Creation of Financial Flows' Planning, Management and Control System

B.A.Gorelov
Vice-principal of economics
Moscow State Engineering Physics Institute
(Technical University),
Moscow, Russia

B.B.Gorelov
Postgraduate student of Cybernetics Department
Moscow State Engineering Physics Institute
(Technical University),
Moscow, Russia
E-mail: gorelov@orc.ru

Abstract

In this paper we focused on the problems and possible resolution keys which would arise when creating a system of financial flows' planning, management and control of a budget entity (university). This system aims to control agreements' execution, which Moscow Engineering Physics Institute (Technical University) signs with the third parties. The system is also able to forecast the future periods expenses and university cash inflow. The system accounts for the university's specifics which has certain scientific research orders and is deeply involved in commercial activities, apart from its state cash inflow.

Introduction

Financial resources of any entity have been under top focus nowadays, especially in Russian market economy at the moment. They are mostly crucial for the state budget entities. Moscow Engineering Physics Institute (Technical University) has a number of different cash inflow sources a part from state budget. This causes differentiation problems, together with planning and control of various resources income. The specifics of budget entities in Russia are in their difference from commercial companies' chart of accounts. The quarterly balance is being submitted not only to State Tax Inspection, but also to relevant ministries and committees. The expense statements of budget entity have to present the future periods planned expenses.

From this point it's reasonable to automate the process of Moscow Engineering Physics Institute (MEPhI) agreements with third parties fulfillment control together with future expense journals and budget classification reports pre-sets. One of the most important conditions seems to be the urgent agreements data update. That's why the following goals were set for this project: the agreement data update and storage automation; the setting and processing of expense statements modification.

Implementation

There's actually 3 ways to resolve such problem in general:

- To purchase the relevant application.
- To update and modify the actual application.
- To create and set a new one, relevant enough to achieve the above goals.

The university administration has chosen the third way taking into account the economic situation today, i.e. create a system for planning and control of financial flows. The coming question is: How to organize the centralized information database for the Institute with the given access to the employees of different departments?

"3M" Company has resolved the problem by creation of the 3-level decision making support system, structured as client/server [1]. Right now this system allows 400 USA company employees to require and analyze the information within client/server, old information systems and Internet infrastructure.

As "3M" many companies were providing the access to support system technologies for their employees with the help of analysis servers connected to information storage's. However Internet at the moment is getting to be the most frequently used application for the support systems optimization.

The aim of this project is creation and setting the program applications for the Institute cash flows planning. It doesn't make sense to give the full organization outline of the system

Permission to copy without fee all or part of this material is granted provided that the copies are not made or distributed for direct commercial advantage, the CSIT copyright notice and the title of the publication and its date appear, and notice is given that copying is by permission of the Institute for Contemporary Education JMSUICE. To copy otherwise, or to republish, requires a fee and/or special permission from the JMSUICE.

Proceedings of the Workshop on Computer Science and Information Technologies CSIT'99
Moscow, Russia, 1999

in this article [2]. However the goal achieving process is given on Figure 1.

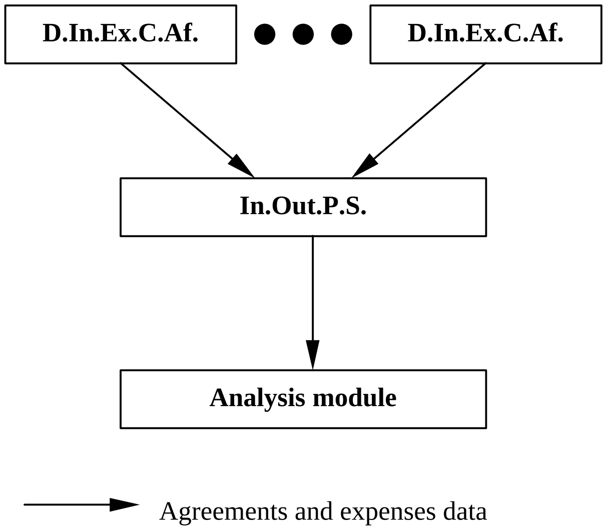

Figure 1: The goal scheme, given in this paper.

