
SEMINAIRE

СЕМИНАР

SEMINAR

STATISTICAL COMMISSION AND
ECONOMIC COMMISSION FOR EUROPE



Distr.
GENERAL

CONFERENCE OF EUROPEAN
STATISTICIANS

CES/SEM.38/22
29 April 1998

ENGLISH ONLY

Seminar on integrated statistical information
systems and related matters (ISIS '98)
(Geneva, Switzerland, 27-29 May 1998)

Topic (iii): management of development and operation of
statistical IT projects, applications and infrastructure

**DEVELOPING STATISTICAL SYSTEMS USING OBJECT TECHNOLOGY
WITH THE "SELECT" PERSPECTIVE APPROACH**

Submitted by the Central Statistical Office of Poland¹

I. INTRODUCTION

1. The paper presents a method that is applied for system development in the Central Statistical Computing Centre of Poland. Production of public statistics can be considered similar to a large enterprise with complex business processes where data has to be delivered according to tight time-schedules while facilitating re-use.

2. The SELECT Perspective approach (elaborated by Stuart Frost) expands system modelling from a collection of unrelated systems to an integrated hierarchy of cooperating systems. It provides a coherent software development method for the client/server environment with the following characteristics:

- Process-driven analysis and design.
- A four-schema architecture that enables the re-use of business objects and supports distributed client server systems.
- Object-oriented techniques with UML (Unified Modeling Language).
- Incremental delivery which is a key element of the development process, as iterative prototyping.

3. We have used this method for the development of the National Business Register (REGON). Based on our experience, the same approach could be successfully integrated into the Polish public statistics system. The paper also outlines the structure of the Polish Business Register REGON.

¹ Prepared by Romuald Iwaszkiewicz and Wojciech Nowak.

II. PUBLIC STATISTICS REGISTERS IN POLISH PUBLIC ADMINISTRATION

4. The public statistics maintains two kinds of registers:

- The Polish Business Register, including agricultural farms;
- The National Register of Territorial Division.

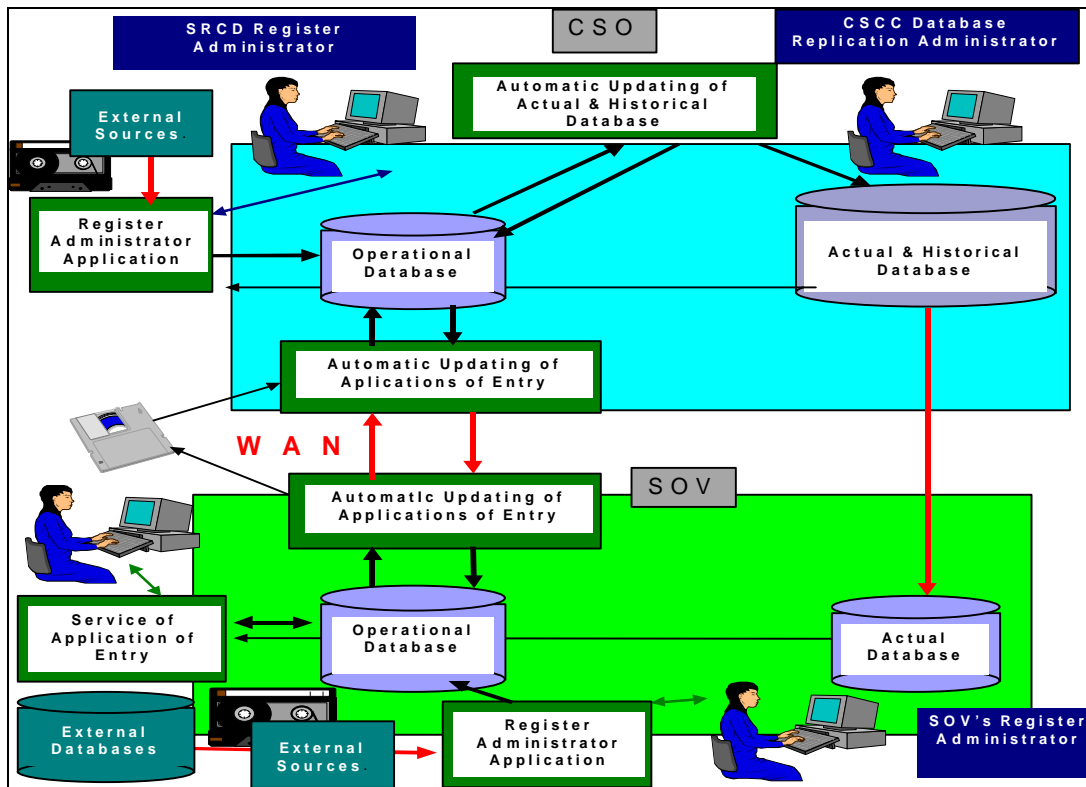
The IT systems maintained by other agencies of public administration are weakly integrated with statistical registers. The information exchange between these systems is performed by updating files which are processed by separate procedures.

