Novi Sad J. Math.
Vol. 37, No. 1, 2007, 155-162

A DATABASE MODEL FOR LIBRARY MATERIAL USAGE[1]

Danijela Tešendić[2]

Abstract. A data model for library material usage in city libraries is proposed. The data model was designed with the aim of supporting all evidenced information requirements for the library material usage. The proposed data model is used in the development of the fourth version of the BISIS library software system.

AMS Mathematics Subject Classification (2000): 68U35

Key words and phrases: Library material usage, database model, UML, BISIS library information system

1. Introduction

The BISIS library software system has been in the process of development since 1993 and an overview of its development is given in [1]. The current version of the system is 3.1. The segment related to the library material usage, i.e., monitoring of library collection utilization, represents a separate part of library activities and, within the BISIS library information system, is fulfilled by the *Circulation* application [2]. This application satisfies all the requirements needed for the work of academic libraries. The application is implemented in Java environment. All user data existing in the system are stored in the tables of a relational database.

After the installation of the BISIS library information system Version 3.1 into city libraries some shortcomings were noticed in the existing implementation of the *Circulation* application in relation to the requirements the city libraries have in their work. Based on the requirement analysis and needs of the city libraries that use the BISIS system, a new model of the library material usage subsystem was developed. Based on this model, an application of the subsystem was implemented within the development of the fourth version of the BISIS library software system.

In this paper, the static structure of the system defined through class diagrams and textual description of the subsystem classes for library material usage is presented. The Unified modeling language (UML 2.0) and the CASE tool – the *Visual Paradigm for UML* 5.3 – are used for modeling of the subsystem [3].

[1] This paper is a part of the research project "Abstract models and applications in computer science", supported by the Ministry of Science and Environmental Protection of the Republic of Serbia (Project No. 144017)

[2]Faculty of Sciences, Department of Mathematics and Informatics, Novi Sad, Serbia

2. Information requirements of the subsystem for library material usage

After the installation of the BISIS system into city libraries, the existing Version 3.1 was not capable of fulfilling all the requirements needed for the work of city libraries. This was particularly true for the subsystem for the library material usage. Information requirements of the usage processes include:

1. Library user data
2. Membership data
3. Library material usage data
4. Reminders sent to the library users
5. Statistics reports
6. User interface performances
7. User data archiving

Library user data. A certain set of user's data is necessary at the time when a user is registered. This data set is defined by the library. Some of the set data should be mandatory at the time of the user's registration and this mandatory character is also defined by the library. The set of mandatory data differs among libraries, and it is thus necessary to ensure adjustments to the mandatory character of data for each library separately.

Membership data. It is necessary to register data of the user membership renewals as well as to provide notes of membership expiration. There are different types of memberships and user categories, both defined by the library. A type of membership determines the duration of membership, whereas a user category regulates the greatest number of publications a user can check out, as well as a maximum borrowing period for checked out publications. A membership fee for a user depends on category and type of membership and is defined by the library. For each registration or membership renewal, it is also necessary to keep track of the librarian and the library department that carried out them.

Library material usage data. These data are related to the publications checked out and returned by users. In contact with the user, the librarian needs to have the following on display: only books currently checked out, reminders of possible overdue books as well as the user's loan history. For every checked out, returned or renewed book it is necessary to keep track of the librarian and the library department that carried out these operations.

Reminders sent to library users. It is necessary to enable periodical printing of reminders to be sent to users with overdue publications. There are several types of reminders and the content displayed on them differs from library to library. Thus, it is necessary to enable changes in the content of reminders. In contact with the user, the librarian needs to have on display the information

on the overdue reminders currently sent to the user. Printing of the annual register of the overdue reminders the library holds in its archive, should be likewise enabled.

Statistics reports. A possibility of generating a certain set of statistics reports about users and library holdings usage data is expected. The reports include the following: registered user book; composition of registered users based on their membership type, category and sex; loan history for each user; loan history for each publication; books most read; most active readers; borrowed and returned publications by UDC groups; borrowed and returned publications by UDC groups and user categories, etc. A possibility of searching the user database by all possible criteria is likewise expected. This would enable obtaining diverse statistical data, including library-specific data not mentioned in the set of predefined reports.

