag, index, and offset fields? 2) Which is the role of the tag field and

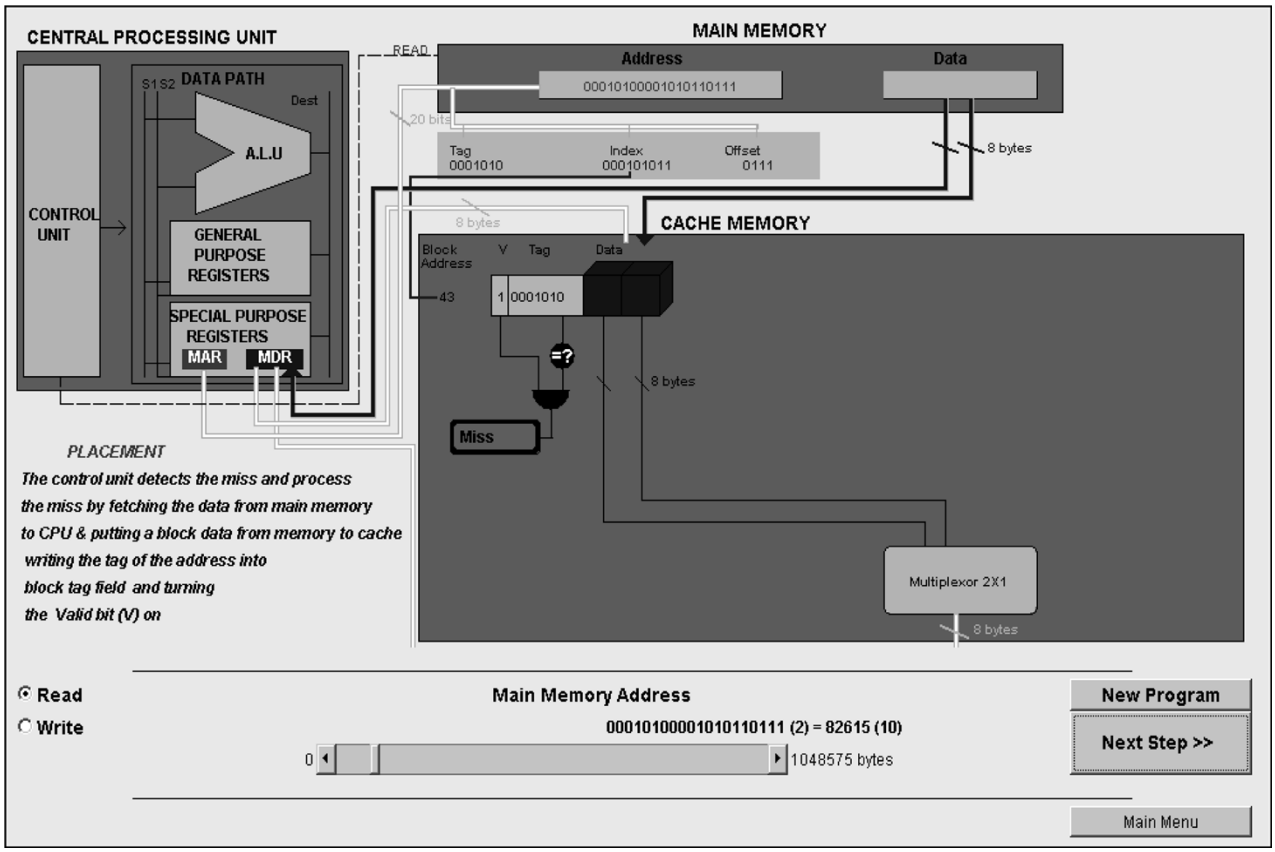

Fig. 3. Placement operation, following a cache miss. The data bus is "activated" to transfer the data from the main memory to cache memory.

the index field in the context of the main memory address? 3) If the identification operation is successful (cache hit) in the case of the cache memory under consideration, would the result be the same in the case of a direct-mapped cache memory? The "driver" and the "observer" communicate through the ACT tool by using the communication acts as scaffolding sentence templates. The students' dialogue and the data about their collaboration are recorded and processed. During the collaboration, students can have access to their model and to the group model in order to have an insight to their own contributions and to their interlocutor's contributions in a graphical form.

IV. Evaluation

The educational activities under consideration were applied under the framework of the course Computer Architecture I at the Department of Informatics and Telecommunications of the University of Athens during the spring semester of the academic year 2002–2003. The students attending the specific course are at the third semester of their studies, having considerable knowledge about logic design. First of all, to identify any learning difficulties and to design activities based on the students' difficulties, the students answered a pretest. The pretest was applied at the end of the semester and had the form of an examination. The analysis of the students' answers revealed that 22 students out of the 64 students had some of the difficulties reported in Section II, concerning the description of cache memory operations sequence and the design of the new cache memory state.

