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

MELLON: A GENERATOR OF INTELLIGENT TUTORING APPLICATIONS

D. Giakovis[1,2], I.D. Zaharakis[1,2], C. Diplas[1,2], A. Kameas[2], and P. Pintelas[1,2]

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

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

Abstract

This work presents MELLON, an ITS Generator, which is designed for non expert authors that desire a rapid development of an ITS. An application produced by MELLON, attempts to transfer to the trainees two kinds of skills: procedural knowledge on how to apply a certain methodology and declarative knowledge as the corresponding theoretical background to support a certain methodology. MELLON consists of two subsystems. Authoring subsystem that provides tools for the applications. and Execution subsystem that supports the tutoring of the application developed using the Authoring subsystem. Execution subsystem utilizes an expert system in order to handle and transfer the procedural knowledge is accomplished using other MELLON tools such as Application Configuration Manager, Domain Manager, Instructional Manager, Text Editor and Test Editor. A trainee model is constructed and attached to the ITSs that MELLON generates. Finally, other "external" tools, which may be more convenient to the authors can be easily accommodated in MELLON due to its open architecture.

Keywords: Generators of Intelligent Tutoring Systems, Authoring Systems, Instructional Strategies.

Introduction

The application of Artificial Intelligence (AI) techniques and practices in the context of Computer Aided Instruction (CAI) has led to systems dedicated to education, which support decision making, problem solving and reasoning. These techniques embed, in most cases, Expert System as basic modules of these systems. Recently, new approaches such as Artificial Neural Networks [1], Genetic Algorithms and Fuzzy Systems in order to approximate the human

behavior during the educational process have emerged. These systems are called Intelligent Tutoring Systems (ITSs) [6]. Such a system, includes three kinds of knowledge, the knowledge concerning a specific teaching domain (Domain Knowledge), the knowledge about student's present knowledge (student model) and the knowledge about the instructional strategy that will be applied especially for each student (instructional strategies model) [2].

The last step in the evolution of ITSs is the authoring systems of ITSs; these systems are called Generators of ITSs. An ITS Generator offers highly interactive environments to authors, in order to describe and structure the knowledge by themselves, allowing them to perform task and domain analysis, requirements specification and prototyping, design of knowledge bases, reasoning mechanisms, student-system interaction, knowledge bases creation, testing, extension and maintenance of the knowledge bases. However, most of the efforts towards the development of generic systems have not yet reached satisfying results, since the problem of representing and understanding both instructional and learning processes is still open [5]. Moreover, the cooperation among researchers of Cognitive engineering, Psychology, Pedagogical science, Informatics and Artificial Intelligence, is needed for the development of functional and reliable ITSs.

In this paper an ITS Generator, called MELLON is presented. It produces applications (ITSs) which teach methodologies. As mentioned in [7], a methodology, is any procedure that consists of distinct, partially ordered tasks, actions to carry out each task and results (outcomes) of each action. It was designed with the objective of addressing the needs of non-computer expert tutors who need to develop an efficient training course as fast as possible. MELLON provides authors with the necessary tools for the construction of a training application and uses in combination declarative and procedural knowledge; as is considered the necessary topics that must to be acquired by the trainees, while procedural knowledge is considered the way that the declarative knowledge will be applied [8]. These two kind of knowledge must be used in combination. An application constructed with MELLON is equivalent to the sequential execution of a set of pedagogically integral units

called stages. Each stage is composed of a number of pedagogically autonomous objects called learning units (Lus) and encapsulates an instructional strategy by having an internal structure that reflects the strategy's principles. Thus, stages are used to represent the instructional semantics of the application, while learning units correspond to integral blocks of knowledge that must be presented to the learners.

The rest of the paper is organized as follows. In the next section the architecture of MELLON is presented with regard to each tool's and module's functionality. Then, the user interface of the system and the instructional issues are introduced and finally the conclusions with the further extensions of this work are discussed.