User interface performance. In order to achieve more efficient work with users special attention is to be devoted to diminish a number of operations that a librarian should perform when processing a user's request, as well as to a faster response of the software system.

User data archiving. Also, in order to achieve more efficient work with users, it is necessary to be able to remove the data of users who did not renew their membership, i.e., to archive them. One example of data archiving using XML documents is described in [4].

3. Data model

Figure 1 shows a proposal of the data model using class diagrams and supporting all the information requirements specified in the previous section.

The *Users* class contains data about individual library users and its attributes are: *sys_id* (unique system identification number of a user), *user_id* (membership card number of a user), *first_name* (user name), *last_name* (family name), *parent_name* (name of a parent), *address* (address), *city* (place of residence), *zip* (zip code), *phone* (phone), *email* (e-mail address), *jmbg* (personal number of a citizen), *doc_id* (document type: identification card, passport, etc.), *doc_no* (document number), *doc_city* (place of document issue), *country* (country of document issue), *gender* (sex), *age* (age), *sec_address* (additional address), *sec_city* (additional address place of residence), *sec_zip* (additional address zip code), *sec_phone* (additional phone), *note* (note), *interests* (interests), *warning* (indicator of reminder), *occupation* (occupation), *title* (title), *index_no* (student number), *class* (year of study). Collective membership to which a user belongs, user category, membership type, user's academic degree, mother tongue and affiliation are identified on the basis of the relations with the *Groups, User_categs, Mmbr_types, Edu_lvl, Language* and *Organization* classes.

The *Groups* class contains data about corporate library users. The class has the following attributes: *sys_id* (unique system identification number of a user), *user_id* (membership number of a user), *inst_name* (institution name), *sign_date* (registration date), *address* (address), *city* (place of residence), *zip* (zip code), *phone* (phone), *email* (e-mail address), *fax* (fax), (phone), *sec_address*