The 22 students constituted the experimental group. Initially, this group performed a number of activities addressing the identified learning difficulties. The educational activities followed the design principles mentioned above, enabling the students to use the simulation program and to collaborate in pairs. After one week, the students participated in a post-test to investigate whether the educational activities helped them to reconstruct and improve their knowledge; the post-test consisted of questions similar to the pretest questions. They also answered a series of questions regarding the activities, the simulation program they used, the model of collaboration they followed, etc. The analysis of the students' answers to the activities and the corresponding post-test pointed out the following.

- The detailed visualization of the steps of the identification and placement operations helped the students to understand that the address mapping techniques are part of the identification operation and the tag field of the cache block is set during the placement operation.
- The visualization of the internal structure of the cache memory and the detailed visualization of the identification operation steps helped 65% of the students to understand the way the identification operation is implemented.
- The visualization of the internal structure of the cache memory helped 75% of the students to describe the organization of the two-way set-associative cache memory.

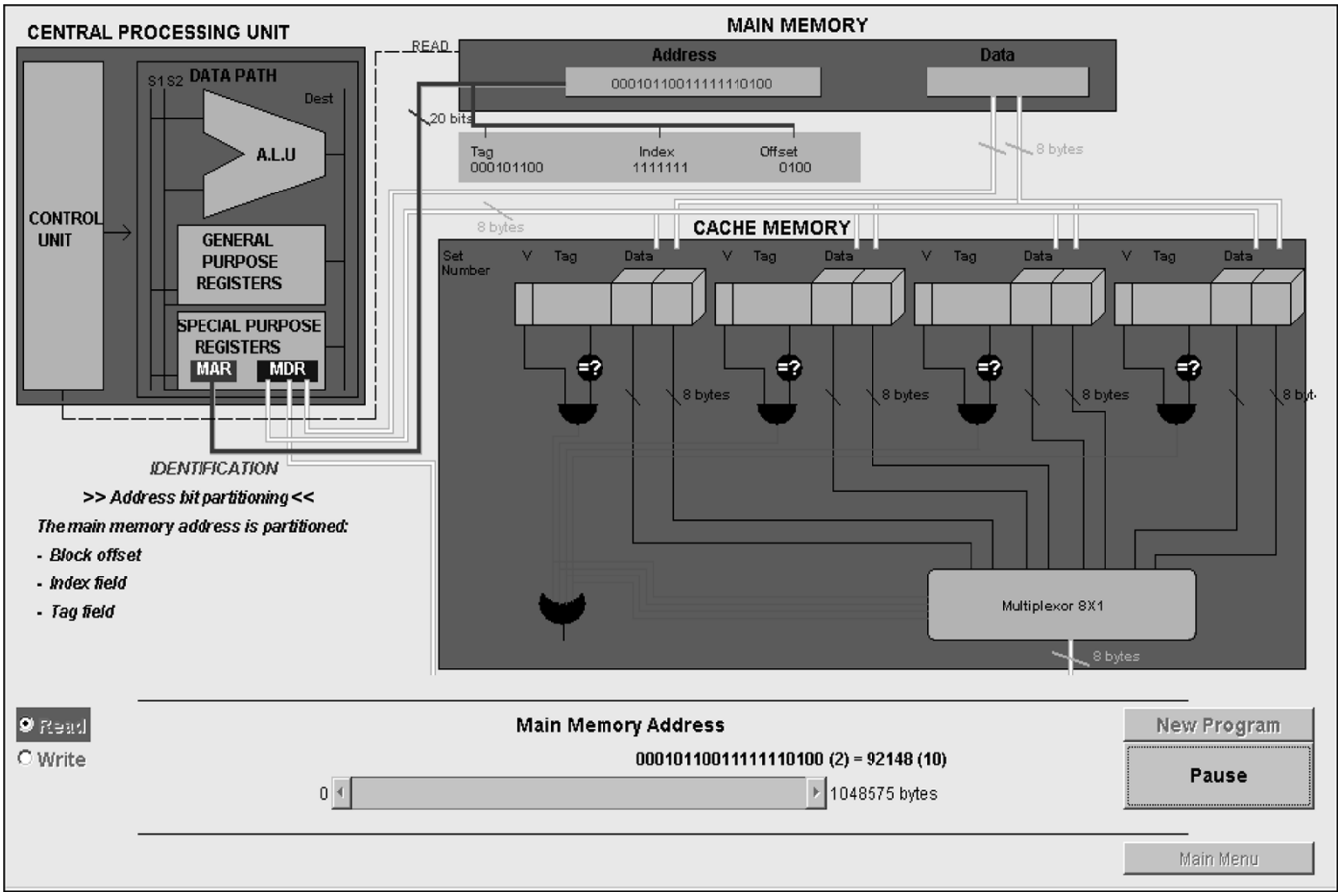

Fig. 4. Identification operation in a four-way set-associative cache memory. The address bus is "activated" to transfer the selected main memory address.

