e-learning, e-content,
MAMS, QRU units,
unit specification, Kay methodology

Jan PIECHA[*], Marcin BERNAŚ[*]

THE ON-LINE WORKS OF VIRTUAL LABORATORY EXAMPLE SOLUTIONS

The paper presents the one e-content example used as a pattern solution for distance learning resources development at the University of Silesia. The Intranet technologies were used as networking services for training processes control and future distribution services at the University of Silesia distance learning network. The given solution provides the development team at Distance Learning Technologies Centre with various unifications and library units of four courses that are under development at present in Department of Computer Systems, institute of Informatics. All the applications are running on MAMS (Multimedia Applications Management Shell) environment as a common frame structure for every e-lesson. The MAMS applications are provided with several mechanisms that allow monitoring the user knowledge level and study progress. Various examples of the applications layout and their characteristic features are presented in this contribution.

1. INTRODUCTION

The Distance Learning technologies are used in many Universities today. The University of Silesia is one of the Academic Unit that decided to put this new Internet technology into learning and teaching practices. To simplify the e-content development process several unification works were undertaken first. The elaborated platforms allows us making the system development flexible, smart, being in accordance with well known principles of the computer aided learning methods.

The learning courses run on our unique platforms: MAMS and QRU described in several papers as [1], [2], elaborated as a support tools for e-lessons solutions. They are also under a main server control package supporting the e-content distribution platform, compatible with the MAMS and QRU units, described in several papers [3], [4]. This paper describes main organisation solutions of two pattern e-learning packages.

2. THE APPLICATION STANDARDISATION

As it was already discussed the applications were elaborated within the MAMS shell provided with an own evaluation mechanisms for the course-flow analysis. All characteristics features, concerning communication with the platform, are dealt with a unique standard interfaces within the Macromedia® Authorware® environment. The courseware developer has to be only familiar with the answer string format given by the DLIE interface [4].

[*] Distance Learning Technologies Centre, University of Silesia, Poland

The courseware is usually produced by different development teams. Trying to make their works easier many standard units were elaborated. They allow to make all works not only in the defined standards but also in a simple organisation forms (unified graphic objects, a common navigation schemes for all courses, etc.).

The standards provide the developer with:

- Menu interface, the user navigation principles, help calling, etc.
- They use the same technologies for comments, explanations, presentation products exploitation, data collecting,

They use the same settings of animation speed, application background, formats, navigation buttons, used in the presentation. The application security is a part of standard interfaces. The developer is able to use the information services frames, with various objects, shapes and dimensions. The common graphic icons are used in presentation sessions (integrating the computer with a printer and other interfaces) with hyperlinks to the website. Graphic interface specification includes also description of how to implement all the items, including templates ready to use.

3. THE COURSEWARE ORGANISATION

Using the given standards the applications are put into several organisational layers. The pre-test of the application allows the user selection of the most suitable application content.

