OBJECT-ORIENTED MODELING TOOLKIT
AND TECHNOLOGICAL FRAMEWORK FOR
LEGACY INFORMATION SYSTEMS REENGINEERING

Software reengineering is a very difficult and expensive process, which requires as a lot of “mechanical” activity: by restructuring of the legacy source code to a new one, and a human intervention by high-level restructuring: for the changes needed in overall legacy system behavior scenario, and in the system architecture. That is why a very important problem is the forming of an effectively Modeling Toolkit and Technological Framework (MTTF), which would be adequate for appropriate OO reengineering paradigm. One of the key component of this complex will be the Domain Ontology Maintenance Facility (DOMF). It is designed and will be implemented as the completely Internet/intranet-based application, using J2EE, JSP, and Oracle 8i - Suite technologies. It also allows the external experts, who have to be involved in the process of system reengineering or maintenance, to participate on ontology specification processing in the remote mode via Internet.

1. Usage of object-oriented approach for legacy information systems reengineering

For the time being the problem of legacy information systems (LIS) reengineering is one of the most important in the whole SW&IT – research and applications domain. The usage of diverse reengineering approaches let make the maintenance of LIS less expensive, and keep the investments already enclosed in an appropriate system. With advent and fast development of the object-oriented (OO) approach, in the beginning of 90s years, its methods and tools also became successfully be applied for reengineering tasks of existing LIS. They were modular systems with hierarchical software architecture, which had been designed and implemented using the procedural paradigm, and procedural programming languages such as PL-1, Cobol, Fortran etc. The several approaches for source code migration and transformation into the new OO environments, e.g. into C++ or Ada have been developed and used [1,2].

On the other hand, by the transition to the next more advanced forms of OO software design, namely component-based software architectures, and reusable software component frameworks (see e.g.[3,4]), it also became clear, that only the usage of OO programming languages such as C++, Modula-2, Smalltalk etc, and post-relational OO DBMS [5] also, can not ensure in itself the strategic goal of LIS reengineering activities automatically. This goal for any LIS is, at the certain phase of its life cycle, to establish and to support such a form (a model) of system architecture [4], which is as much as possible invariant concerning the influence of diverse disturbing factors, and which could later be improving evolutionary during the system's operation [6,7].
Thus, we understand by the term *system reengineering* the process of the designing, the implementation and the usage (for an appropriate LIS-class) a *special instrumental complex*, which includes *models, architecture solutions, and information technologies* with respect to the reengineering goal mentioned above.

The LIS usually are the complex computerized systems for data capture and data processing, for decision making support etc., which earlier have been developed, and are used at present in any industrial (business) organizations, and which have a *vital importance* for providing of their normal production (business) processes [6,8]. They are supposed be operating very stably, and they include a lot of local databases and application programs. Their *stored data* and *business logic* collect and represent the *applied knowledge* of an appropriate organization. Accordingly to this, if we are going to provide a effective system reengineering, it means that not only the data structures and the program organization should be changed, but also the *knowledge* about LIS development and about its operating history have to be re-organized, to be analyzed, and to be processed permanently [8,9].

Why actually the organizations having LIS in their structure are motivated (or are forced) to perform the reengineering activities? In some later publications concerning these issues [10,11] the following main reasons are listed:

- Growing trend toward unbundling of LIS into diverse subsystems, that can be tested, delivered or marketed separately
- Real challenges to system performance improving
- Necessity of porting LIS to other hardware or operating platforms, and exploitation of new information technologies

Below we compare these factors with the real situation with LIS at the Ukrainian industrial enterprises in one of the high-technological branch: at some technical objects of Ukrtransgas, where nowadays the problem of information infrastructure improving becomes more and more actual [12,13]. We are deeply convinced, that it could effectively be done only using the combined approach: the elaborating of new systems, and the reengineering activities for LIS.

2. Our LIS Reengineering Case Study

During last two years our research team at the Kharkiv National Technical University “KhPI” in the closed cooperation with the design firm «PromAvtomatika» (Kharkiv, Ukraine) is working on some reengineering tasks for LIS in gas-compressor stations (GCS). Such stations are the essential part of the gas-main pipeline system in the Ukraine. This system moreover is a part of the gas pipe-line connecting Eastern Siberia (Russia) and Western Europe, which provides more than 90% of Russian gas-export volumes, and ca. 30% of the West-Europe annual consumes of this resource. There are more than 70 such stations in several regions of the Ukraine. We have been dealing with
reengineering issues for the LIS at the Romny GCS, which is situated at the northeast of the Ukraine as the first facility of this complex distributed technological system. The appropriate LIS is a typical example of large real-time information management system of the type “7 x 24 x 365”: it constantly is operating 7 days per week round the clocks during several years. In [7,14] some our results achieved in this project are presented more detailed.

By the analysis of the LIS GCS we have found out, that it has the 2-tier network architecture including several software control subsystems, partly so-called COTS (Commercial Off-The-Shelf) soft-and hardware solutions such as Serck Ltd. [15], WinTECH Software [16]. We also had provided some new software solutions for the selected subsystems, which have been integrated into the LIS structure (see, e.g. in [17]).