***Abbreviations: D.In.Ex.C.Af.** – **D**epartment, **In**volved in **Ex**ternal **C**orporate **Af**fairs; **In.Out.P.S.** – **In**flow and **Out**flow **P**lanning **S**ystem.*

Software creation choice method

The certain method should be followed when creating the high-quality software. These methods could be generally classified as functional and object-oriented. Functional methods treat functions and such systems separately, while object-oriented ones unite functions and objects data. The traditional way of treating the software the software development is presented by the following functional methods: SADT and RDD. Functional methods separate functions and data, setting functions for system behavior and the data information presentation, which is operated by function. The system is described as a set of functions which exchange data. The only draw back of such way is the certain difficulty of such systems' support and modification.

Object-oriented method consists of objects identification and then the description of procedures, which brings into effect certain operations, presenting the object behavior. System, set in this way, is modified easier, because it's based on the actual sphere objects, not on the functional requirements to the system itself. The system project can be obtained which would not depend on implementation. Object-oriented models help to understand the demands' to the system, simplify projecting and system support. Object is the main construction point of such method. The object descriptions consists of its attributes and methods. Object-oriented programming bears certain features such as encapsulation, inheritance, and polymorphism. These are really useful when creating the numbered of simple application.

Visual FoxPro 5.0 was chosen as an instrumental means of programming in this project. Visual FoxPro 5.0 (VFP) is a new version of the well-known databases management system Microsoft FoxPro, which operates within Windows'95 and is actually 32-level application [3]. VFP represents a new programming style, which bring it up on the new stage. It's an object-oriented, visual programming language, operated by actions and completely relevant to the new required programming means. The very distinctive feature of VFP is co-existence with the previous versions of VFP, which allows created earlier applications' transfer to Windows'95 quite easily. VFP uses Windows means to communicate with its other applications. Experienced user has great variety of possibilities to modify information with the help of full-functionally integrated programming field on VFP. Using ODBC standard [4] VFP supports the most popular SQL-servers access of Microsoft SQL Server, Oracle, Informix, etc databases.

System setup and development

It's not enough anyway to use only programming means, even such top ones as the 4-stage language for creation of safe and effective programming applications. The basement is needed - data architecture. The object-oriented set allows to identify and organize the object sphere. This method is based on the real world construction models. With the help of such models the system project can be created which doesn't depend on its performance. The object-oriented modeling and projecting helps to understand the demand to the system, simplifies projecting and system's support. An object is the main construction unit of the system. Its description consists of its attributes and operations.

The inquiry model was set in order to avoid the incomplete and controversial impression of the application. The set of the application starts from taking into consideration the user's wishes. It makes the application to meet the user's requirement us close as possible, which simplifies negotiations of the model with the customer and its modifying. The model itself consists of 3 parts: applications behavior model from the user's point of view, object model, and interface description. The behavior model uses the following terms:

- "User" or "Class of users" to present a user's role when working with this application;
- "Use case" or "Work script" to present a certain way of application usage. Each "Use case" is a definite consequence of actions done by user in this application.

Users can be set out of the description, since they represent the external world of applications. They are determined according to the roles which they present in co-operation with the applications.

"Dogovor" application is set for planning and control of financial flows of the Institute. Thus the user is the main user, who analyses the data received and makes actions due to this information. When using "Dogovor" application we can classify users into 2 unequal groups operators and administrator. The operator are making data entries of agreements, stages, acts, customers, expense statements, make information corrections. The administrator is the next to work with the application, and he's responsible for the database in general.

His main responsibility is backing up information and restoring the data in case of program bugs and maintenance problems. Besides the administrator checks the program's unity and makes system corrections and budget expense journal pre-sets.

"Use case" is a presentation of internal program process. Each "use case" describes the certain way of application's functionality and sets the user's and application's cooperation. A number of "use case" represents all the application's possibilities. Users can show them all. The set of operations is used for the full application usage, and performed easily by any user.