Architecture of MELLON

MELLON consists of two subsystems, the Authoring Subsystem and the Execution Subsystem, in order to separate the authoring and the prototyping processes. The Authoring Subsystem supports the development of intelligent training applications by providing a set of highly interactive tools, each of which supports an integral development phase. Using Authoring Subsystem authors can construct all the files and organize the file dependencies, that make up an application. The Execution Subsystem supports the execution of an application constructed with the Authoring Subsystem. It is made up of all the run-time programs that access and execute the files produced with the authoring subsystem tools. It also contains programs that initialize the application and the trainee environment. Additionally, MELLON embodies an expert system, called methodology expert system, which enables authors to specify all the methodology elements (activities, group of activities, artifacts), as well as the relationships among them.

Authoring Subsystem

The Authoring Subsystem supports the application construction process. A MELLON application is made up of Lus; with this subsystem, the author can specify these units. To do this, the Authoring Subsystem consists of a set of

internal and external tools which are provided to the authors in order to support each phase of the authoring process. More specific, the internal tools are:

The Application Configuration Manager (figure 1), which is used for the specification of general application parameters such as title, author, password etc., and is responsible for the overall control of authoring actions.

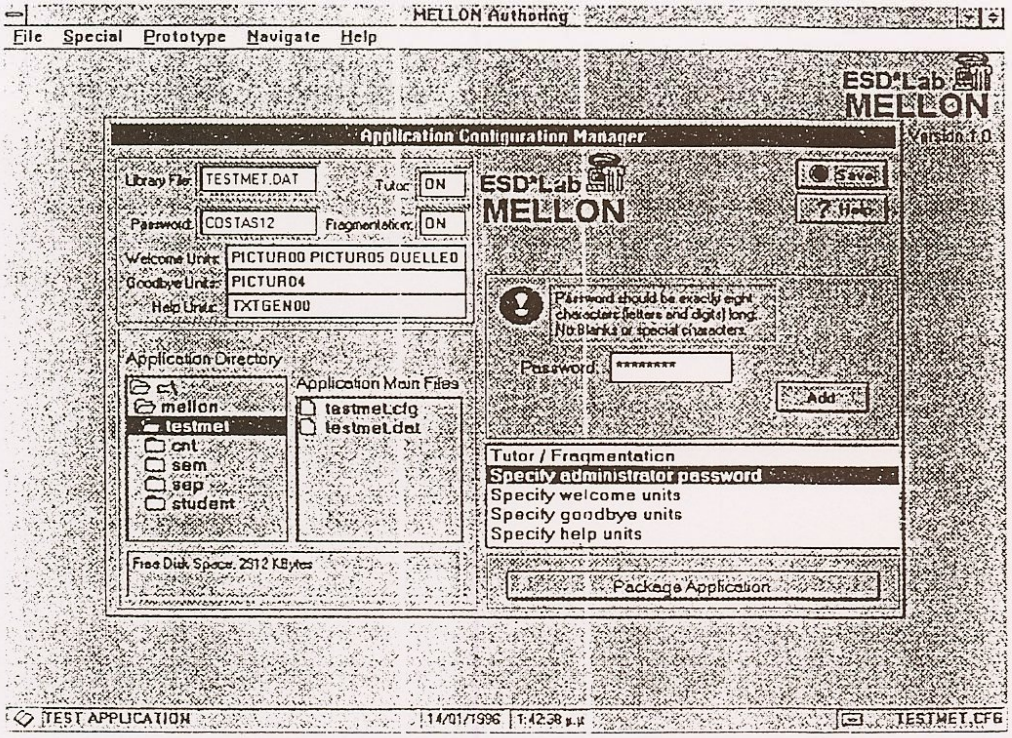

Figure 1: The Application Configuration Manager

The Domain Manager is used for the specification of the type (for example text, test, image, external) of Lus that make up the application declarative knowledge and the contents of the LU library contained in the application.

The instructional Manager, which is used for the definition of the instantiation of the stage templates that will be included in the application. It specifies the application-dependent attributes of the stage, like title and configuration, or the functions of the strategic states that will be included in the stage environment.

The Discourse Manager, which is used to specify the learning cycle of the application. This produces the application script file, together with the prerequisites and termination criteria of each stage.

The domain Combination Manager, which can be used to combine procedural and declarative knowledge.

