Bull. Nov. Comp. Center, Num. Anal., 11 (2002), 27–32
© 2002 NCC Publisher

The on-line library of algorithms and programs ACCORD*

S.P. Gololobova, V.I. Kozlov

The on-line library ACCORD includes algorithms for solving different tasks of mathematical modeling. The description of the algorithms is in the special database, which contains description of algorithms, the author software source code and executable components, which allows the users to try algorithms. The access to the library resources is put into effect through the Word Wide Web.

1. Introduction. The last fifteen years brought the qualitative change in the conception of the on-line documents. In the beginning of the Internet era, just an ASCII file and a set of graphical files were meant. In the early nineties, the conception of compound documents came into life. This made closer the meaning and appearance of the on-line document and the printed one. But the on-line documents became really significant and indispensable only when multimedia appears. Text, graphics, sounds, and video allow the user to comprehensively depict the subject of a document.

This is especially important in education, because it allows the use of different mechanisms of esthesia of information.

However, for the complete study of a subject it is not sufficient to read only one book or one article. Usually, one tries to get information on a subject from different points of view. To do this, the conception of the hypertext documents was developed in the last eighties. This technology allows creating a set of connected on-line documents with the navigation means from one document to another.

In the early nineties, the hypertext technology was extended to the Global Area Network. It is called the World Wide Web technology. Now, the Internet user has access to the documents in different parts of world using special search servers. In the middle nineties, the hypertext technology was combined with a multimedia technology. It was the beginning of the era of the HTML and Internet browsers. The unification of the HTML and database management systems has allowed the Internet user to form sets of documents himself or herself.

It is obvious that modern instrumentation for developing on-line documents allows the creation of powerful and handy and effective information

*Supported by the Russian Foundation for Basic Research under Grants 99-07-90402, 01-07-90367.

and learning resources. Unfortunately, these means are applied, basically, in the e-commerce and in advertising. Certainly, there are many different on-line libraries dedicated to different subjects. But in general, they are just sets of the ASCII files.

Here we present our point of view as far as the architecture and the functioning of the on-line libraries are concerned, which could be used in universities or in high technology enterprises. The functional structure and the architecture of the library are described using the UML diagrams (the Unified Modeling Language) [1].

2. The main subject of the library. Our on-line library is dedicated to issues of numerical algorithms in mathematical modeling. It includes topics on numerical analysis, numerical solutions of the boundary value problems of mathematical physics, the inverse problems of mathematical physics, statistics, geometrical modeling, etc.

3. The library architecture. The library is a classical World Wide Web application implemented in the Client/Server technology. This means that it inherits the architecture of such applications. They are called multilayer applications [2]. Their structure is shown in Figure 1.

The first layer is called Data Capture. This means that data are captured and converted from a human representation to a computer representation. The previous sentence could be more easily termed as punching in data. However, using such a type of definition implies that the user has something that they can actually punch-such as keyboard.

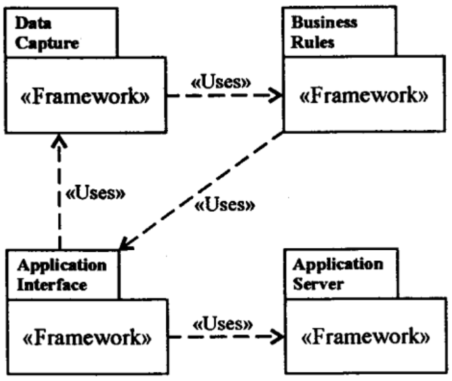

Figure 1

The reverse of input is output, and the statement also applies in this context. Examples of output devices are monitors, printers, or tape drives. Programming operations that are acceptable in this layer would be filling or reading the contents of a list box or a combo box, and then packing data into a pre-defined structure.

However, it is important to note that this layer is only responsible for the translation of data from one to another, human to computer or vice-versa. The actual contents of the data are not verified for correctness or accuracy here.

The second layer is responsible for applying the Business Rules to the data captured in the first layer. It is responsible for converting the data to a business context, and adding information about the business rules. The user does not interact with the software in this layer at all, however this layer is critical because it validates data to make sure that they are in the correct form-and are applied to the data that are both coming from or going to the server. The business rules must only be rules, they must not process the data. In our case, there are no specific business rules.

The third layer is called Application Interface layer. It is responsible for converting data from a business context to a technology context. The technology context is whatever the final layer, the Application Server layer, requires.

There are some peculiarities for the Web Client/Server applications. Layer one is the human-to-computer interface, and it would typically be a HTML based browser. There may be some client side controls or scripts to add to richness to the user interface, but this is purely optional.