But despite of the steady work of these subsystems, and because of:

- the new requirements for main technological process (swap of large natural gas volumes by the certain temperature values, and with needed pressure in the main pipe-line)
- the changes arisen in the company’s information infra-structure, e.g. the building of intranet – system in the regional control office (based on MS Windows NT ? MS Internet Information Server)

the reengineering solutions have to be provided in the whole LIS. Thus we have defined the following main points [14]:

- elimination a functionality redundancy in LIS software applications using OO component-based technologies
- porting the legacy “file-server” data processing scheme to the 3-tier “client-server” architecture for the sake of process control quality improving
- providing an unified operating data visualization interface in the most important control subsystems of LIS based on Web-platform.

At the performing of all those reengineering activities we have been faced with the problem, how the motivated choosing of appropriate models and CASE-tools has to be done, with respect to the OO reengineering approach mentioned above.

3. Forming of Modeling Toolkit and Technological Framework

The reengineering of complex LIS is a very difficult and expensive process, which requires as a lot of “mechanical” activity: by restructuring of the legacy source code to a new one (sometimes called also as low-level restructuring [10]), and a human intervention by high-level restructuring(see the same): for the changes needed in overall legacy system behavior scenario, and in the system architecture. Besides that, it usually has to be done without completed system documentation, without the possibility to contact LIS former developers personally, etc. That is why a very important problem is, from our point of view, the forming of an effectively modeling toolkit and a technological
framework (MTTF), which would be adequate for appropriate OO reengineering paradigm. It also is to note, that these questions are not discussed enough till now in the Ukrainian SW&IT-publications.

Thus, e.g. in [18] some criteria for such MTTF are formulated, and the one of possible choosing variants is proposed. Its structure really reflects the main stages by the OO design and implementation of any IS, but at the same time, from our point of view, it has the following shortcomings:

- such important conceptual component as domain ontology modeling (see e.g., in [19,20]) is not presented as a whole in this MTTF, and the appropriate CASE-tools are lacking there also
- the usage of so-called design patterns (e.g., in [10]) accordingly to this scheme is supposed to be considered only after legacy system analysis and modeling, that seems to be not correct completely from the methodological point of view.

More correct methodologically, and more completely concerning CASE-tools, which are needed for this purpose, is the MTTF-structure depicted on the figure 1. It supposes the following framework by the providing of OO reengineering for LIS, and includes such modeling concepts and tools as (see figure 1):

1) At the first stage of the reengineering process the ontology specifications as an effective modeling abstraction for the representation and analysis of application domain’s information and knowledge can be used (in this case by the term ontology its pragmatic meaning should be understood [20]). The several special CASE-tools such as Protege’2000, SMART, Ontolingua [21,22,23] are supposed to support this task. The example of some ontology specifications for application domain «Reengineering and Maintenance of Complex LIS for Technological Process Control» is given in [14].

2) In parallel for this activity the analysis of existing LIS software architecture using design patterns or architectural pattern solutions will be carry out, and an appropriate software architecture type by the LIS has to be defined. The analysis result for the considered LIS GCS is already discussed in the Section 2 of our paper.

3) At the next stage, which is dedicated for the modeling and elaborating of the improved (target) software architecture for appropriate LIS, the usage of Unified Modeling Language [24] should be recognized at the most powerful description environment. The well-known industrial CASE-tools such as Rational Rose, System Architect, Visual Modeler and some others can be used for this purpose.
Figure 1. The MTTF structure and content
4) The architecture design solutions, which are elaborated at the stage described above, can be implemented using one of two most prevalent at present component-based software technologies: MS DCOM / COM+ with MS ActiveX [25], or Java Beans / EJB (Enterprise Java Beans) of Sun Microsystem [26]. The CORBA standards (IDL specifications) [27] could provide additionally the possibility of usage the both type of components together.

As shows our own experience collected by the performing of real-life reengineering projects, the most actual and complex problem by the usage of such MTTF is the activity (1): elaborating of domain ontology specification. It is also not computerized enough, because the existing CASE-systems are quit limited in their options, and mostly are research facilities only (see e.g the system Protégé’2000 [21] ).

That is why on the Department of Information Management Systems, at the National Technical University “KhPI” we are working on the elaborating of the concept and tools for integrated reengineering environment. One of the key component of this complex will be the domain ontology maintenance facility (DOMF), some examples of its user-interface as the screen snap-short are presented on the figure 2.
It is designed and will be implemented as the completely Internet/intranet-application, using J2EE (Java 2 Enterprise Edition), JSP (Java Server Pages), and Oracle 8i Suite technologies. This solution should also be integrated in the Web-based target distributed system architecture, elaborated and proposed for the LIS GCS [28]. It also allows the external experts, who have to be involved in the process of system reengineering or maintenance, to participate on ontology specification processing in the remote mode via Internet. It could be also used for the content management in the Web-based system for the gas-compressor station staff training, which is also in the development stage in the framework of our reengineering activities [29].