Преглед НЦД 9 (2006), 53–58

Mile Jovanov,
(Institute of Informatics,
Faculty of Natural Sciences and Mathematics, Macedonia)
Marjan Kindalov, Nikolche Mihajlovski,
Goran Kolevski, Pece Miloshev
(students, FNSM Skopje, Macedonia)

SYSTEM FOR STORING DATA ABOUT NATIONAL HERITAGE WITH ADVANCED SEARCH TECHNIQUES

Keywords: National heritage, digitization, search, database, WEB, server, client

I. Motivation

The culture is a phenomenon that characterizes every person. All of the nations through their existence build a culture they maintain and develop. It makes people feel like a part of a community. But the existence of different cultures divides people into separate groups. To overcome the prejudices, members of one cultural community have to learn characteristics of other cultures. Following the tendency of digital communication as a way of successful information sharing, a need for cultural heritage digitization occurs.

But digitization of the cultural heritage solves only a part of the problems. We have to find an effective way of data organization to achieve simple but reliable information access. We need a system with the following characteristics:

- Fast data access,
- Data searching by any of the attributes,
- Simple, but controlled information update,
- Simple information structure upgrade,
- Multi-user data management support.

As we can see on Fig.1 there are different subjects that digitize the cultural heritage. For example, those subjects can be museums, churches, archeologists, collectors, etc. Their materials, artifacts, digitized in some form, are placed on Internet. Global information about the exhibits can be found by any Internet search engine, but some specific information, e.g. churches built in Macedonia, between 1400 and 1540 AD, which occupies area of 200-500 square meters are hard to find.

II. Basic concepts of the solution

The idea of the NHISS (National Heritage Information Storage System) is to provide the users with answers to the questions of this type. It stores the attributes of the artifacts, so the user can easily search by any of them. Users are categorized into:

Digitizers are the users that import data into NHISS. Information about digitized objects they announced on Internet and key attributes are entered into NHISS.

Administrators are responsible for managing NHISS. They add new classes on digitizers' demands and verify the data.

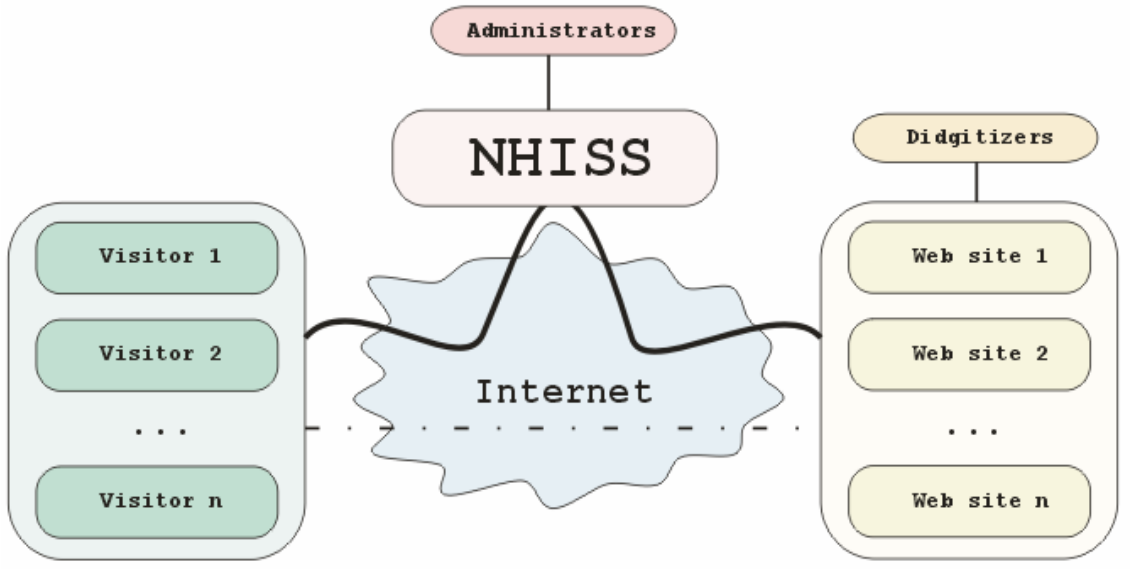

Figure 1: Connection establishment

Visitors are the target group of NHISS. They can access it via Internet and search for digitized material. Then NHISS provides them with references to the exhibits they are looking for.

