*multimedia databases, internet databases,*
*web databases, distance learning*

Jan PIECHA[*,**], Robert KRÓL[*]

# THE SEQUENTIAL DATABASE ATTRIBUTES FOR INTERACTIVE APPLICATIONS MANAGEMENT

Many works has recently been done within presentation databases attributes development. The multimedia applications use various graphical interfaces and rich formats of graphics, animation and various audio effects. Although many efforts have already been done on this field new challenge one can observe that provide the Internet application [1], [2]. The paper shows the database organisation for multimedia applications available on web servers.

## 1. INTRODUCTION

Many applications that consists of various data formats [3], [4] as graphics and sound effects use interfaces allow the user to change the application attributes (Fig.1), among them:

- colours for the application background and for the navigation board selection,
- audio effects (active or not active),
- animation speed,
- the user interaction and navigation facilities.

Complexity of the database contents implies a structure of an interface that makes possible to go through the database via sequence of questions. This way the database part can be selected. Fig. 2 shows the example of the application interface that allows navigating the user within the application.

The user defines:

- the application repetition mode activity, with selected number of repetitions,
- a score level for the user interactions that control frame's mode for repetition,
- a repetition limit of the application,
- score levels (thresholds for positive and negative answers) defining switches for current levels of the lesson.

The central databases have to be protected from not illegible users. Anyhow the illegible user will permanently sent to the main database its application status that describes the user activity.

[*] University of Silesia, Institute of Informatics, Dept. of Computer Systems, ul. Będzińska 39, Sosnowiec, Poland
[**] Technical Silesian University, Dept of Transport Informatics, ul. Krasińskiego 13, Katowice, Poland

The approach to the network database is provided by streaming or workstation technology. When the application runs in workstation technology the user status file has to be sent to the server any time the application is closed. This way the supervisor of the database is able to spot the user activity during his interaction processes.