- The sequence of operations that take place in the cache memory after a read or write processor command helped 97% of the students to understand the conditions that are necessary for starting the identification and the write operation.
- The relation of the CPU command with cache memory operations helped 86% of the students to explain the execution steps of the placement operation after a read or write CPU command.
- The execution steps of cache memory operations and the description of the results of each operation helped 75% of the students to design the cache memory state for each one of a series of five main memory addresses that the CPU read sequentially.

In addition, the collaboration model that was followed (i.e., on the basis of "pair-programming" model) and the corresponding scaffolding questions seemed to help the students rethink their beliefs, externalize their thoughts, argue about their opinion, and revise their knowledge structure.

As far as the students' opinions are concerned, their comments on the context of the activities and the simulation program were positive. Indicative comments include, "I had the chance to do something which helped me to understand how the cache memory works"; "The detailed visualization supported by the simulation program helps to understand how the operations are performed"; and "The form of collaboration model helped me to rethink my point of view and clarify various issues."

V. Concluding Remarks

This paper presented a Web-based educational setting based on principles derived from the conceptual change approach, the students learning difficulties/misconceptions, the text comprehension theory, and contemporary teaching approaches. The educational setting consists of appropriate Web text-based educational material, a cache memory simulation program, and a set of educational activities designed taking into account the students' learning difficulties. The Web-based educational material in question was designed in accordance with the Denhiere and Baudet's [8] text comprehension theory, aiming at facilitating students to reorganize their framework theory and resolve any inconsistencies. In order to help students acquire a better understanding of the cache memory operation, the results of the evaluation of existing simulation programs and of the students' learning difficulties and concepts were taken into consideration in the specification of the design principles of the cache memory simulation program. The simulation program uses relevant real sizes of cache memory and main memory; supports the visualization of the internal structure of the CPU, the cache memory, the operations performed, etc.; and enables students to set various parameters, such as the selection of main memory address, the selection of the command to be executed, the replacement strategy, etc. A number of educational activities were designed, aiming to help students with the conceptual change regarding the concept of the cache memory. The design of these

educational activities exploits characteristics from contemporary teaching approaches, enabling the students to explore on their own the functional characteristics of the cache memory by experimenting with the simulation program and to learn by collaborating in groups.

The results obtained from the application/evaluation of a set of educational activities have been encouraging, indicating that the simulation program and the context of the activities can effectively support the learning process and enhance learning. However, the use of the above educational setting in real classroom conditions under longer periods of time is considered necessary. Moreover, the enrichment of the simulation program with additional visualization facilities (e.g., visualization of the program counter) and the provision of adaptive capabilities to fulfill students' needs and provide personalized support are under consideration.

REFERENCES

[1] A. Clements, "The undergraduate curriculum in computer architecture, computer architecture education," *IEEE Micro*, vol. 20, no. 3, pp. 13–22, May–Jun. 2000.

[2] A. Das, "Computer organization teaching—Create your own workbook approach," in *Proc. IEEE SoutheastCon*, 2000, pp. 193–196.

[3] D. C. Hyde, "Teaching design in a computer architecture course," *IEEE Micro*, vol. 20, no. 3, pp. 23–28, May–June 2000.

[4] L. Cassel *et al.*, "Distributed expertise for teaching computer organization & architecture," *ACM SIGCSE Bull. (*Column: ITiCSE 2000, Working Group Report*)*, vol. 33, pp. 111–126, Jun. 2001.

[5] M. Grigoriadou, M. Toula, and E. Kanidis, "Design and evaluation of a cache memory simulation program," in *Proc. 3rd IEEE Int Conference on Advanced Learning Technologies*, 2003, pp. 170–174.

[6] H. Grunbacher, "Teaching computer architecture/organization using simulators," in *Proc. Frontiers in Education Conf.* , vol. 3, 1998, pp. 1107–1112.

[7] S. Vosniadou and V. Kollias, "Using collaborative, computer-supported, model building to promote conceptual change in science," in *Powerful Learning Environments: Unravelling Basic Components and Dimensions. Advances in Learning and Instruction*, E. De Corte, L. Verschaffel, N. Entwistel, and J. Van Merrienboer, Eds. Amsterdam, The Netherlands: Elsevier, 2003, pt. 11.

[8] G. Denhiere and S. Baudet, *Lecture, comprehension de texte et science cognitive*. Paris, France: Presses Universitaires de France, 1992.