Database project

Visual FoxPro 5.0 is the operational system of relative databases, which are now commonly used and are nearly production standard [5]. In this version all relative DBMS attributes are set. First of all database definition is given here, which consists of tables group. In this database you can define the data unity conditions by primor and external table keys. VFP keeps triggers and stored procedures, which allow centralized analysis at any database changes.

Two major problems are resolved when projecting the database:

- object reflection to abstract models without opposition to its semantics together with efficiency. This problem is also called a problem of logical database projecting;

- rational pre-set of data in the memory which allows the efficient requirements meeting, creation of the useful additional structures (indexes).

The direction of semantic models was set by databases programmers' needs. Each advanced semantic database model as well as relative model consists of structural, manipulation and solid part. The main point of semantic model is their provision of data semantics performance possibility.

ER model is considered to be one of the most popular semantic database models. Most of relative database are based on its usage. It was suggested by Chen in 1976 [6]. Objects modeling is based on graphic diagrams usage, including a small number of unrelated components. One to the presentation visibility of conceptual databases ER models became widely used in CASE systems, which support autoprojection of related databases.

System implementation

Standard objects identifying when creating he application is considered to be one of the best methods of production and development growth. Such users classes on the basis VFP ones were used when developing "Dogovor" application. In order to achieve the full standardization of the system several users' classes were created: data entry class (text-box), which is available read-only, edit data entry class (text-box), etc. If I'm forced to change the standard in future, it would be enough to change the users classes which are actually applications objects.

Class libraries are used for storage of users' classes created in VFP [3]. All these classes for "Dogovor" application are kept in 2 class libraries. We used subclasses when creating applications, which allowed me to use users' classes as a basement for next classes creation. Since the subclass inherits all the mother class features, it was enough to add new features and methods or change the base class features a little bit.

Let's see how the subclasses were used, looking at the example of several new classes for data entry creation.

- subclass "Base-textbox" for data entry Field, which sized, modified by field color, font, font color.

- depending on this class, several subclasses were created for data entry, which differed in data type or symbol numbers. For instance "Chr30" subclass allows to entry data 30 symbols length of Character type, "Nomer" allows the similar entrance (6 symbols length) and "Rpsumm" gives on opportunity to enter currency up to 99999999,99 RUR.

If there's a need to change reflections features of data entry fields in the general, it would be enough to make corrections of these features in the first of classes created ("Base-textbox" data entry field) and the corrections would spread in all the subclasses hierarchy which refer to this class. In addition I would like to give the structure of "Dogovor" system, which is shown on Figure 2.

Conclusions: interpretation of the result

When resolving the problems we did the software market research in the field of company's activity planing. Several systems were considered from the point of view of such problems resolution. We also analyzed other entities' experience when having the same challenges. The software in use either were inappropriate for the above goals or too costly.

We also checked for budget entities and did the actual analysis of MEPhI software package. All the peculiarities of organization and technical system of Institute assets were taken into account. Detailed research was also done in information flows analysis. The 10 top debit sources of MEPhI were outlined, and basing on the results received the application model was set. For instance the users' classes were created which form the external applications' world. We also described the applications functions by identifying and describing "use cases" of the applications.

The objects, which data should be kept active after program Shut-down were set out. These objects actually formed the database. The ER-diagrams were created for applications databases. We also did relative database projects. Visual FoxPro 5.0 was used as a system of databases management. This allowed me to set the cooperation of these applications with other MEPhI software package.

When setting and creating the applications object-oriented advantages were used such as inheritance, encapsulation, polymorphism. As a result I've built the objects hierarchy: hierarchy of Windows, users classes. Inheritance does well for objects' re-use and simplifies the process of solid interface with user organization. We also created the class libraries, which served as a basis for the interface of applications in general.

We actually based on the projecting and developing experience when creating the applications. This experience was gained during pre-diploma practice when I was developing the system of the Institute's areas accounting support system. As a result the paper "Object-oriented tools in SQUARE application for the enterprise areas registration support" [7] was published within the international conference ADBIS'97.