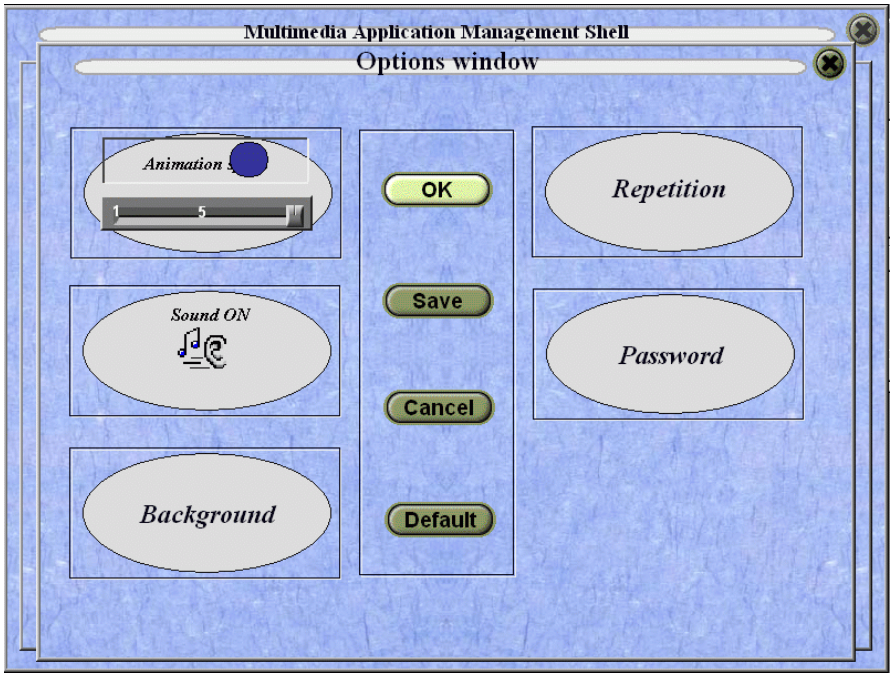


Fig.1. The application attributes settings

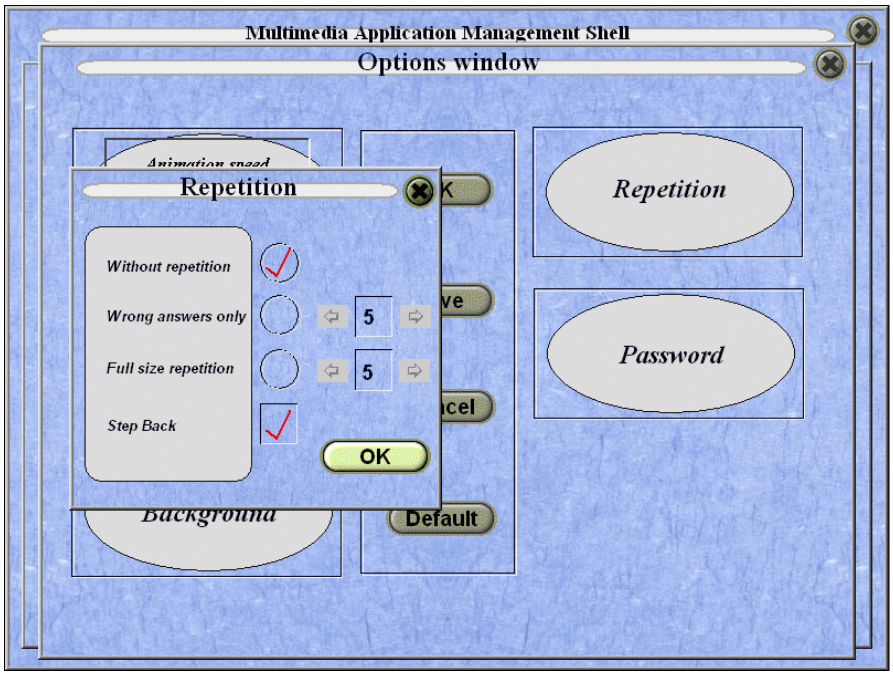


Fig.2. The application repetition attributes

The way of the application exit has to be noticed as well. Either the user quits the application (by pressing the button “Quit” - within the application) or by the command “Exit the task” of an operating system. When the user quits the application via the operating system the application status can not be sent into the server as the link between network station is not active. The user obtains an application message that his entrance to the database has been rejected. Anyhow, the application status transfer into the main database will predict a next sample of running the application.

The user status file contains *date* and *hour* of the user actions, as:

- login and logout with an exit to the system,
- the application user database modification,
- a current password modifications,
- time markers of entering and leaving the presentation and question frames,
- starting, breaking and quitting the course units.

The example status file has been mentioned bellow:

*ID=Student1*

| *Date* | *Time* | *Action* |
|---|---|---|
| *7 May 2002* | *16:34:01* | *Login* |
| *7 May 2002* | *16:40:04* | *Modification AppSettings, Write to DataBase* |
| *7 May 2002* | *16:42:24* | *Begin Course: CourseName1, Mode:3* |
| *7 May 2002* | *16:42:45* | *Jump To Presentation Frame: Name1* |
| *7 May 2002* | *16:47:22* | *Return From Presentation Frame: Name1* |
| *7 May 2002* | *17:52:00* | *Jump To Question Frame: Q_Name1* |
| *7 May 2002* | *17:54:10* | *Return From Question Frame: Q_Name1* |
| *7 May 2002* | *19:33:32* | *End Course: CourseName1* |
| *7 May 2002* | *19:34:11* | *Logout and Exit To System* |

Although all of the application attributes are not used in the same time they are available in the application.

## 2. THE DATABASE ORGANISATION EXAMPLE

The application user can interact with the application units (components) in the database. The data flow within the MAMS application has been presented in Fig. 3.

The SQL explorers allow selecting the units and to transmit them into the local database (to workstation). The application “client” contains data pointers localising the data on the workstation. When this data is not available on the workstation the searching operations move into the server, then the found data unit is transferred to the workstation via FTP services.