[9] P. J. Smith, A. diSessa, and J. Roschelle, "Misconceptions reconceived: A constructivist analysis of knowledge in transition," *J. Learning Sciences*, vol. 3, pp. 115–163, 1993.

[10] S. Vosniadou, "Conceptual change in science learning: From coherence to fragmentation," in *Proceedings of the 21st Annual Meeting Cognitive Science Society*. ser. Scientific Explanation, Systematicity, and Conceptual Change, D. R. Kaufman, A. diSessa, P. Thagard, L. Gleitman, and A. K. Joshi, Eds. Mawah, NJ: Lawrence Erlbaum Associates, 2000, pp. 5–9.

[11] J. L. Hennessy and D. A. Patterson, *Computer Architecture a Quantitative Approach*, 2nd ed. San Mateo, CA: Morgan Kaufmann, 1996.

[12] D. A. Patterson and J. L. Hennessy, *Computer Organization & Design*, 2nd ed. San Mateo, CA: Morgan Kaufmann, 1999.

[13] M. Grigoriadou and K. Papanikolaou, "Learning environments on the Web: The pedagogical role of the educational material," in *Themes in Education*. Athens, Greece: Leader Books, 2000, vol. 1, pp. 145–161.

[14] Educational Material for the Computer Architecture Course, Dept. of Informatics and Telecommunications, Univ. of Athens, Athens, Greece. [Online]. Available: http://www.di.uoa.gr/architecture

[15] Memory Hierarchy Design, Dept. of Computer Science, Iowa State Univ.. (2004). [Online]. Available: http://www.cs.iastate.edu/prabhu/Tutorial/CACHE/mem_title_2.html

[16] H. A. Ramadhan, "Programming by discovery," *J. Comput. Assisted Learning*, vol. 16, pp. 83–93, 2000.

[17] S. Vosniadou. (2001) *How Children Learn*. Educational Practices Series, 7, The International Academy of Education (IAE) and the International Bureau of Education (UNESCO). [Online] . Available: http://www.ibe.unesco.org/International/EducationalPractices/prachome.htm

[18] V. Aleven and K. Koedinger, "An effective metacognitive strategy: Learning by doing and explaining with a computer-based cognitive tutor," *Cogn. Sci.*, vol. 26, pp. 147–179, 2002.

[19] A. Gogoulou, E. Gouli, M. Grigoriadou, and M. Samarakou, "ACT: A Web-based adaptive communication tool," in *Proceedings of Computer Supported Collaborative Learning 2005: The Next 10 Years!*, T. Koschmann, D. Suthers, and T. W. Chan, Eds. Mahwah, NJ: Lawrence Erlbaum Assoc., 2005, pp. 180–189.

[20] R. Lischner, "Explorations: Structured labs for first-time programmers," in *Proc. ACM SIGCSE Conf.*, 2001, pp. 154–158.

[21] L. Williams and R. L. Upchurch, "In support of student pair-programming," in *Proc. ACM SIGCSE Conf.*, 2001, pp. 327–331.

Maria Grigoriadou received the B.A. degree in physics from the University of Athens, Athens, Greece, and the M.Sc. and Ph.D. degrees in computer science from the University of Paris VII, Paris, France, in 1975.

She is currently an Associate Professor of computer science and Director of the Educational and Language Technology Laboratory at the Department of Informatics and Telecommunications, University of Athens. She has authored over 100 technical papers. She has worked in the fields of didactics of informatics, intelligent tutoring systems, adaptive educational hypermedia systems, student and open learner modeling, Web-based adaptive tools, educational software, and natural language processing.

Evangelos Kanidis received the B.A. degree in mathematics from University of Athens, Athens, Greece, the B.S. degree in computer science from the Technological Education Institute (TEI) of Athens, Athens, Greece, and the M.Sc. degree in advanced information systems from the University of Athens. He is currently working toward the Ph.D. degree in the area of didactics of informatics at the Inter-University Program of Basic and Applied Cognitive Science, Athens, Greece. His Ph.D. research focuses on students' conceptions on computer cache memory, the influence of textbook material in students' understanding, and the development of the appropriate Web-based educational material.

His research interests focus on applying cognitive science principles to enhance the teaching of computer internal operations.

Agoritsa Gogoulou received the B.A. degree in computer science from the University of Crete, Greece, and the M.Sc. degree in advanced information systems from the University of Athens, Athens, Greece. She is currently working toward the Ph.D. degree in the area of didactics of informatics at the Department of Informatics and Telecommunications of the University of Athens. Her Ph.D. research focuses on the development of Web-based adaptive tools supporting exploratory and collaborative learning in informatics.

Her research interests lie in the areas of didactics of Informatics, computer-based collaborative learning environments, Web-based adaptive and intelligent educational systems, learner modeling, and assessment methods and tools.