III. STRUCTURE OF THE BUSINESS REGISTER OF POLAND - REGON

5. The Business Register system has the following functions:

- At the level of the Statistical Offices of Voivodeships:
 - Registration of the social and economic events on the voivodeship's territory, especially registration of economic units, statistical survey data input and editing.
 - Data processing and dissemination to local recipients, including public administration.
- At the level of the Central Statistical Office(CSO):
 - Maintenance and dissemination of the basic registers, nomenclatures and classifications.
 - Aggregate data processing at country level for the majority of statistical surveys.
 - Country data processing and dissemination to:
 - ✓ Public statistics system (survey results, analysis, printouts).
 - ✓ External recipients (country aggregates, excerpts from registers, analytical comparisons).

The basic structure of the designed register servicing system is shown below.

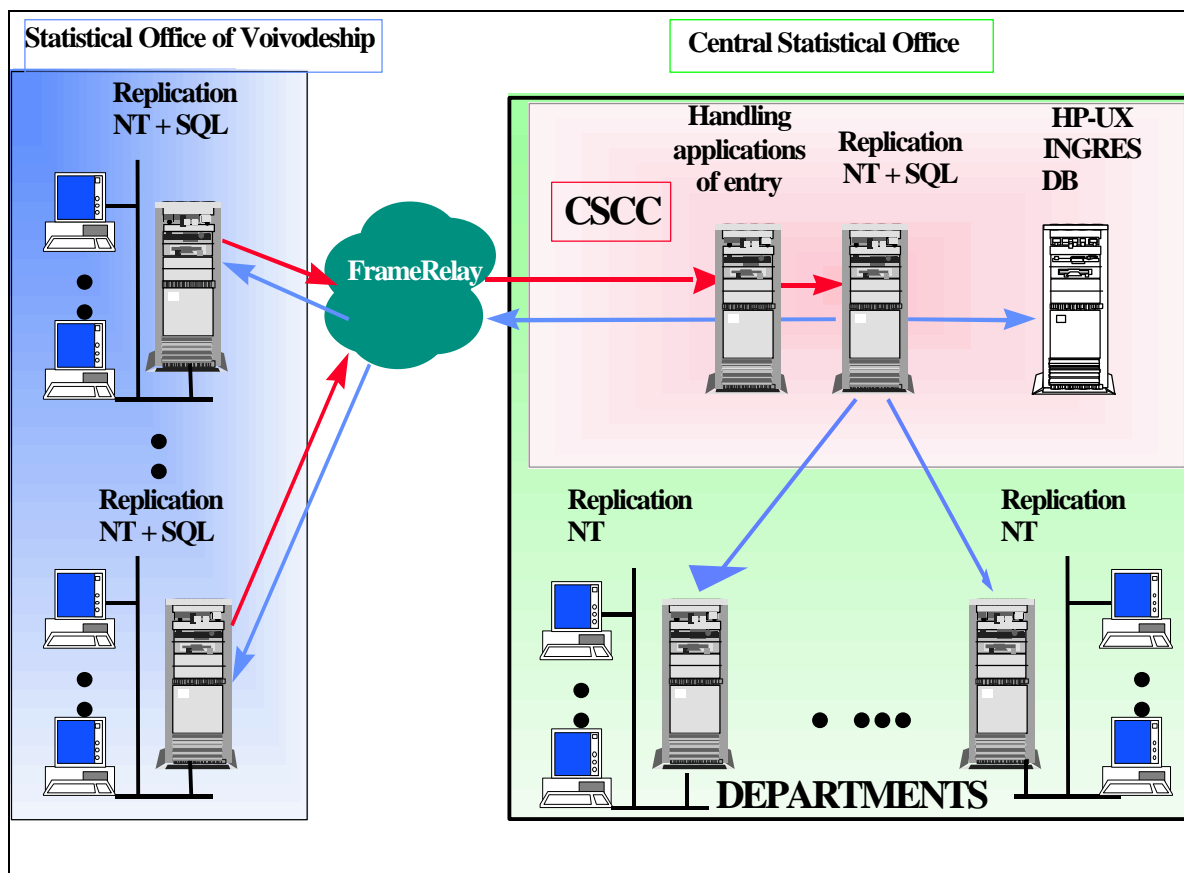


IV. HARDWARE ARCHITECTURE OF THE POLISH STATISTICAL SYSTEM

6. The Polish public statistical system forms an intranet with CSO and Statistical Offices of Voivodeships (SOV) as nodes. Each node constitutes separate local network with a hierarchical structure linked with other nodes through a wide area network based on FRAME RELAY. Hardware architecture of The National Business Register includes:

