The First International Conference on Computer and Advanced
Technology in Education. March 18-20, 1996, Cairo, EGYPT.

AN INTELLIGENT TUTORING SYSTEM FOR THE YOURDON REAL-TIME SYSTEMS DEVELOPMENT METHODOLOGY

Vasilis C. Gerogiannis[1,2], Ioannis D. Zaharakis[1,2], Achilles D. Kameas[2], Panayotis E. Pintelas[1,2]

[1]Department of Mathematics, Division of Computational Mathematics & Informatics, University of Patras, HELLAS

[2]Educational Software Development Laboratory, Department of Mathematics, University of Patras, HELLAS

Abstract

TeachReal is an Intelligent Tutoring System (ITS) for training in a structured development methodology for real-time applications. Structured methodologies for real-time systems have been successfully used in real-life industrial applications for at least a decade. TeachReal helps real-time systems engineers in understanding and applying the Yourdon Structured Method (YSM), a methodology which is widely recognized as one of the most comprehensive approaches in the literature. This methodology combines all the fundamental characteristics of other similar approaches in a practical, educational and easily applicable framework. TeachReal has been developed using MELLON, an ITS generator that supports the design, implementation, evaluation and revision of ITSs which specialize in methodology training. The main characteristics of TeachReal is that it combines computer coaching and simulation-based training. It follows a three-stage training cycle: during the first stage, it provides real-time engineers with a step-by-step guide to the structured development methodology, by using an expert training system; the second stage presents, under the control of another expert system, supportive and explanatory information on the context of methodology application, while the third stage supports the self-evaluation of the acquired skills.

1. Introduction

The evolution of CBT systems during the past two decades that led to the development of Intelligent Tutoring Systems (ITSs). ITSs incorportate Artificial Intelligence principle, comprise several expert modules and apply knowledge representation techniques to all elements of the tutoring process (e.g., teaching domain, instructional strategies, student modeling) [5,10,12].

In this paper, TeachReal, and ITS for training in the use of the Yourdon Structured Method (YSM) [14], a structured methodology for real-time systems development, is presented. YSM has been recently described based on a tutorial guide [6]. and combines all the fundamental characteristics of other similar structured approaches in a practical and easy-to-be-applied and learned framework. Although structured methodologies involve more time and effort in the initial development phases, they have been applied with a great success in actual large-scale insustrial applications.. This is particularly true in the case of complicated real-time systems, where a structured methodology provides a set of communication tools (models) that everybody involved in the development (software, hardware and quality assurance engineers, project managers, customesrs etc.) can understand. In addition, these methodologies provide structured procedures in order to identify and organize the system requirements, and specify the functionality, the dynamics and the hierarchy of the system components. According to YSM, the development of a real-time system is a repetitive process that involves two contexts: analysis and design. During analysis, the system is viewed as network of processes and stored data. Analysis produces an essential model which describes how the system should operate within its environment in terms of data flow, state trasnsition and entity relationship diagrams. The aim of design is to transform this essential model into an implementation one, by considering the underlying hardware and software configuration (i.e., hardware and software-based design) and defining a model of the actual procedures to be implemented (i.e., program design).

TeachReal was developed by using MELLON, a prototype of the GENITOR ITS generator [9]. MELLON suports the design, implementation, evaluation and revision of appliations specializing in methodology training. The applications developed with MELLON attempt to transfer to the trainees two kinds of domain knowledge: procedural skills on how to apply a certain methodology and the corresponding declarative knowledge that provides the theoretical background to support it. A methodology is any procedure that consists of distinct, partially ordered tasks, actions to carry out each task and results of each action [15]. Therefore, MELLON produces intelligent training applications in subjects that are not necessarily related with each other, unlike systems like GUIDON [4], SEDAF [1]. or SOPHIE [2], which generate and solve different problems in a specific teaching subject domain. Intelligent tutoring applications developed with MELLON are stand-alone, since they may be used independently of MELLON, and encompass in simulation-based training the solution of computer coaching [11], combined with informal discovery learning in a gaming environment [3].

In the next section, the authoring process in MELLON is presented, followed by a short description of the YSM, which constitutes the training subject. Successively, the representation of the training subject, the instructional strategy in TeachReal and the way the application can be used are described. Concluding, the experience attained from the application usage and the future research directions of the authors are discusses.