The Stage Domain Manager, which is used for the definition of the part of LU library that will be accessible by the trainees in current stage; this tool functions as the interface among the stage-related tools and the rest Authoring System tools, the Stage Methodology Manager which is used for the association of stage library LUs and test LUs with the phases of the methodology and the Stage Interface Manager, which is used for the assignment of tutoring actions to the buttons used by the system.

The Methodology Manager (figure 2) supports the description or modification of the model of the methodology which will be taught and the execution of this model in order to verify and validate it during the authoring process. It determines the structure of the methodology simulation that will be presented to the trainees and at the same time produces its visualization elements.

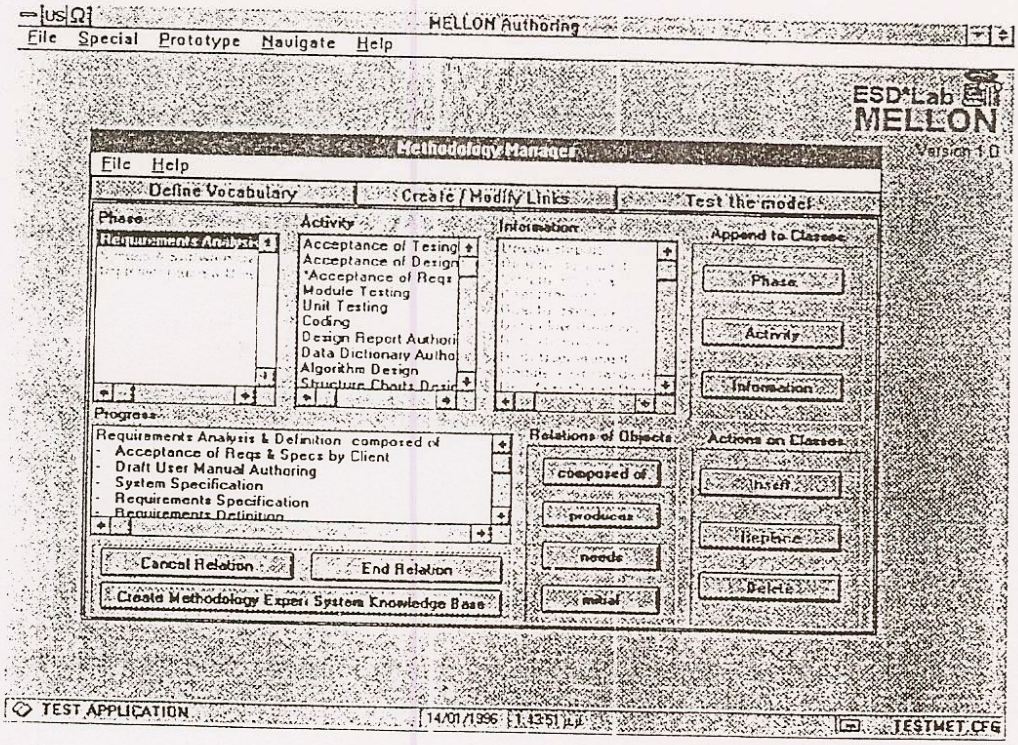

Figure 2: The Methodology Manager

The external tools are the Text Editor and the Test Editor. The Text Editor, is a multiple document interface (MDI) editor and may be used in the construction of text types LUs. In place of the Text Editor any other text editor can be used, as long as a certain text format is preserved.

Finally, the test editor may be used to construct tests where each test is made up of any number of questions; this tool also includes a test prototyping facility.

Execution Subsystem

The Execution Subsystem, supports the execution of the application, which has been created by the Authoring Subsystem. It is included with every application package, although one such set of modules is required for the execution of any number of training applications. Execution Subsystem may also be called from within Authoring Subsystem, thus offering a prototyping capability to authors. It supports the building of a library of training application by using the same runtime support system for each of them. A result of this concept is the application menu that is presented to the application user. Every training application supports two modes of operation: tutoring and administrative. The administrator environment is made available to the user who enters the correct password as this has been specified by the authors of the application. The administrator tasks include removal of a trainee from the class of trainees, renaming the trainees, and viewing of statistics on trainees performance. With each application, a comprehensive interactive environment to support the administrator tasks is included. In tutoring mode, each trainee can follow an individual learning trajectory, always within the limits set by the author, by selecting the next stage to be executed. After execution of a stage terminates, the system presents the trainees with a list of all the stages that have been executed, having the next stage in the sequence highlighted. The trainees can accept the proposal of the system, make a different selection or restart the application.