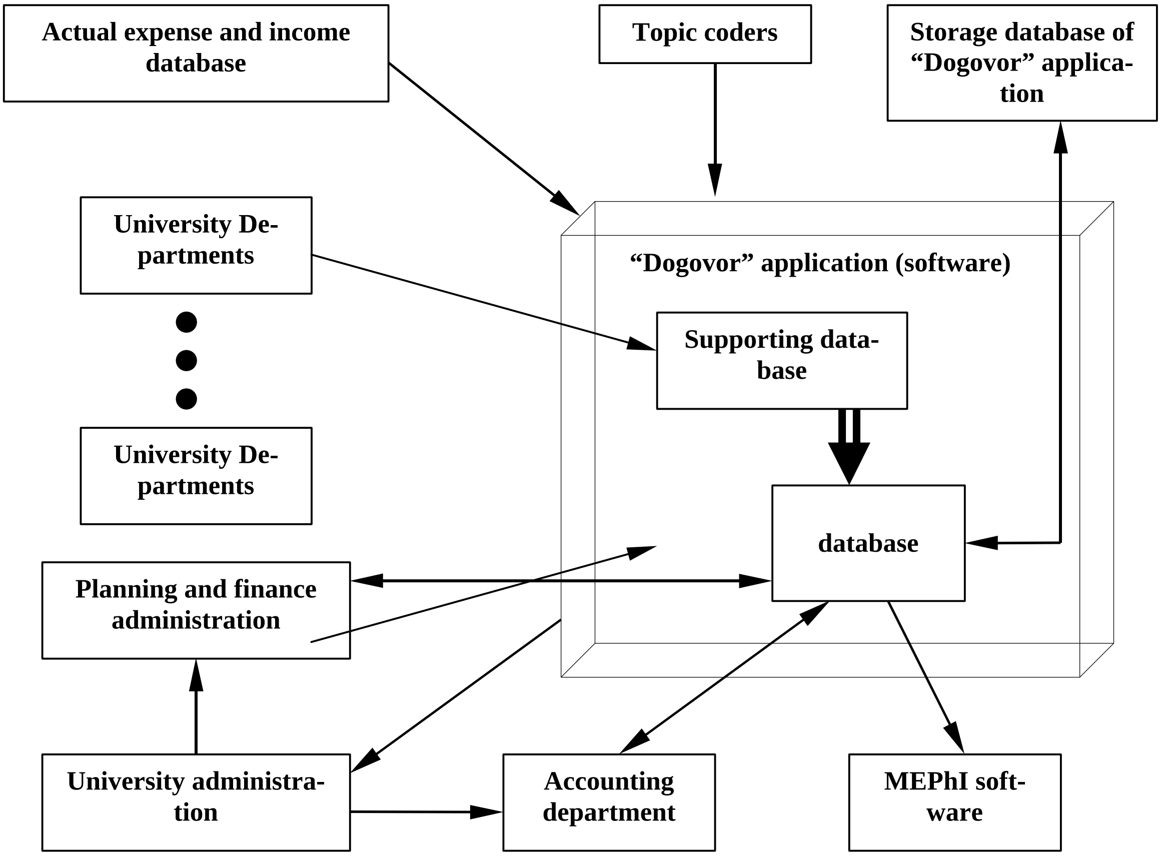
Actual expense and income database
Topic coders
Storage database of “Dogovor” application
University Departments
“Dogovor” application (software)
Supporting database
University Departments
database
Planning and finance administration
University administration
Accounting department
MEPhI software

Figure 2: Information flows of Dogovor-application.

References

1. Software meas of decisions making. Computer week 32(238).

2. B.A. Gorelov. Scientific conference MEPhI 98 workout. Moscow, 1998.

3. S. Caratygin, A. Tihonov, L. Tihonova. Visual FoxPro 5.0 top proficiency level. Moscow, 1997.

4. R. Signore, M. Stegman. The ODBC solution. Moscow, 1995.

5. S.D. Cuznecov. Introduction to database management system. DBMS 1-9, Moscow, 1995-1997.

6. P. Chen. Entity-Relationship model. 1976.

7. B.B. Gorelov. Object-oriented tools in SQUARE application for the enterprise areas registration support. ADBIS, 1997.