2. Authoring with MELLON

Authoring of a training application is a sequential process, during which authors describe the domain knowledge. and specify the instructional strategy. The MELLON Authoring Subsystem includes a set of tools controlled by the Authoring Subsystem Manager, which is responsible for monitoring all the authoring actions.

In a MELLON application, declarative knowledge consists of Learning Units (LUs) which are elementary blocks of knowledge described with a set of attributes (title, media type, storage path etc.). LUs can be used independently of any instructional strategy and contain highly unstructured information. Thus, no traditional database system could be used to archive LUs satisfactorily. As a result, with each application, a domain base is implemented to store the associated Lus.

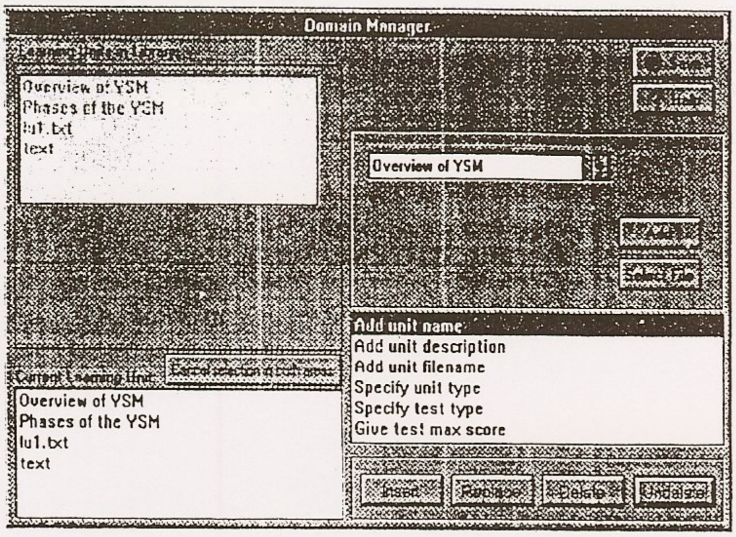


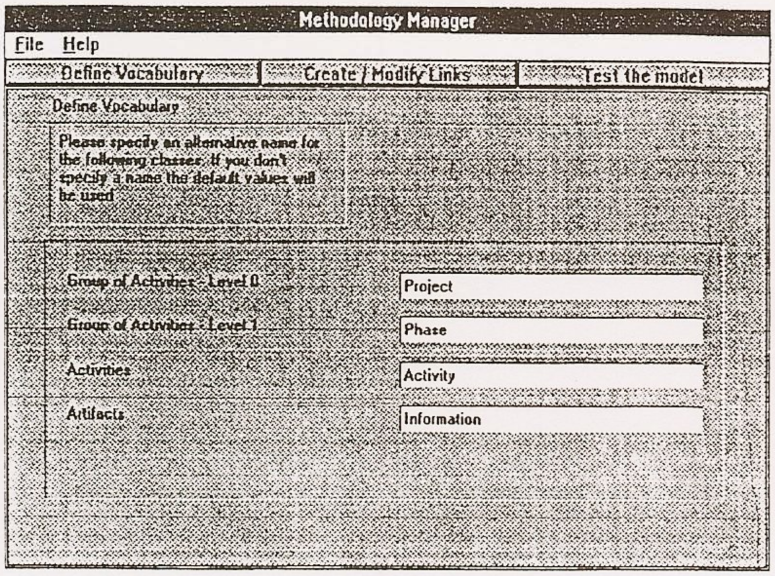
Figure 1: Domain Manger

Lus can be developed independently of any application and stored in a reusability base. The declarative knowledge of the application is constructed by combining monomedia Lus. Authors have to relate Lus to the application context by using the Domain Manager (Figure 1).

MELLON includes a Methodology Expert System (MES) which is used for both construction and execution of the methodology [15]. The Methodology Manager (Figure 2) is the authoring interface of MES. Using it, authors can perform an elementary knowledge engineering process, in order to describe the structure (static contents) of the methodology, the special vocabulary used by the methodology at-hand, and the ordering of the tasks and actions (dynamic behaviour of the methodology) MES produces at authoring time the inheritance relations among the various elements of the methodology, orders its tasks and actions at run time, and decides which one should be carried out next.