For each stage, the suitable run-time program which constructs the stage environment is executed. MELLON supports four types of stages: Coach,

which aims at helping the trainees discover the correct evolution of the methodology, Exploration, which supports the trainees through a self controlled exploration of the declarative knowledge of the domain, Judge, providing an objective, system controlled evaluation of both the declarative and procedural knowledge acquired by the trainees and Consultation, providing access to the correct methodology evolution and to the test answers [4].

The Methodology Expert System (figure 3) controls the trainee performance with respect to the methodology. Methodology Expert in included with every application and is responsible for teaching the procedural knowledge of the application. In other words, it is used to present the taught methodology to the trainee and to guide him through its components; it is also responsible for correcting any misconceptions the trainee may develop as far as the methodology is concerned. Methodology Expert operates in two modes: judging, and coaching. During judging mode, Methodology Expert leaves control of the simulation entirely at trainees hands, and simply judges their selections. In coaching mode, Methodology Expert supports a learn-by-discovery process of the methodology. The role of Methodology Expert is to coach the trainees by commenting on their selections, responding to their commands and assuming control when the trainees appear to be lost.

The User Interface of MELLON

MELLON has been designed so that it can been used even by novice users. For this reason, great care has been taken into the design of a system that effectively uses the available screen space in order to achieve better interaction between the user and the system. Two features that also enable the effective interaction are, the partition of the authoring process into discrete phases (that is data entry, specification of instructional strategies, definition of the methodology etc.) and also the support of a robust and structured model of interaction.

The authoring process is considered as the achievement of a number of macro-goals that constitute the final goal which is the construction of the application. Each tool supports the achievement of one macro-goal, while each screen window corresponds to one tool. Inside every screen window certain

areas are used to achieve even less significant goals (sub-goals). In this way, the user knows exactly the goals that must be successively achieved in order to built the application. These goals are carried through with actions by the user. The context of each action is defined automatically by the goal with which is associated.