Layer two is the business rule layer, and is generally handled on the client by the scripts and controls in the HTML page. In the current version of the library, we do not use special business rules.

Layer three, the conversion from a business context to a technological context, occurs on the server. It could be that an HTTP request triggers a routine that creates a structure of data, or that the ASP converts it to a new representation ready for the final layer.

Layer four, the application server layer, is again located on the server, and is the "back end" that actually does the processing and produces the results. This may be a database or other business object, and the result might be retrieval of information for return to the client, or just storage of data sent from it.

So in our Web-based model, layers one and two are on the client machine, and the server contains layers three and four. We will use the Microsoft NT Server with Internet Information Server and Active Server Pages as the server side technology. As the access to the database we will use the Microsoft SQL Server and Microsoft Access.

Let us consider how the structure of our library reflects this model.

4. Data capture. The first layer is a set of the HTML and Active Server pages implementing the functionality shown in Figure 2.

The HTML page in the use-case "Choosing the interface language" just allows the user to choose a preferred language (Russian or English). The rest use-cases should be considered more carefully.

4.1. Subject catalogue and alphabetical index. The first use-case is implemented in three components (Figure 3). The structure of the second use-case is similar (Figure 4). All Active Server Pages form an appropriate SQL query to the database.

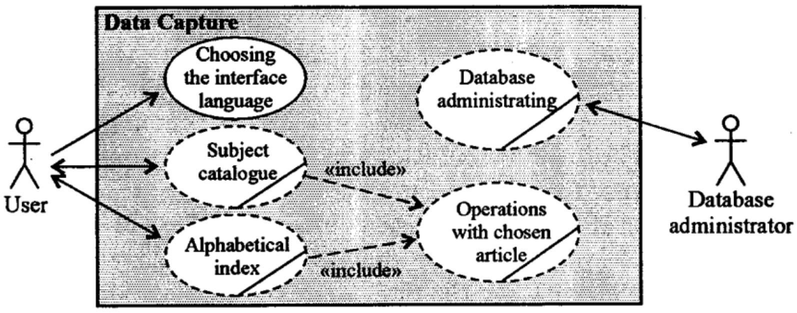

Figure 2

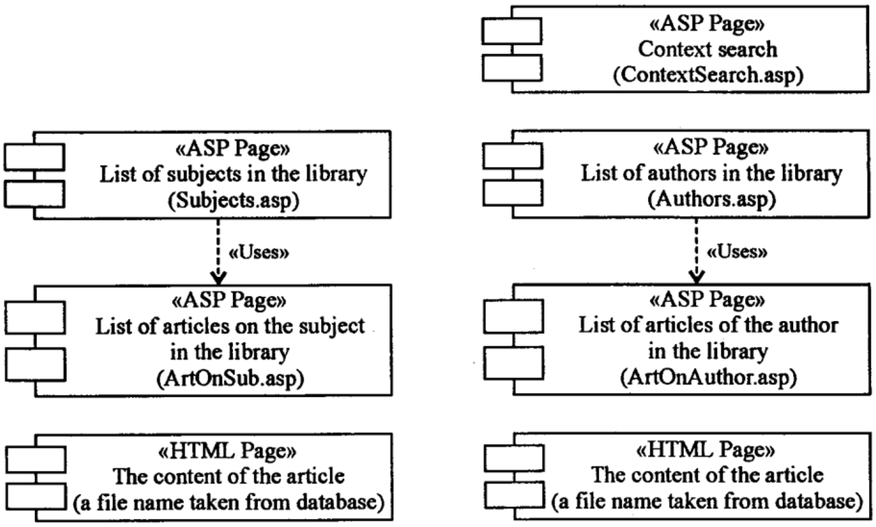
«ASP Page»
Context search
(ContextSearch.asp)
«ASP Page»
List of subjects in the library
(Subjects.asp)
«Uses»
«ASP Page»
List of articles on the subject in the library
(ArtOnSub.asp)
«HTML Page»
The content of the article
(a file name taken from database)
«ASP Page»
List of authors in the library
(Authors.asp)
«Uses»
«ASP Page»
List of articles of the author in the library
(ArtOnAuthor.asp)
«HTML Page»
The content of the article
(a file name taken from database)