- about 250 workstations in 49 voivodeships (Pentium 120, 32 MB RAM, MS Windows 95),
- 49 local servers (ProLiant 2500, 128 or 256 MB RAM, MS NT WorkStation 4.0, SQL Server)
- 2 central servers located in CSO in Warsaw (ProLiant 2500, 256 MB RAM, MS NT Server 4.0, SQL Server).

The scheme of the network is presented below.



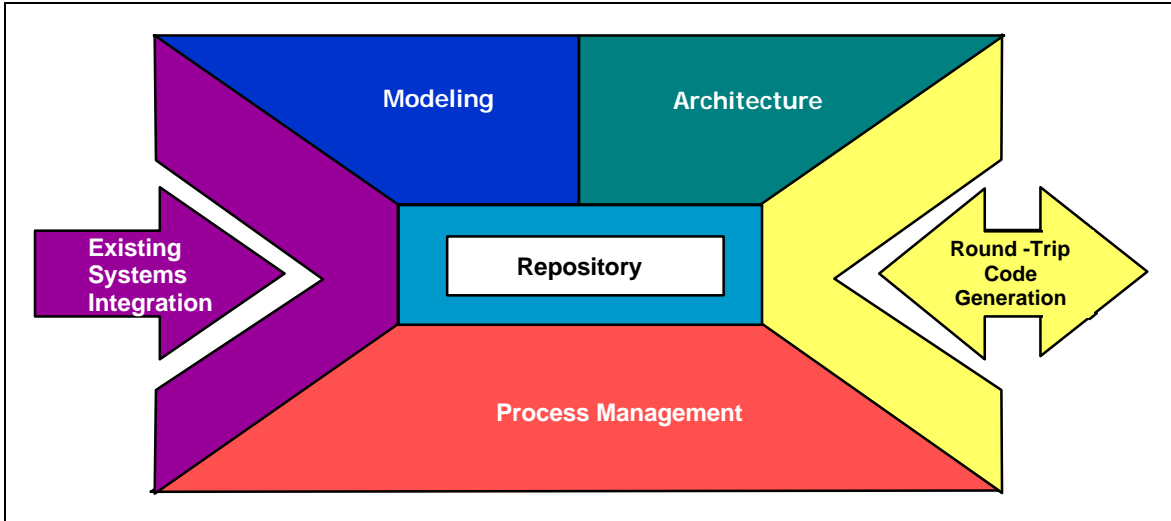
V. THE SELECT PERSPECTIVE

7. The SELECT Perspective, first published in 1991 by Stuart Frost and Paul Allen, is a component method for developing enterprise systems. We have chosen that method for developing statistical systems. SELECT Enterprise for Visual Basic is a modeling toolset for building scalable client/server applications in Microsoft Visual Basic Enterprise Edition (MS VBEE) and implementing the Perspective development approach. It incorporates object-oriented modeling support for multi-tier client/server applications, and an integrated code generator for a scalable application architecture in VBEE. The SELECT Perspective has many parallels with Microsoft Solution Framework (MSF), adding industry standard object modelling techniques to the architectural and process frameworks espoused by MSF.