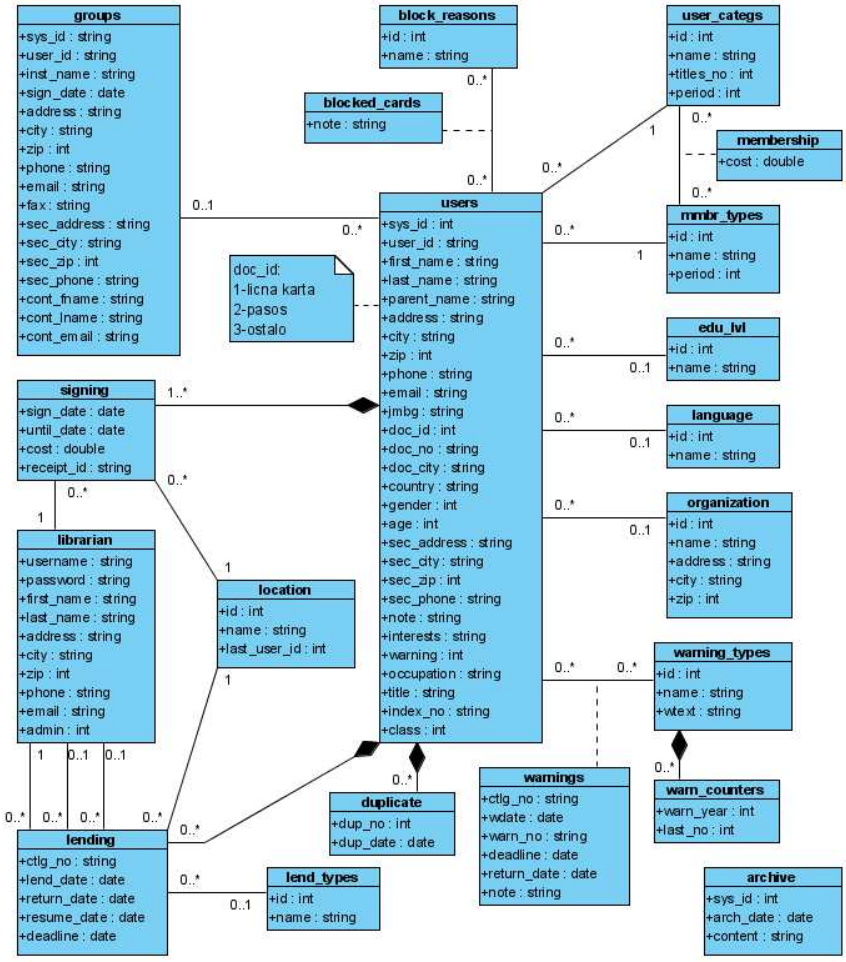

Figure 1: Data model

(additional address), *sec_city* (additional address place of residence), *sec_zip* (additional address zip code), *sec_phone* (additional phone), *cont_name* (contact person name), *cont_lname* (contact person family name), *cont_email* (contact person e-mail).

The *Signing* class contains data about the membership fee of an individual member. The class has the following attributes: *sign_date* (beginning date of the membership fee), *until_date* (expiring date of the membership fee), *cost* (amount), *receipt_id* (receipt number). The library user, librarian who renewed membership, and location of membership renewal are identified on the basis of the relations with the *Users, Librarian* and *Location* classes.

The *Lending* class contains data about user's checked out and returned items and has the following attributes: *ctlg_no* (accession number), *lend_date* (check out date), *return_date* (return date), *resume_date* (renewal date), *deadline* (due date). The library user, the librarian who charged out an item, the librarian who renewed an item, the librarian who discharged an item, and check out location, are identified based on the relations with the classes *Users, Librarian* and *Location.*

The *Librarian* class contains data about librarians having access to the system. The class has the following attributes: *username* (user name), *password* (password), *first_name* (first name), *last_name* (family name), *address* (home address), *city* (place of residence), zip (zip code), *phone* (telephone), *email* (e-mail address), *admin* (indicator of administrator privileges).

The *Location* class is a codebook of check out and return locations. The class has the following attributes: *id* (location code), *name* (location name), and *last_user_id* (counter for the automatic assigning of numbers to new users).

The *User_categs* class is a codebook of users categories. The class has the following attributes: *id* (category code), *name* (category name), *titles_no* (maximum number of publications checked out by a user of that category), and *period* (check out period).

The *Mmbr_types* class is a codebook of membership types. The class has the following attributes: *id* (type code), *name* (type name), and *period* (membership fee duration).

The *Membership* class is a codebook of membership fees and has the following attributes: *cost* (membership fee). The membership types and user categories are identified on the basis of the relations with the *User_categs* and *Mmbr_types* classes.

The *Edu_lvl* class is a codebook of academic degree and has the following attributes: *id* (academic degree code) and *name* (academic degree name).

The *Language* class is a codebook of languages. The class has the following attributes: *id* (language code) and *name* (language name).

The *Organization* class is a codebook of organizations and has the following attributes: *id* (organization code), *name* (organization name), *address* (address), *city* (place of residence), and *zip* (zip code).

The *Blocked_cards* class contains data about blocked user cards and reasons for their blocking. The class has the following attributes: *blocked_reason* (reasons for blocking) and *note* (note). The library user is identified on the basis of

the relation with the *Users* class.

The *Duplicate* class contains data about duplicate cards. The class has the following attributes: *dup_no* (duplicate number), and *dup_date* (date of duplicate issue). The library user is identified on the basis of the relation with the *Users* class.

The *Warning_types* class is a codebook of reminder types. The class has the following attributes: *id* (reminder code), *name* (reminder name), and *wtext* (reminder text).

The *Warn_counters* class contains data about counters of the reminder numbers for each year and class has the following attributes: *warn_year* (year) and *last_no* (the last number for the year in question). The reminder type is identified on the basis of the relation with the *Warning_types* class.

The *Warnings* class contains data about generated reminders. The class has the following attributes: *ctlg_no* (accession number), *wdate* (date of reminder generating), *warn_no* (reminder number), *deadline* (return deadline), *return_date* (return date), *note* (note). The library user and reminder type are identified on the basis of the relations with the *Users* and *Warning_types* classes.