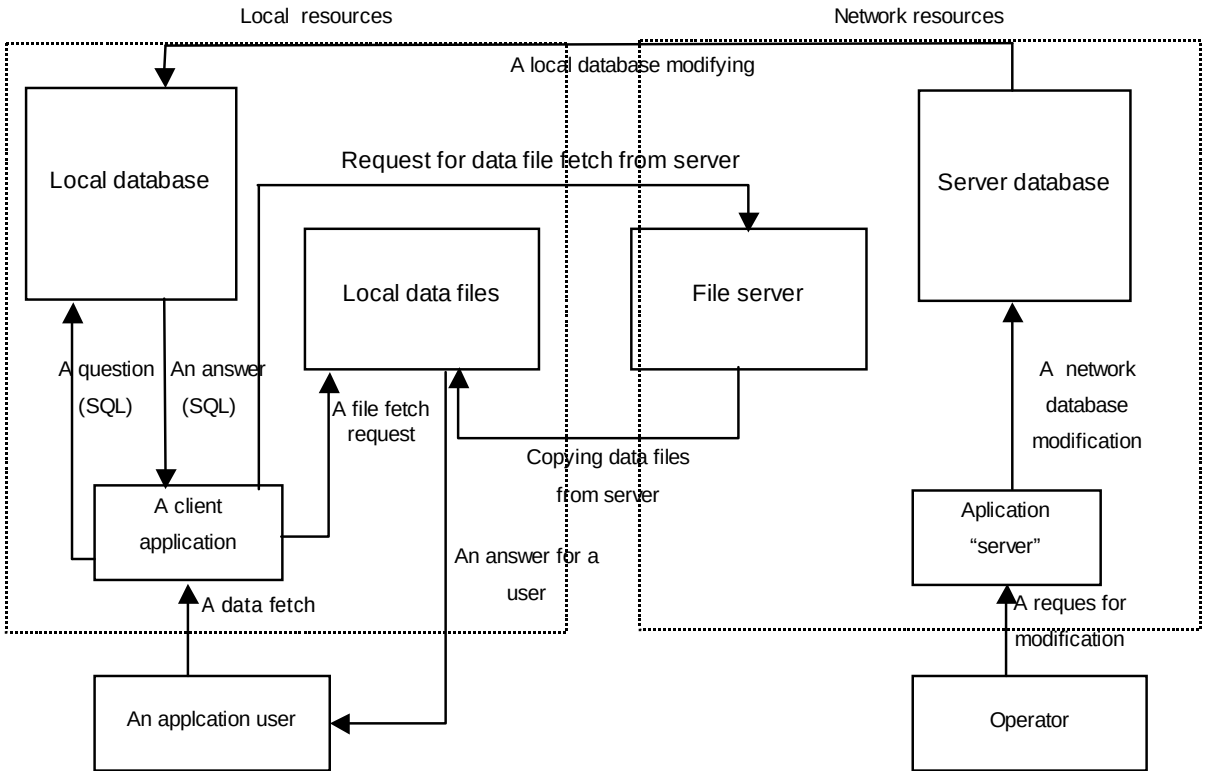


Fig.3. A data flow in application (MAMS)

## 3. THE APPLICATION DATABASE STRUCTURE

The application database control unit allows establishing three categories of presentation files: courses (containing lesson sequences), separate lessons and single frames (Fig.4).

The table *Question* contains frame numbers where at least one question format is noticed. Each frame is recorded as a separate file in table *Frames*, with its private identifier *FileName*. The frame identifier calls a path into the file. Each frame can also be recalled by the application platform via *FrameID's* or *QuestionID's* from tables *Files*, *Frames* or *Questions*. They allow associating the selected frames with an application. Good example of these associations could be a unified background or sound effects (or both) for several presentation frames.

Additional part of the database concerns the application user (Fig.5) with his own data files: password, user-status, application-status, application settings and history.

The personal data of the user (name, login, study level, home address) are kept in the table *UserData*. The table *ApplicationSettings* controls an animation speed, a sound switch (on/off) and an application colour background attributes.

When the application run is stopped its current status is saved in table *UserStatus* - together with all settings defined earlier.

The table *History* collects the application progress. Tables *ApplicationSetting* and *UserData* are available in local workstation resources. Tables *History* and *UserPassword* are available on server only. Table *UserStatus* is available both on local workstations and on server.