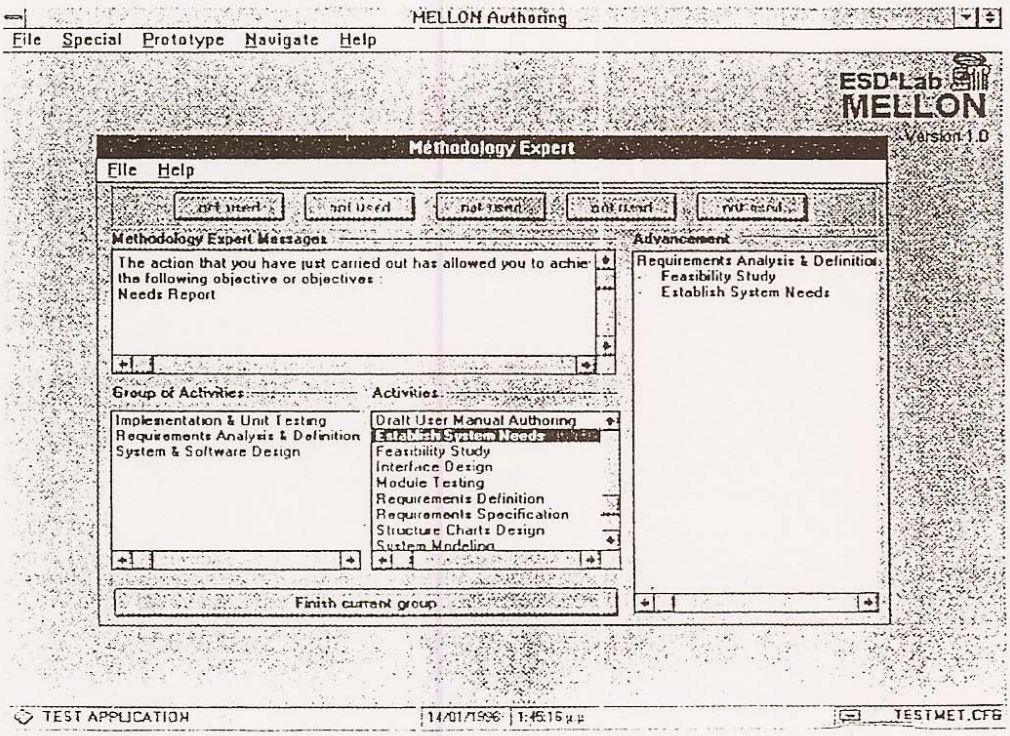

Figure 3: The Methodology Expert System

The windows are designed according to a general format for reasons of uniformity and to supply accuracy to the user. A typical tool format, where the below described areas are shown, is depicted in figure 4. In particular, each screen window contains:

- the menu bar, a line across the top of the authoring subsystem window, which contains the menu titles: FILE, SPECIAL, PROTOTYPE (special menu which has different names depending on the active tool), NAVIGATE, HELP,

- the main data area, the left-hand half of the tool window, where the file contents are displayed in an appropriate format,
- the current data area, which appears at the lower part of the main data area and contains the contents of the currently used file entry,
- the main control area, which covers the upper part of the right-hand half of the tool window. It contains the currently open control and finally,
- the auxiliary control area, which covers the lower part of the right-hand half of the tool window, and contains the control selector and a set of buttons that represent the most frequently used actions of the SPECIAL menu.
- the top data area, which may optionally appear at the upper part of the main data area,
- the status line, which displays the name of the tool currently in use and the name of the file currently open.

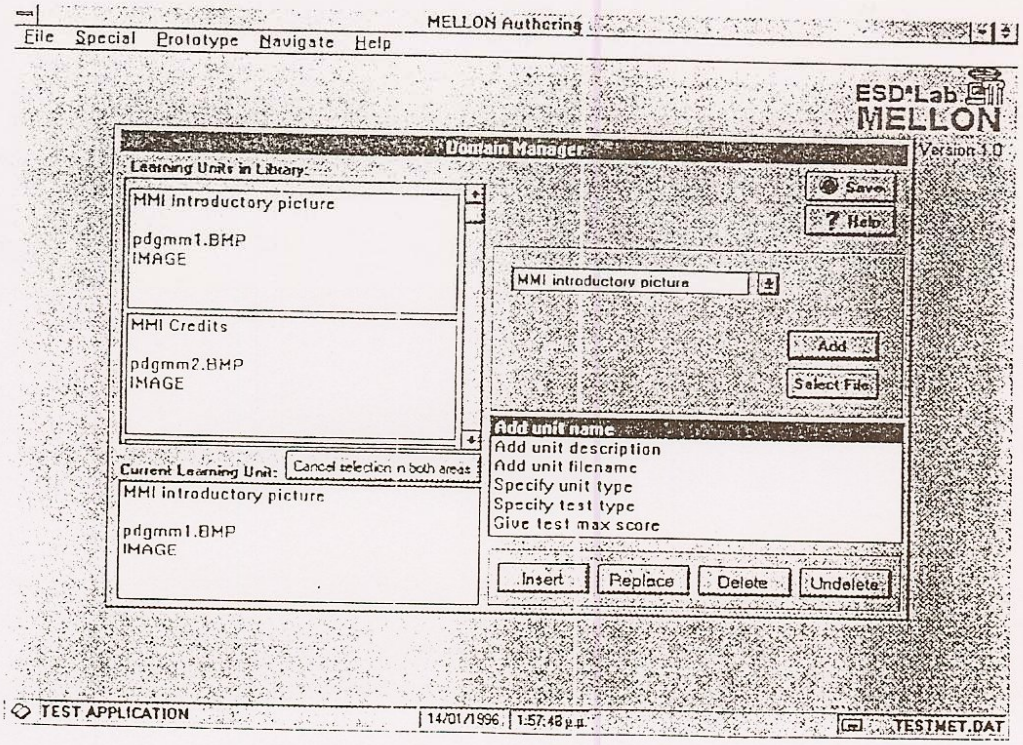

Figure 4: The Domain Manager

The user can access any tool directly through the NAVIGATE menu, All tools are accessible from within any tool, since the only prerequisite for correct execution of MELLON is the application name, which is specified upon MELLON initialization. If there are incomplete sequences authors are notified with an error message. The FILE menu contains file related operations. The SPECIAL menu contains operations that are exclusive for every tool. The PROTOTYPE menu allows the execution of the application prototype, and the HELP menu naturally provides help to the users.