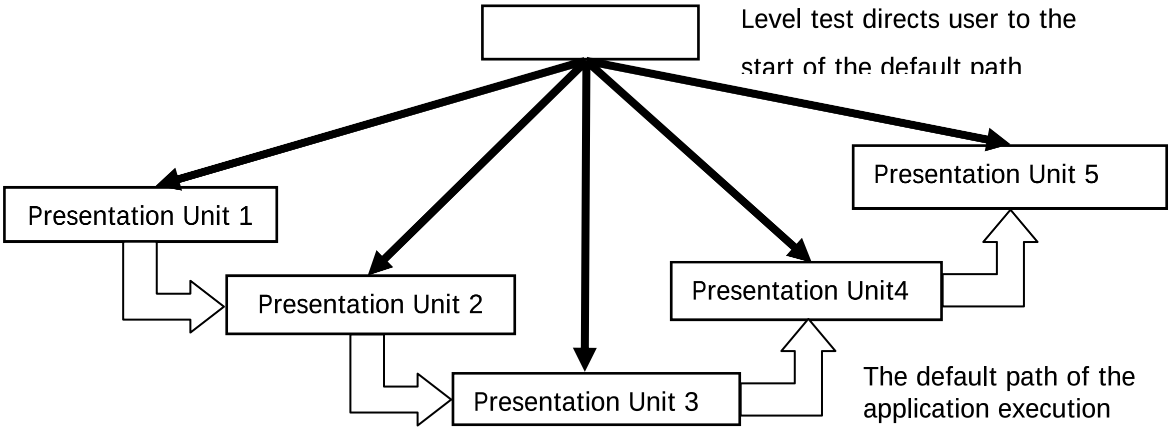


Fig. 2. The application structure selection

The example units of the application were presented in Fig. 2. The MAMS engine starting set-up values are initialised by the pre-test results or by the default setting file. Then, the courseware content will follow in accordance with the selected application flow - path. Based on the user interactions the application route (path) can be continuously modified. The application units' interrelations are presented in Fig. 3.

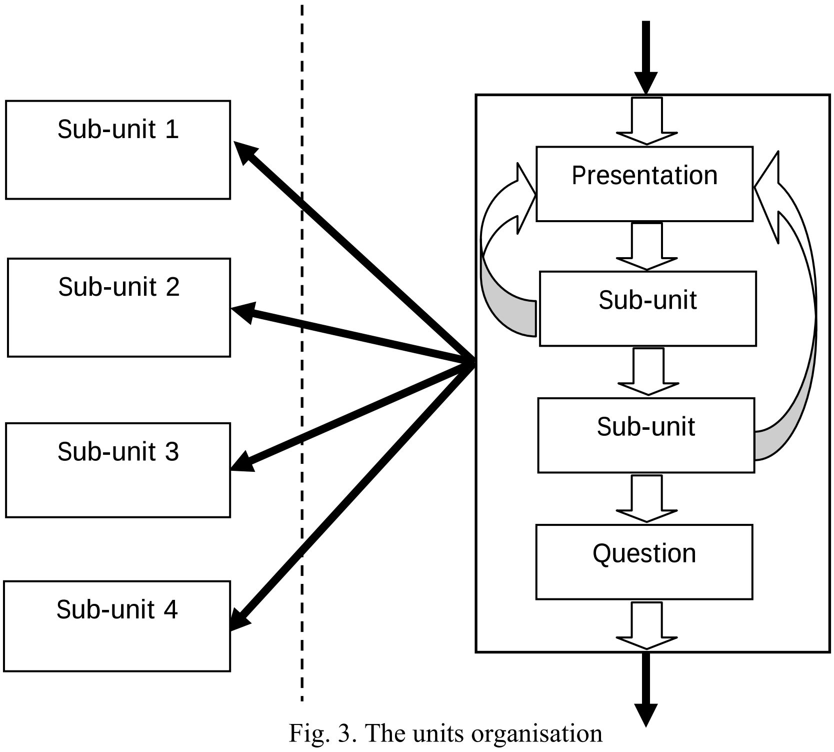


Fig. 3. The units organisation

The selected unit (or sub-unit) have the following characteristic features:

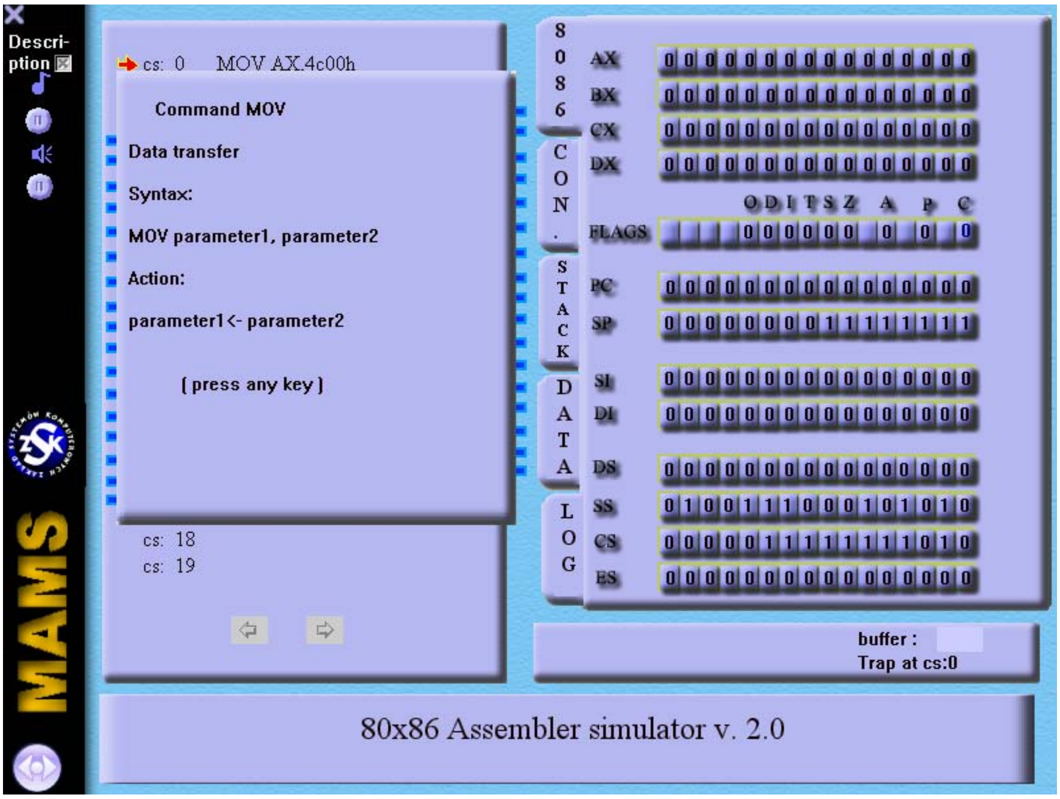
- concerns one specific topic / item,
- contains the knowledge validation mechanisms,
- the users feedback is classified in parametric measures describing the progress of the user's knowledge,
- evaluation cycles are limited to the specific knowledge threshold.

4. ASSEMBLY LANGUAGE (80X86) PROGRAMMING PRINCIPLES

A next courseware unit concerns the assembly language programming techniques for 80X86 processors Intel family. The recommended starting point of the application concerns illustrations of mechanisms explaining the processor codes execution. In this application the three level structure of the application can also be distinguished (that fulfils the Kay's strategy ideas [5], [6], [7]. The fundamental information layer provides the user with programming interfaces. The programming steps are under control of the debugging units, checking general programming principle. After the debugging process the programme codes are entered into the processor registers.

The register level programming results are illustrated by the assembly language facilities of the Intel 80X86 processor unit. The programming debugger simulates all main processing utilities. The simulation unit interface has been presented in Fig.4.

This very uncomfortable programming technique is presented and commented in a very friendly and simple way. Every example units will run any time thy user need them. The presented tasks are divided into several parts that can are recalled for solving the appropriate problem, listed in a form a code-boo. They can be automatically copied into the editing user board (Fig.5) then analysed line by line.



```
cs: 0    MOV AX,4c00h

Command MOV
Data transfer
Syntax:
MOV parameter1, parameter2
Action:
parameter1 <- parameter2

[ press any key ]

cs: 18
cs: 19
```

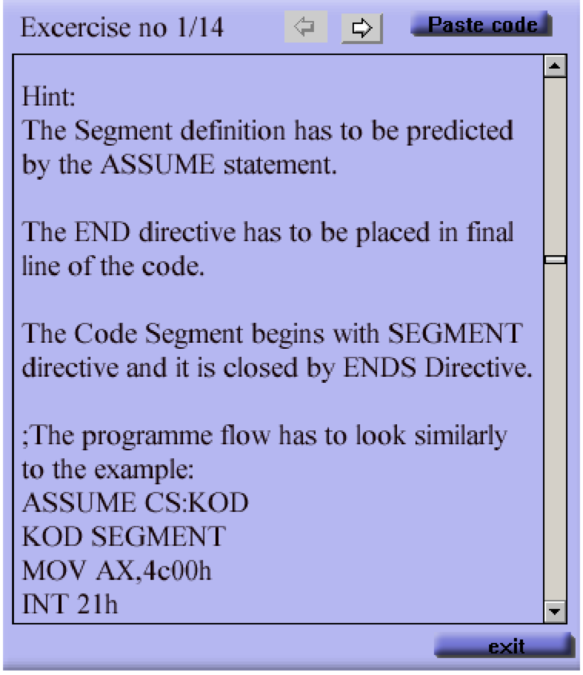
```
AX     0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
BX     0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
CX     0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
DX     0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
                O D I T S Z   A   P   C
FLAGS           0 0 0 0 0 0   0   0   0
PC     0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
SP     0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1
SI     0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
DI     0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
DS     0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
SS     0 1 0 0 1 1 1 0 0 0 1 0 1 0 1 0
CS     0 0 0 0 0 1 1 1 1 1 1 1 1 0 1 0
ES     0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

buffer :
Trap at cs:0
```