The learning scenario embedded in an application developed with MELLON is called learning cycle and is made up of stages. A stage represents some basic tutoring approach and constitutes an integral application module with respect to instruction. Authors have first to define the learning cycle of the application by specifying which stages it includes, and subsequently to specify the type and configuration of each stage.

Finally, since all tools will have to be used for the achievement of the ultimate goal (the development of the contents of a training application) a common interaction metaphor is used for each [8].



Please specify an alternative name for the following classes. If you don't specify a name the default values will be used

Group of Activities - Level 0: Project

Group of Activities - Level 1: Phase

Activities: Activity

Artifacts: Information

Figure 2: Methodology Manager

3. The Structured Development Methodology

YSM, as it is described in [6], combines all the basic characteristics of other similar approaches [7, 13] in a flexible, easily applied and tutorial framework. The methodology does not force real-time engineers to follow strict step-by-step guidelines, since not all of its steps are relevant in all developments and its main characteristic is the use of models (expressed by diagrams), which serve as communication means among all persons involved in the development phases. The proposed methodology aims to capture the complexity issues that often appear in real-time systems. Real-time processes often communicate asynchronously and execute under strict timing constraints. In order to meet these issues, YSM proposes the hierarchical decomposition of a real-time system into component (concurrent or sequential) processes.

YSM, as taught by TeachReal, consists of four major phases (Figure 3): Structured Analysis, Processor Environment Modeling, Software Environment Modeling, and Code Organization Modeling. The first one belongs to analysis context, while the other three make up the design context. Structured Analysis phase produces the Essential Behavioural Model (EBM), which describeds what the system will look like from the user's point of view and captures its fundamental requirements. PEM refers to choosing the appropriate physical processors onto which the product will be implemented, allocating the EBM to the physical processors, and defining the Human Computer Interface. The SEM phase comprises three activities: evaluation of a software architecture, removal of "perfect technology" and software allocation. Finally, during the COM phase, a detailed hierarchy is associated with the already produced models (Data Flow, State Transition and Entity Relationship Diagrams) and a structure chart is used as a mean for documentation and communication. The produced model is further refined in such a way that it will give an implementable and well-structured program.

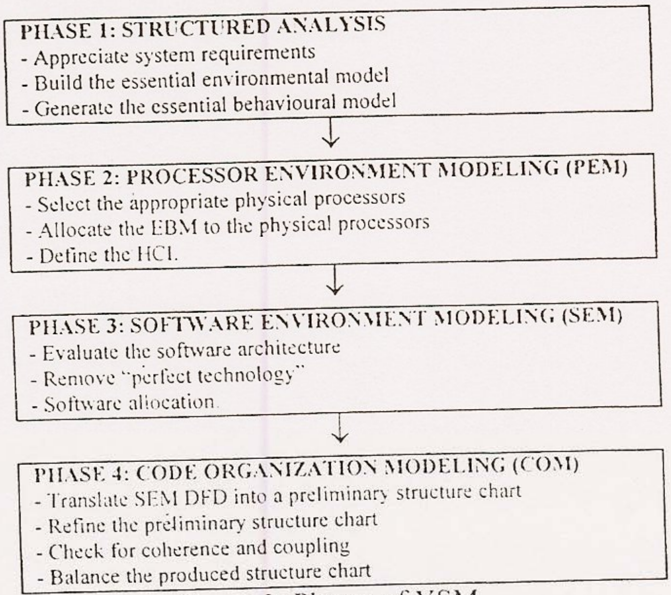

PHASE 1: STRUCTURED ANALYSIS
- Appreciate system requirements
- Build the essential environmental model
- Generate the essential behavioural model
PHASE 2: PROCESSOR ENVIRONMENT MODELING (PEM)
- Select the appropriate physical processors
- Allocate the EBM to the physical processors
- Define the HCI.
PHASE 3: SOFTWARE ENVIRONMENT MODELING (SEM)
- Evaluate the software architecture
- Remove "perfect technology"
- Software allocation
PHASE 4: CODE ORGANIZATION MODELING (COM)
- Translate SEM DFD into a preliminary structure chart
- Refine the preliminary structure chart
- Check for coherence and coupling
- Balance the produced structure chart

Figure 3: Phases of YSM

4. LU Development

The major source that has been used for LU development was the description of YSM as it appears in the bibliography. In particular, Goldsmith [6] synthesizes all characteristics of the Yourdon Analysis/Design in a generalized framework that can be used for educational purposes. TeachReal contains three types of LUs: text pieces, diagrams and tests. Text LUs provide definition, description and explanation of methodology basic concepts and decisions, while diagrams summarize and visualize certain procedures involved in the methodology.