The system consists of four main components:

Database: The main part of the database stores the data entered by digitizers. The second part stores the information needed for database management. The data is structured in classes. Each class has its own attributes which can be inherited by its subclasses. Administrators can add new classes without changing the logic structure of the database.

Server application: This application is interface between the client applications and the database. It receives requests from them and translates those requests into appropriate SQL queries. Then the server translates the SQL answers into messages for the client applications.

Client application: Several instances of this application are connected to the server via Internet or LAN. Administrators can manage the database using the application. By managing we consider:

- Adding, changing and removing classes of artifacts (objects),
- Adding, changing and removing administrator accounts1,
- Verifying the data entered by digitizers.

Web application: This application is the interface for the visitors and digitizers. Visitors can search for announced artifacts while digitizers can add new information.

III Database solution

First question that arises is, can we put whole digitized material in a central database. The disadvantages of this solution are:

- Administrators will be also responsible for the visual presentation of digitized material
- There should be the part of the system that allows making presentations of digitized material, not necessarily for simple use
- The possibility for variety of presentations will be minimal
- Expensive servers, high speed internet connection

[1] Only some administrators can manage system data (e.g. administrator accounts), the rest of them can only manage the digitizers data.

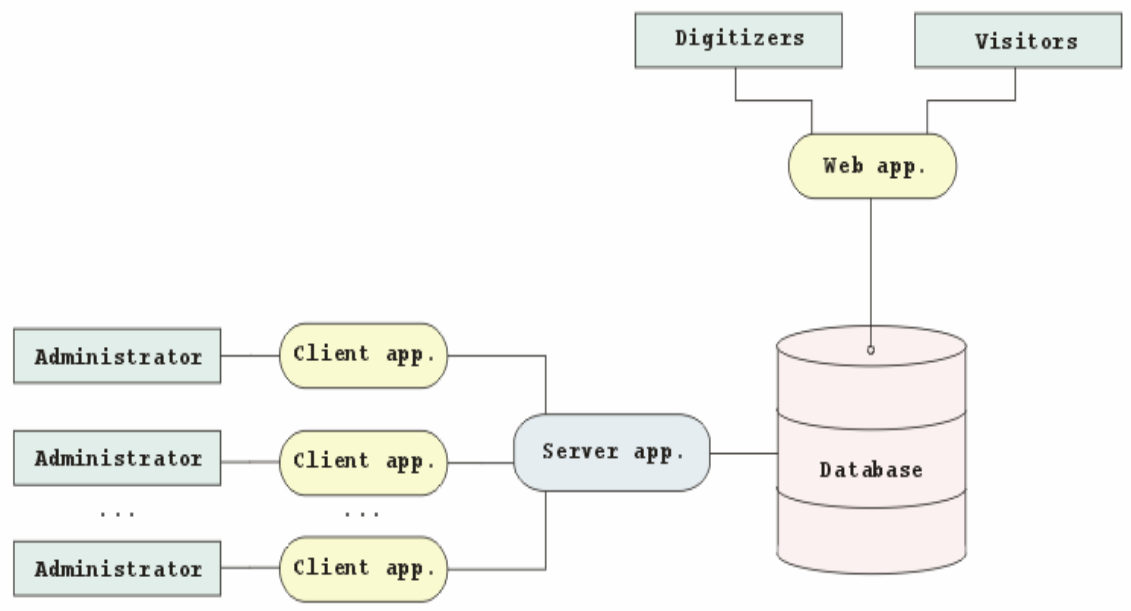

Figure 2. Structure of NHISS

This solution is based on assumption that the digitizers will be responsible for the presentation, and the database will contain only the main attributes for the artifact and a hyperlink which point to the site where it is presented.

Because the system should allow every artifact, object, etc to be announced, the main problem are different attributes for different classes of artifacts. It isn't possible to predict and to know all different classes in advance, so the solution should allow the set of artifacts that can be announced to be easily extensible.