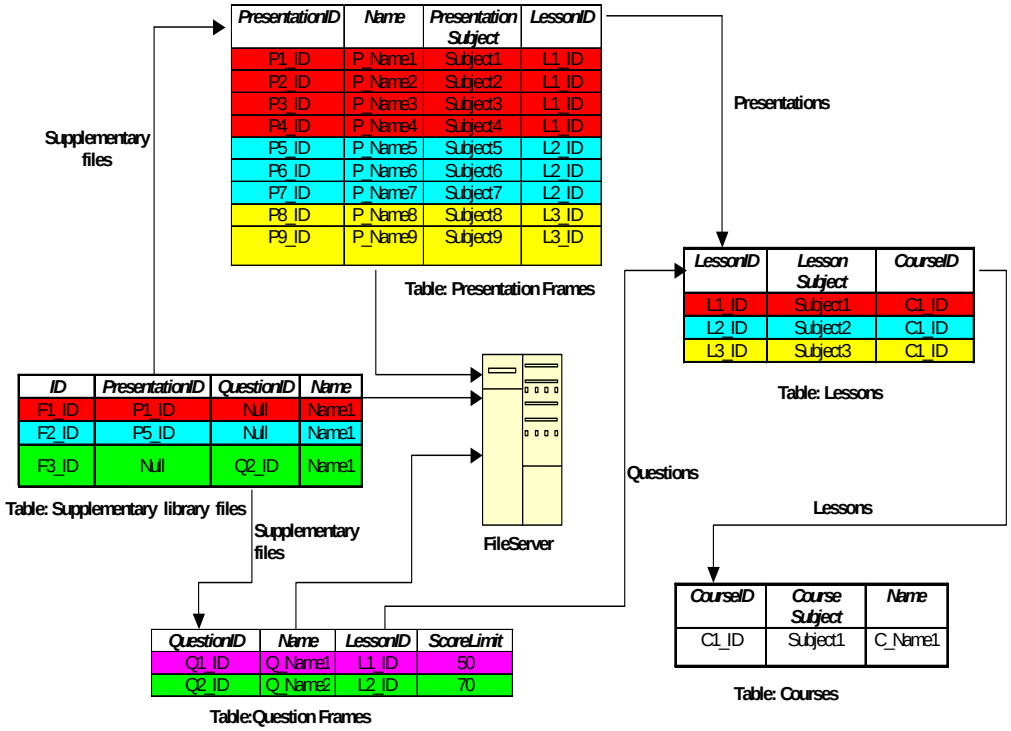


Fig.4. The application database structure

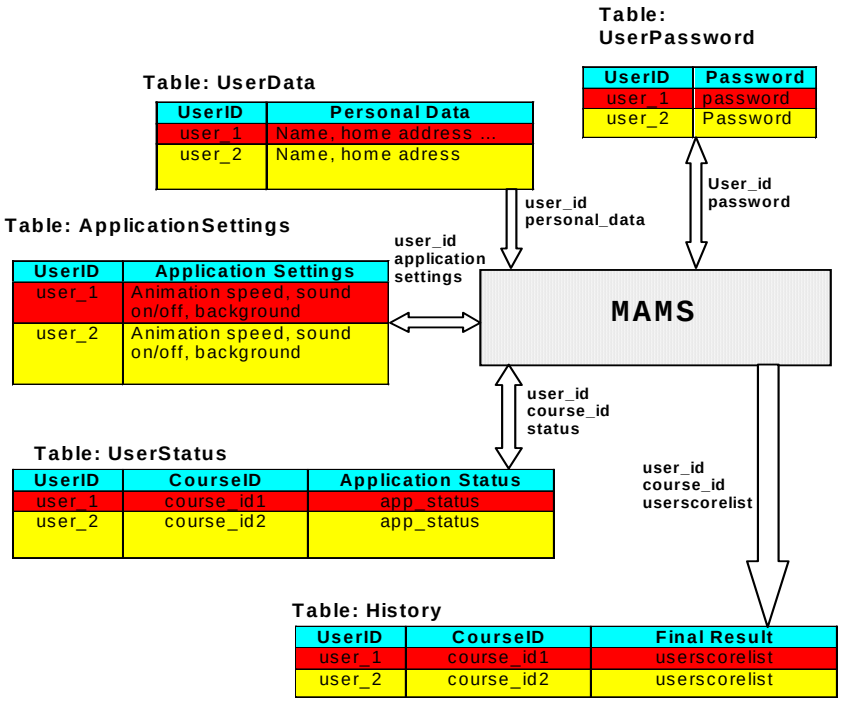


Fig.5. The user database for: personal-data, password, user-status, application-status, application settings, and history

## 4. CONCLUSIONS

The database complexity and the application files length implies the user interface characteristics. They make possible to go through the database via sequence of questions, provide the user with several unification principles and simplifications of the application structure. The database structure described in this paper is the example solution of a fast explorer that allows to navigate within the application contents.

The database platform provides the application designer with several unification modes for a whole set of database units.

BIBLIOGRAPHY

[1] KRÓL R. The access method to internet databases developed within the macromedia environment. Journal of Medical Informatics and Technologies. Vol.5, Nov. 2000, ISBN 83-909518-2-7, pp. IT 91-98.
[2] PIECHA J., R. KRÓL, P. PAWEŁCZYK A network node management shell for macromedia applications. Proc. of Conf. KOSYR 2001, ISBN 83-911675-2-6, pp. 493-499.
[3] 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.
[4] PIECHA J. The Intranet Databases and some Approach Troubles into Multimedia Files. Proc. Int. Conf. *Computer Based Learning In Science - CBLIS'99*, Enschede, the Netherlands, 1999, G7.
[5] EBERTS R.E. Learning strategies in CAI design. The International Journal of Applied Engineering Education, No. 2/86, Oxford 1986.
[6] YOUNESS S., SQL Unleashed, SAMS Publishing, ISBN 0672317095, November 1999