80x86 Assembler simulator v. 2.0

Fig. 4. The programming simulation interface

There are also available in different checking modes, as ‘execute all sequence’ with ‘set trap’ extensions. During this machine code execution the user is able to modify the processor states demanding the descriptions displaying short info about current actions. All the programme execution results are not only visible in the processors registers. The observation panel of the memory occupation and the console status can be explored as well.



Excercise no 1/14

Hint:
The Segment definition has to be predicted by the ASSUME statement.

The END directive has to be placed in final line of the code.

The Code Segment begins with SEGMENT directive and it is closed by ENDS Directive.

;The programme flow has to look similarly to the example:

```
ASSUME CS:KOD
KOD SEGMENT
MOV AX,4c00h
INT 21h
```

Fig. 5 The programming user – board

The application user writes the programme codes then the application will point out every mistake. The knowledge analysis concerns the discussed simulation processes. The programming results are evaluated by twenty separate parameters. This way the application user is provided by additional instructions in places where he did the mistakes (in relevant help formats). In case the questions level is too complex the application format will be automatically rearranged.

The courseware illustrates following programming problems:

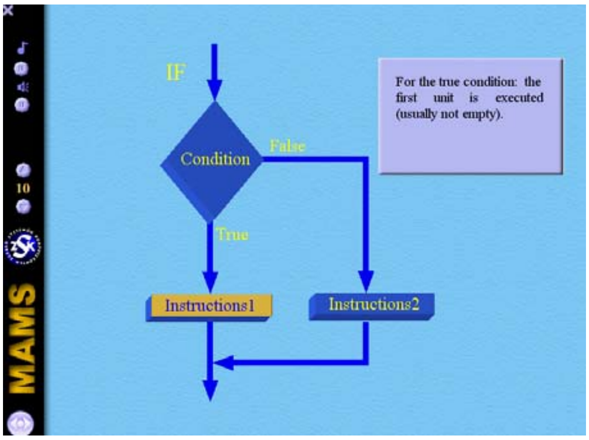
- programming facilities within a processor structure,
- programme segments memory space occupation,
- the assembly language instructions characteristics,
- the loops techniques within the assembly language,
- main loop-solutions.

In the application units development the following techniques were used:

- processing simulation technology,
- simple linear presentation,
- branched presentation formats,
- electronic book format,
- tutorial packages,
- several supplementary technologies.

The linear presentation was implemented for “in loop” idea explanation. This unit creates knowledge of basic statements, such as: IF, CASE, FOR, WHILE and REPEAT.

This course level is also recommended for not experienced students / pupils in the programming studies. The presentation schedule follows according to the block diagram presented in Fig. 6.



For the true condition: the first unit is executed (usually not empty).

Fig. 6. The IF instruction example-structure

To achieve the application success in creation of user friendly interfaces the specific needs have were considered, as: motivation factors and knowledge quality measures.

The electronic courseware unit was supported by the user and the course supervisor guide books.

5. THE PROCESSOR INTERRUPTS

A first of the discussed units illustrates the interrupt mechanisms available in 8 - and 16 - bit processors (of Intel). The courseware starts the presentation by simulation of interaction between an interrupt controller 8259 and a 16-bit microprocessor of Intel.

After that the questions sequence is provided with three levels of analysis. The user interactions are recorded within the "fill – up textual formats", where the user enters the missing expressions. The correct answer produces the highest score (100% result) with the next question given the user. A wrong answer of the user produces additional questions with better matched with expected formulas. Ex. the questions concern the same problem but the user asked for selection of a proper expression from combo-box.