The *Archive* class represents an archive of user data. The class has the following attributes: *sys_id* (user number), *arch_date* (date of data archiving) and *content* (content).

4. Implementation of the subsystem for library material usage

The subsystem for library material usage is implemented in the *Java* environment [5]. The *Eclipse v3.2* environment [6] is used as a development kit. The forms of graphical user interface are implemented using the *Swing* package as a desktop application, together with all the functionalities and advantages that the desktop applications have over *web* applications. The application part that works with the database is carried out by means of the *ORM* (*Object/Relational Mapping*) technology and the *Hibernate* package [7] is used for the implementation. The *MySQL Enterprise Server 5.0* [8] is used as database server.

The database schema is composed on the basis of the class diagram that is described in the paper. Each diagram class corresponds to one table of the database schema. All class properties are mapped to corresponding table columns. Based on the relations among the classes, the corresponding referential integrities among the tables are made.

The application is implemented as a classical client application, the execution of which completely takes place on the client. Such an implementation is chosen primarily because of the functionality and performance of the screen forms that are the basic requirements on the part of libraries. Beside the screen forms, the application also includes an implementation of the object database model by which the rest of the application has access to the database data. The part of the application that performs communication between that object model and the database is implemented by the *Hibernate* package. The implementation uses JDBC driver for communication. For further application development,

another such part is planned that would not use the JDBC driver. Instead, the objects would be sent to the server using the HTTP protocol and the server would communicate with the database. Such an implementation should enable work for clients without permanent connection with the database, or, when the database is on a remote server. (This was also one of the library requirements.)

The subsystem for library material usage requires the displaying of the data stored in the library records. These data are obtained by the text server implemented within the BISIS system. The text server enables searching and downloading records from the database of the BISIS system. Database searching is done in accordance with accepted prefixes.

The implementation of the application for library material usage is carried out in such a way that all the library requirements specified in Section 2 are fulfilled. Testing and verifying this application was performed on the data obtained from a city library. The scope of the data was 7.315 active library users and 95.209 library records. Verification results of the application showed that all specified information requirements of the libraries were fulfilled.

5. Conclusion

The BISIS Library Information System Version 3.1 consists of the two basic segments. One is related to the processing of library material in the UNIMARC format whereas the other is related to the process of library material usage. During the installation of the system into city libraries the segment related to the library material usage it was found that it could not meet all the needs concerning the work in these libraries. This was true for the volume of input data, data display ability, statistics reports, decrease in number of operations that librarian should perform in order to process a user request, as well as response speed of the software system.

Based on the additional information requirements, a data model for library material usage is proposed and the application based on this data model was implemented. Testing and verifying this application were performed on the data obtained from a city library and all specified library information requirements are stated as fulfilled.

A further development of the application for library material usage would consist of enabling clients to work with a remote database via Internet. This was also one of the library requirements. This would enable the work of the branch libraries belonging to the city libraries that have a centralized library user database.

References

[1] Surla, D., Konjović, Z., Overwiev of the Development of the Library Information System BISIS. Proc. of International Conference on Distributed Library Information Systems, Ohrid, FYROM (2004) 13-18.

[2] Tošić, T., Surla, D., Usage of the Bibliographical Material. Distributed Library Information System BISIS (Chapter 10), Group for Information Technologies, Novi Sad, 2004. (in Serbian)

[3] Visual Paradigm SDE for Eclipse, v.4.0, http://www.visual-paradigm.com/product/sde/ec/ (8. November 2006)

[4] Tešendić, D., Surla, D., XML Schema for members' data. Proceedings of the Conference PRIM 2006, Kragujevac. (in print)

[5] Java homepage, http://java.sun.com/ (23. July 2006)

[6] Eclipse SDK v3.2.1, Eclipse homepage, http://www.eclipse.org/ (15. October 2006)

[7] Hibernate v3.2.2, Hibernate homepage, http://www.hibernate.org/ (12. December 2006)

[8] MySQL homepage, http://www.mysql.com/ (25. September 2006)

Received by the editors April 11, 2007