Tests are used for evaluation of the acquired knowledge. They are scored according to their type and permit the gathering of GREs (Grades of Excellence), which can be used as a measure of the trainees' understanding, and at the same time give TeachReal a feel of a gaming environment. The system supports several types of tests: Steady tests and Decreasing tests consist of multiple choice questions; Gamble tests consist of true/false questions. With each test, the authors have associated a maximum score of GREs that trainees can get by answering correctly to all test questions. The actual GRE socre depends on the number of correct questions answered with respect to the total number of test questions, and on the test type. The maximum obtainable score for Decreasing tests decreases with the number of times that the trainees take the test. This score remains constant for tests of type Steady. For tests of type Gamble, the trainees can obtain a double maximum score, if they answer all questions correctly in the first time; otherwise, these tests behave as Decreasing tests.

5. Instructional issues - Working with TeachReal

The learning cycle of TeachReal consists of three stages: Introduction to YSM, Clarification of concepts. and Self-evaluation. This scenario aims at presenting the procedural and the declarative knowledge that relate to YSM, and then at leading the trainees through a self-evaluation process.

During the first stage (Introduction to YSM), the trainees are coached through a simulation of the methodology, in order to acquire the basic procedural skills. This stage aims at helping the trainees discover the correct evolution of the methodology, while obtaining a basic theoretical knowledge. The objective of the second stage (Clarification of concepts) is to permit the trainees review the declarative knowledge that supports the application of YSM. Finally, the Self-evaluation stage aims at providing the trainess with a means of self-evaluation of the expertise they acquired by using TeachReal.

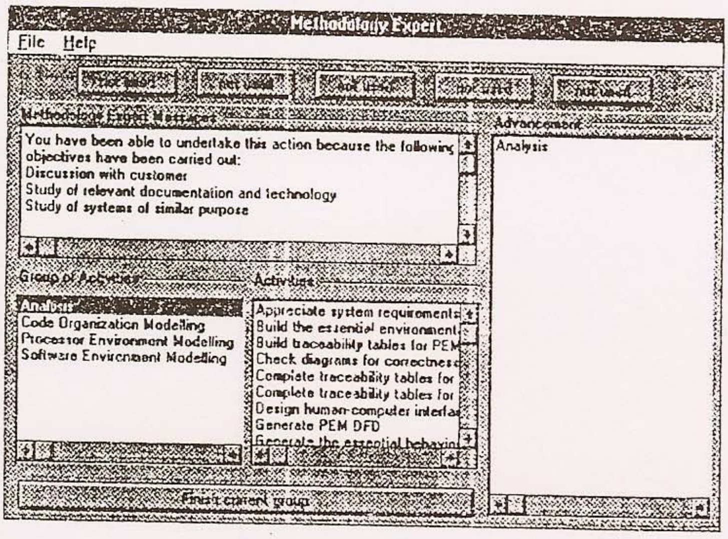
In order to ensure that by the end of training, the trainees will have acquired a minimum level of expertise, execution prerequisites and termination criteria are associated with each stage. In fact, since the first two stages aim at presenting the domain knowledge that relates to the structured methodology and the methodology itself, no pedagogical prerequisites exist for their execution. In addition, the trainees may quit each of them at any time and resume their position in a future session. The last stage, however, requires that at least 75% of the tests of the second have stage been taken by the trainees, in order for it to be available, and has to be completed in one session. Each stage manages its own environment and determines its state of completion. The user interface of any stage contains at most three areas: procedural, declarative and utility.

The development of a real-time system using a structured methodology is a complex and iterative prodcedure, which, in many cases, is associated with a learning curve. Lack of experiene in the specific activities which are required to complete each step of the methodology introduces problems in the smooth running of a project and consumes additional manpower. The main objective of TeachReal is to minimize this learning curve by systematically providing real-time designers with the necessary knowledge that will help them exercise their skills.