First conclusion, during the research was that all attributes can be divided in three groups: numerical, text and generic. So, all numerical characteristics of an artifact could be announced as numerical attributes, the textual ones in text attributes and finally for each class of artifacts we define some generic attributes. Generic attribute is attribute that can have a value from a finite set of values for that attribute. In our case generic attribute can have only string value, but the string could represent path to pictures, figures or other files on the computer which make the scope of the generic attributes usage larger. Example for generic attribute is attribute color. Color can take only one of these values: white, yellow, red, green, blue, orange, gray and black. So wherever we have attribute with finite set of values we use generic attributes.

This concept for object description is an abstraction for already existing models like DUBLIN CORE. If the sets of attributes are well defined, then the concept can be described using some languages like XML.

Database, if implemented in this way, can be infinitely extended, i.e. large tree structure will be defined, step by step, and this tree structure will consist of all entities organized by category (classes) by their characteristics. The proper attributes can be added to classes, at any time, without significant changes in logical organization of the database and no changes of database structure. Logical structure of the database is presented in figure 3.

Database description. *Artifact* is the main attribute in the database schema. Every artifact must be a member of some *class*. The class is linked with itself using the relation *Sub_Class*. That allows defining logical structure Class – Sub_Class for structural organization of the artifacts.

`

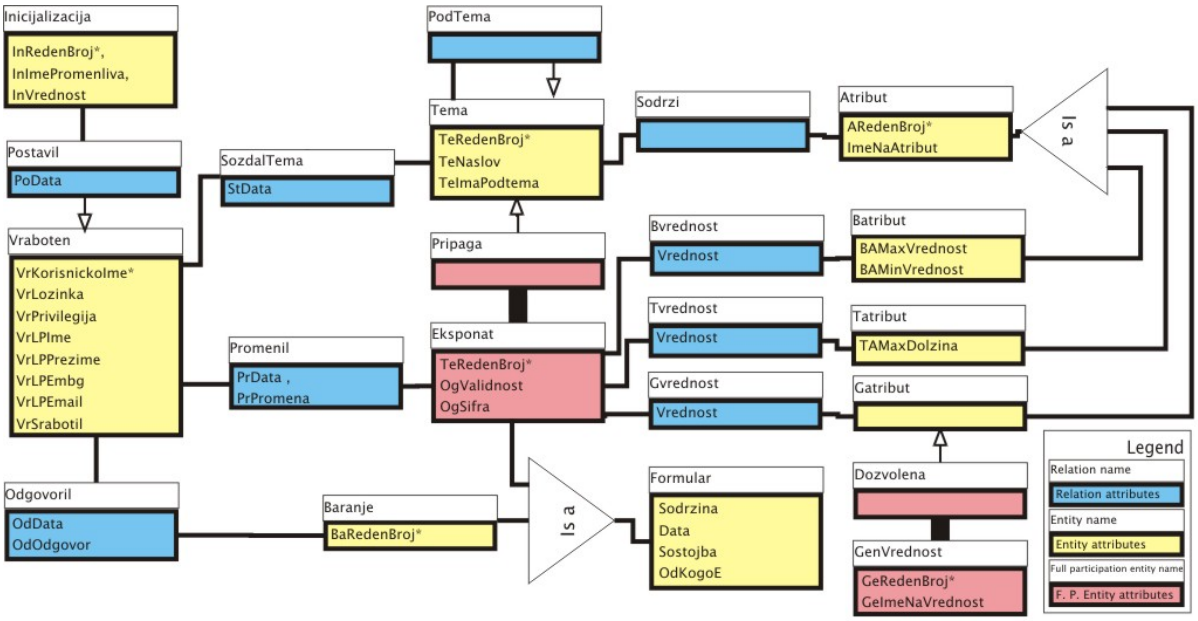

Figure 3. Logical structure of the database

All classes (and artifacts) are characterized with a number of numerical, texts and generic attributes which are elements of the entities. To define the attributes of some class, the relation *contain* is used. It consists of pairs, *class – attribute*. Since every *artifact*, as part of a *class*, must have value for every *attribute*, three more relations are defined, containing triples *artifact – attribute – value*.