Figure 3 **Figure 4**

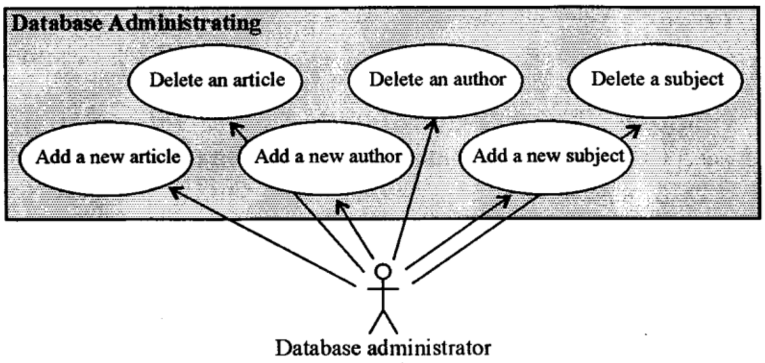

Figure 5

4.2. Operations with a chosen article. The end point of each use-case connected to the user is the context of an article. It is the HTML page, which contains the body of an article and hyperlinks to the source code of the programming realization of the algorithms described in the article and to the Active Server page, which allows the user to run a computational component illustrating algorithms in the article. The computational component is realized as ActiveX Control [3], which can be run from the ASP. Those components are registered on the same server where ASPs are. The results of the computation are put on the FTP server, which is a part of the library. The user receives an e-mail message about the completion of the computation. The message contains the URL of the file with the computation results. The format of the computation results allows using different CAD programs (like MathCAD or MathLab) for their representation.

4.3. Database administrating. This use-case includes standard database administrating operations (Figure 5). Every administrating operation is accompanied by checking of the database integrity. For example, if the administrator deletes an author from the database, all the articles of this author should be deleted as well.

5. Application interface. In our case, the application interface is the SQL queries. They are performed by the ADO objects included in the Microsoft Internet Information Server [4].

6. Application server. As mentioned above, we use the Microsoft SQL Server [5] and the Microsoft Access as DBMS (Database Management System). We will not discuss the structure of the queries as they are here. We will only consider the database structure, which allows the realization of all the use-cases of the system.

As one can see in Figure 6, the database allows forming a very wide set of queries. The developer of the Web application can easily create the ASP for getting lists of author or subjects in the library and realize all the operations needed to run the chosen article.

7. Conclusion. The authors believe the combination of technologies like the World Web Application, the modern Database Management Systems, the multimedia, and the component technology make it possible to develop very flexible and powerful on-line libraries. Such libraries allow the comparison of different computational technologies "on the fly" while studying them. This makes such a kind of research much more effective than reading conventional books and articles. On the other hand, such an architecture allows making modifications and reconfigurations of the library without any changes on the client side.

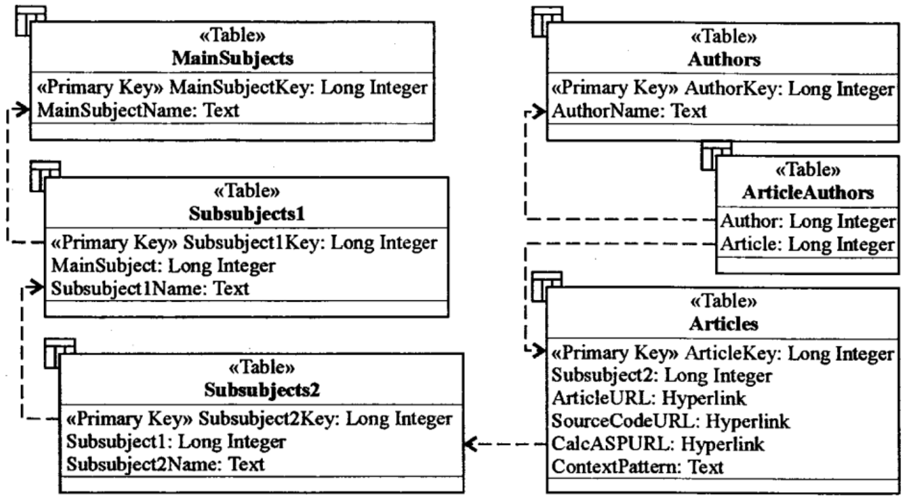

Figure 6

References

[1] Booch G., Rumbaugh J., Jacobson I. The Unified Modeling Language User Guide. – Addison–Wesley, 1999.

[2] Homer A., Enfield A., et al. Professional Active Server Pages. – Wrox Press Ltd., 1997.

[3] Li S. Professional Visual C++ 5 ActiveX/COM Control Programming. – Wrox Press Ltd., 1997.

[4] Williams A., Barber K., Newkirk P. Active Server Pages Solutions. – Coriolis Group, 2000.

[5] Moffatt C. Microsoft SQL server network integration architecture / Microsoft Technical Note. – 1994. – Vol. 4, № 5.