TeachReal operates in two modes: training and administrator. Training with the use of TeachReal is carried out at trainees' pace and initiative. The application starts with the trainee registration, during which each trainee has to register his/her name, which is used for the creation and naming of the files that make up the student model. The application maintains an elementary student model for each trainee, where identification, history and performance data are recorded by the two expert systems. This data is reset when trainees restart the application. The training administrator can access this information through a special environment provided with every MELLON application, the Administrator Mode. This environment can be accessed using a special, author-defined password. The functions offered in this mode include reviewing trainees' performance, changing a trainee's registration name and removing a trainee's session record from disk.

Then, the application presents the trainees with a menu containing all the pedagogically available stages, since, as has been explained earlier, the basic training unit is the stage. In the beginning, only the first two stages are available, which may be reviewed in any order, since there exists no pedagogical prerequisite for their execution. The stage menu is displayed after each stage is completed. Using it, each trainee can determine an individual learning trajectory, always within the limits set by the authors, by selecting the next stage to be exectued. In addition, trainees may repeat the last stage, restart, or exit TeachReal, by pressing the appropriate buttons on the stage menu.

During the first stage (Introduction to YSM), MES acts as a computer coach that guides the trainees through a simulation of the phases of the structured methodology. The latter is presented as a simplified version of problem-solving: the problem posed each time is to find the correct next action that should be carried out in in order to advance inside the methodology. The trainees have to choose from a list of all the actions, based on the results of the actions carried out so far, and on the goals that have to be achieved at this time. In Figure 4, the initial screen of the first stage is shown, which contains the list of all pahases of YSM. The trainees must pick the first phase of the methodology ('Analysis'). MES constantly monitors their responses, providing help and explanation both on mistakes and on correct moves (Figure 4). Help consists of three levels: explaining the expected outcomes of the correct move, describing the outcomes of the selected move, and naming the correct move. In addition, the trainees can use specific buttons in order to review the collective results of the actions they selected in the past, ask MES to carry out the simulation itself with a justification of its selection (s), access a glossary of methodology-related terms, or request access to a library of methodology-related topics. In order to pedagogically complete this stage, the trainees have to correctly exectue the entire methodology, although they can quit the stage at any time.



You have been able to undertake this action because the following objectives have been carried out:
Discussion with customer
Study of relevant documentation and technology
Study of systems of similar purpose

Analysis

Analysis
Code Organization Modelling
Processor Environment Modelling
Software Environment Modelling

Appreciate system requirements
Build the essential environment
Build traceability tables for PEM
Check diagrams for correctness
Complete traceability tables for
Complete traceability tables for
Design human-computer interfa
Generate PEM DFD
Generate the essential behavio

Figure 4: Introduction to YSM

In the second stage (Clarification of concepts), the LUs of the declarative knowledge are partitioned into sets on a per methodology pahse basis. The trainees may review the LUs by first selecting the application pahse of interest, and then by selecting one out of a list of LUs.

After the first two stages are completed (and only then), TeachReal displays the third and final stage (Self-evaluation) in the stage menu. It is reminded that the pedagogical prerequisite for the execution of this stage is that the trainees should have thoroughly reviewed both procedural and declarative knowledge. That is, they have been through the first two stages, and now they feel competent with the procedural knowledge (steps of YSM), as well as the declarative knowledge (methodology theory and terminology) they have acquired so far. During this stage, MES again guides the trainees through a simulation of the structured methodology. This time however, MES judges the responses of the trainees, whose GRE score is lowered every time they make a mistake. After a methodology phase is successfully completed, a pahse-revision test is presented. Again, the GRE score is decreased with every wrong answer. During the stage, the trainees have access to the topics library, but no help or explanation is provided by MES. In order to successfully finish this stage (and the application), the GRE score of the trainees must not fall below zero. If by the end of the stage, a trainees score remains greater than zero, it can be certified that he/she has acquired the basic skills necessary for the design of real-time systems. On the contrary, if their score reaches zero before the end of the stage, then they have not acquired a satisfactory level of expertise (by TeachReal standards), and the system suggests that they restart the application and carry out one more training session.