The positive answer score is reduced into lower level, as 60% load into the total knowledge measures. In case a subsequent answers were not satisfactory the user is directed to the third level of application that produces once more the most needed presentation units; with comments to the wrong answers.

The courseware screenplay for quizzes contains eight versions with different interactions formats and questions set. A random function calls one of the questions scenario at a starting point.

The application random generator provides the user with following interfaces of the questions:

- fill the text gaps,
- mach the textual expressions,
- multiple choice of the question,
- single choice of the question.

In the discussed courseware two question formulas were applied:

- a not standard knowledge check – up,
- different authoring solutions for knowledge analysis.

6. THE APPLICATION PROTECTION UNITS

To protect the application from a not eligible user approach a four level security mechanisms were provided. The first protection level concerns all personal data recorded in the management unit of a main database sever of Distance Learning Centre (DLC), where only an authorized user will be able to download a main application-management platform from the DLC website. After the platform is installed the acknowledge code is sent to the main server via TCP/IP protocols, then the user has to log-in to the system by his password authentication coded. The user is provide by his personal rights to access particular resources that are checked any time the user runs the application. In case the system cracking is noticed the application progress can not be evaluated. The final security level concerns the Server File Management System. The course access is limited by all net and IP restrictions. The connection with the server is not possible in case the IP address is not recognised.

7. CONCLUSIONS

The e-learning content development principles depend on the subject they concern. Anyhow for the application discussed in this contribution several works of the application components standardisation were undertaken.

The remarks given above, concern one laboratory experiment for a subject “Computer Architecture”, at the University of Silesia – Computer Systems Department. The e-content of several subjects is under development since couple of years, as plenty of our experiments concern methods of the applications principles development:

- media-format definitions,
- development standards definition,
- tools library production and modularity definition,
- interactive measures evaluation and quality measures for units / frames repetition,
- the content costs a production time reduction, ,
- the Macromedia® environment extensions,
- the application databases security.

The application structure uses the independent client server controlling structure, instead of the solutions based on the internet browsers technology. This way we obtained professionally protected applications although not compatible with several world standards. Anyhow this incompatibility can be omitted in easy way, in case it was needed.

BIBLIOGRAPHY

[1] PIECHA J. The programmable shell for multimedia applications development. *Journal of Applied Computer Science*, Vol.7, No 2, pp.31-43, ISSN 1507-0360, Łódź 1999

[2] PIECHA J. KRÓL R. The MAMS an interactive applications management engine,. *Proc. of Int. Distance Learning Workshop' 04*, June 2004, Katowice, Poland ISBN 83-909518-6-X pp. 25-37

[3] PIECHA J., KRÓL R. PAWEŁCZYK P. A network node management shell for macromedia applications. *Proc. of Int. Conf. KOSYR*, ISBN 83-911675-2-6, pp. 493-499. Miłków 28-31 maj, 2001

[4] OLSZEWSKI A. An advanced validation formulas for interactive control of e-learning applications. *Proc. of Int. Distance Learning Workshop' 04*. June 2004, Katowice, Poland

[5] EBERTS R.E. Learning strategies in CAI design. The International Journal of Applied Engineering Education, No. 2/86, Pergammon Press, Oxford 1986.

[6] PIECHA J. The multilevel model for microcomputer ICAI systems. The International Journal of Applied Engineering Education, Vol.5, No 3/89, Pergammon Press, Oxford 1989.

[7] PIECHA J. Principles of e-learning applications characteristics and evaluation procedures Distance Learning Workshop DLW'04 Sosnowiec 2004

[8] KRÓL R. The access method to internet databases developed within the macromedia environment. Journal of Medical Informatics and Technologies. Vol.5, ISBN 83-909518-2-7, pp. IT 91-98. Szczyrk 2003

[9] PIECHA J. The Intranet Databases and some Approach Troubles into Multimedia Files. Proc. Int. Conf. *Computer Based Learning In Science - CBLIS'99*, Enschede 1999, the Netherlands, G7.

Works done for University of Silesia Distance Learning System (DLS) development.