Implementing generic values complicates the schema. The entity *GenValue* is added for defining a generic attribute and a list of proper values. For example, generic attribute "Material" is defined, and the list of proper values is: wood, stone, earth (soil), bronze, silver, gold.

Description of data search. With database defined as above, it is clear that SQL statements can't be static. They are generated dynamically with special syntax that allow acceleration of searching process, i.e. avoid unnecessary filtering. For the simplicity, search process will be described by the next example:

```
select * from oglas where teRedenBroj=0 and ogRedenBroj in
 (select ogRedenBroj from tVrednost where tvVrednost like '%crv%' and taRedenBroj=0 and ogRedenBroj in
  (select ogRedenBroj from bVrednost where bvVrednost >=1900 and baRedenBroj=1 and ogRedenBroj in
   (select ogRedenBroj from bVrednost where bvVrednost <=2100 and baRedenBroj=1 and ogRedenBroj in
    (select ogRedenBroj from gVrednost where gvVrednost=1 and gaRedenBroj=3 and ogRedenBroj in
     (select ogRedenBroj from tVrednost where tvVrednost like '%ena%' and teRedenBroj=0 and taRedenBroj=0))))) order by ogRedenBroj;
```

As shown, by the example SQL query, the filtering is done step by step, not on the whole set, but on already filtered data. The idea is to lower the complexity. So, there are n-1

steps of filtering, where n stands for the number of characteristics of the class. At the beginning, the whole set is considered, and than just artifacts with specific value for some attribute are filtered. Category is another criterion for filtering.

IV. Example

In this section we will give an example of how the whole system works. Suppose the museum of Bitola wants to announce the information about Heraklea (archeological occurrence in Bitola). They digitize the information and then they build a website www.heraklea.info.mk. It's up to them to organize their site in the way they think is most appropriate to their research. After building the website they can visit the web application of NHISS. With this web application they can enter data into the database, about the artifacts they want to announce. Let consider they want to announce a mask. Following steps best describes that process:

- Check if there is a class ARTIFACT and subclass MASK. The mask is artifact so its class should be a subclass of the ARTIFACT.
- If ARTIFACT is missing send a request to the administrator for creating this class.
- If MASK is missing send a request to the administrator for creating it. In this request the digitizers should specify the attributes of the class MASK e.g. material, place, shape etc.
- After all the classes are present the digitizers are filling a form, where they are entering the exact characteristics of the mask.
- As final they are entering the link to their website www.heraklea.info.mk

Next when an administrator comes to work she is login to the server using the client application. If there are any requests for new classes she can create them or reject them. After the mask is announced she can check if the link is correct and appropriate to the announcement.

Finally some visitor will visit the web application. He can search for artifacts, or specifically masks. The search can be done by any of the attributes of the mask. If he was looking for the mask in Heraklea he will get the link www.heraklea.info.mk where information about the mask can be found.

V. Real Implementation

All the system parts (database, server application, client application and web application) are implemented. As database we use the Oracle system, Server application was developed with Borland Delphi 7, Client application with Microsoft Visual C++ and web application was developed in JSP.

System was set up and tested with small amount of data. Results are very good. As soon as categorization of national heritage problem is solved, we will implement the system as a part of CDNH (Center for Digitization of national heritage) and test it in the real world.

References

[1] Database System Concepts, 4th Edition, Abraham Silberschatz, Henry F. Korth, and S. Sudarsan, McGraw-Hill Higher Education, 2001
[2] Chao Chih Jang, relational Databases, Prentice Hall 1986
[3] Slavko Tkalac, relaciski model podataka, Informator Zagreb, 1984 .
[4] Mastering Delphi 7, Marco Cantu Sybex © 2003

[5] Sun Java Documentation
[6] Teach Yourself Visual C++ 6 in 21 Days, Macmillan Computer Publishing

mile@ii.edu.mk
svetla1@freemail.com.mk
nikuco@freemail.com.mk
gorank83@gmail.com
pmilosev@gmail.com