Instructional Issues

Selection of learning scenario and other pedagogical strategies can be regarded as a part of the general problem of discourse management, which refers to the participation of the computer in a tutoring dialogue in equal terms with the trainees. The learning scenario embedded in an application developed with MELLON is called learning cycle and is made up of stages. The instructional base of the application consists of stages. Stages are of different type, depending on the instructional strategy each supports. Each stage offers its own user interface, through which navigation and learning actions are made available to the trainees, while information is exchanged between them and the system. MELLON consists of four stage types, which are described in previous section, each of which represents a different tutoring approach.

Each stage type includes a basic set of type-pertinent attributes; these differ among stage types, depending on the pedagogical objectives of each. Authors have to select which attributes will be activated with each instantiation of a stage type template that is included in the learning cycle of the application. These selections will also affect the facilities that the user interface of the stage will be offering to the trainees. Authors may also define the execution prerequisites and the termination criteria for each stage. The former consist of all the stages of the learning cycle that the trainees must terminate successfully in order for the stage to become available, while the latter may be based on two internal evaluation parameters: the current score and the percentage of the stage tests that the trainees have executed.

Conclusions

In this paper an ITS Generator, called MELLON was presented. MELLON consists of two subsystems, Authoring and Execution Subsystem. The Authoring Subsystem provides several tools for the authoring process while the Execution Subsystem provides the run-time support required for the execution of the produced application. The produced application is an Intelligent Tutoring System which can been used autonomously. ITSs produced by MELLON are able to teach methodologies. MELLON addresses the needs the non-computer expert authors that have to encode their expertise in an application as quickly and efficiently as possible. Our future work includes the development of the evolution of MELLON, the Generator of Intelligent Tutoring System (GENITOR) [3], with open architecture, full multimedia capabilities and workgroup capabilities.

References

[1]. Beale and Finaly, "Pattern Recognition and Classification in Dynamic and Static User Modelling", Neural Networks and pattern recognition in human-computer interaction (Beale and Finlay, (eds)), Ellis Horwood, pp. 65-89, 1992.

[2]. Gerogiannis V., Giakovis D., Pintelas P., and Kameas P., "Intelligent Systems for Education: an overview", Technical Report, TR 93-01, Department of Mathematics, Sector of Computational Mathematics and Informatics, University of Patras, Greece, 1993.

[3]. Kameas A., and Pintelas P., "The Functional Architecture and Interaction Model of a Generator of Intelligent Tutoring Applications", Journal of Systems and Software, to appear in 1996.

[4]. Pintelas P., Kameas A. and Crampes M., "Computer-based Tools for Methodology teaching", Proceeding of the 34th International ADCIS Conference, Norfolk, USA, November 8-11, pp. 341-355, 1992.

[5]. Rickel J. W., "Intelligent Computer-Aided Instruction: A Survey Organized Around System Components", IEEE Transactions on Systems, Man, and Cybernetics, 19(1), Jan-Feb 1989, pp. 40-57.

[6]. Yazdani M., "Intelligent Tutoring Systems: An Overview", Expert Systems 3(3), pp. 154-162, 1986.

[7]. Zaharakis I., Kameas A., and Pintelas P., "MeT: The Expert Methodology Tutor of GENITOR". Microprocessing and Microprogramming, NH Elsevier, 40, pp. 854-860, 1994.

[8]. Zaharakis I. D., Kameas A. D. and Pintelas P. E., "A Hybrid Expert System as an Embedded Module in Tutoring Systems". TR 95-02, Division of Computational Mathematics & Informatics, Department of Mathematics, University of Patras, Hellas.