6. Conclusions

The functional view and the instructional issues of TeachReal, an ITS for training in the use of structured development methodology for real-time systems, have been presented. TeachReal was developed with MELLON, an ITS generator that specializes in the production of methodology-teaching intelligent applications. The use of an ITS generator greatly facilitated application development. Since MELLON had already been used for the development of several training applications, no unsurpassable development issues came up.

The development of an application using MELLON involves the description of the methodology to be taught (YSM in this case), the declarative knowledge that supports application of the methodology, and the instructional strategy to be used. During TeachReal development, the longest phase of these proved to be the construction of the LUs of the declarative knowledge. Although most LUs were obtained from the bibliography, it took about 12 personweeks to compile the declarative knowledge, which consists of more than 80 LUs. The description and prototyping of the methodology lasted 3 personweeks, while the definition of the stages of the learning cycle took about 2 personweeks. Finally, prototyping and validation of the application took another 3 personweeks, amounting to a total of 20 personweeks for the complete application development.

TeachReal is executed under Microsoft Windows and requires a minimum of 1 Mb RAM. The authors of TeachReal are now developping intelligent applications that teach other real-time systems development methodologies (e.g., Coad & Yourdon, HOOD, Booch etc.). The library of intelligent applications which will be produced, will stand as a complete training framework for real-time engineers.

7. REFERENCES

[1] AIELLO, L. and MICARELLI, A.: 'SEDAF: An intelligent educational system for mathematics. Applied Artificial Intelligence, 1990. 4(1), pp 15-37.

[2] BROWN, J.S., BURTON R.R. and CLANCEY, W.J.: 'Pedagogical, natural language and knowledge engineering techniques in Sophie I, II and III'. In Sleeman and Brown (eds.) Intelligent Tutoring Systems, (Academic Press, 1982), pp 227-282.

[3] BURTON, R.R. and BROWN, J.S.: 'An investigation of computer coaching for informal learning activities'. In Intelligent Tutoring Systems, Sleeman and Brown, Eds. (Academic Press, 1982), pp 79-98.

[4] CLANCEY, W.J.: 'Medthodology for building an Intelligent Tutoring System'. In Kearsley, G.P. (ed) Artificial Intelligence and Instruction, (Addision-Wesley, 1987), pp 193-227.

[5] GEROGIANNIS, V., GIAKOVIS, D., PINTELAS, P. and KAMEAS, A.: 'Intelligent Systems for Education: an overview'. Technical Report TR 93-01, Dept. of Mathematics, Univ. of Patras.

[6] GOLDSMITH, S.: 'A practical guide to real-time systems development', (Prentice-Hall, 1993).

[7] HARTLEY, D. and PIRBHAI, I.: 'Strategies for real-time system specification', (Dorset House, 1987).

[8] KAMEAS, A., GEROGIANNIS, V., DIPLAS, K. and PINTELAS, P.: 'Encapsulating multiple perspectives in interaction specification'. Proc. EUROMICRO 94, Liverpool, U.K., (IEEE Comp. Society Press, 1995) pp 463-469.

[9] KAMEAS, A. and PINTELAS, P.: 'The functional architecture and interaction model of a GENerator of Intelligent TuTORing application'. Journal of System and Software, to appear in 1996.

[10] KAPLAN, R. and ROCK, D.: 'New directions for Intelligent Tutoring'. AI Expert, Feb. 95, pp 31-40.

[11] NAWROCKI, L. H.: Artificial Intelligence applications to maintenance training. In Artificial Intelligence and Instruction, Kearsley, Ed. (Addison-Wesley, 1987).

[12] RICKEL, J. W.: 'Intelligent Computer-Aided Instruction: a survey organized around system components'. IEEE Trans. on Systems, Man and Cybernetics, 1989, 19(1), pp 40-57.

[13] WARD, P. and MELLOR, S.: 'Structured development for real-time systems, Three volumes', (Yourdon Press, 1985).

[14] YOURDON, E.: 'Modern structured analysis', (Prentice-Hall, 1990).

[15] ZAHARAKIS, I., KAMEAS, A. and PINTELAS, P.: 'MeT: The expert methodology tutor of GENITOR'. Microprocessing and Microprogramming, 1994, 40, pp 855-860