8. The SELECT perspective is a collection of best practice modelling techniques. They are applied and adapted using process templates across a wide range of developments in a component-based setting. The SELECT perspective uses a set of standardized notations that employ Unified Modelling Language (UML) in a pragmatic way. The Perspective framework includes:

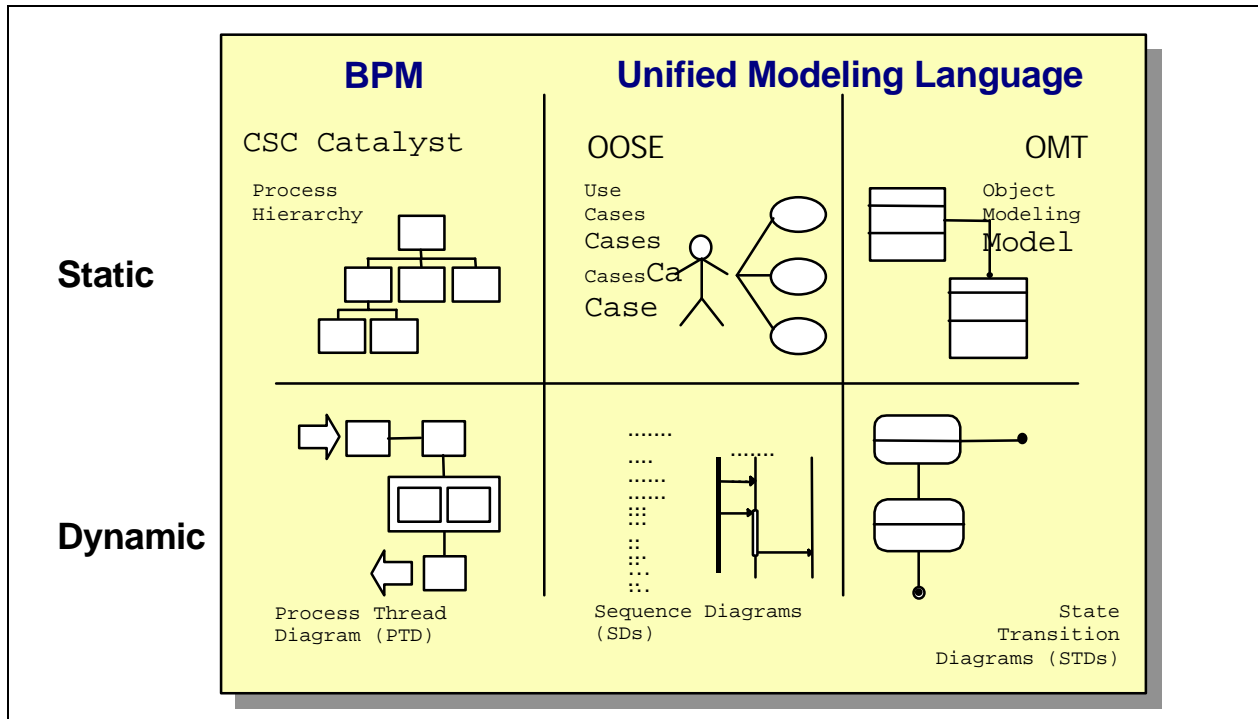
- Modelling (methods and techniques)
- Architecture (scalable, distributable, re-useable)
- Process management
- Repository (component management)
- Legacy component management (component wrapping, existing system integration)
- Solution process management (code generation, round-trip code generation)

A general framework of the SELECT perspective is presented below.

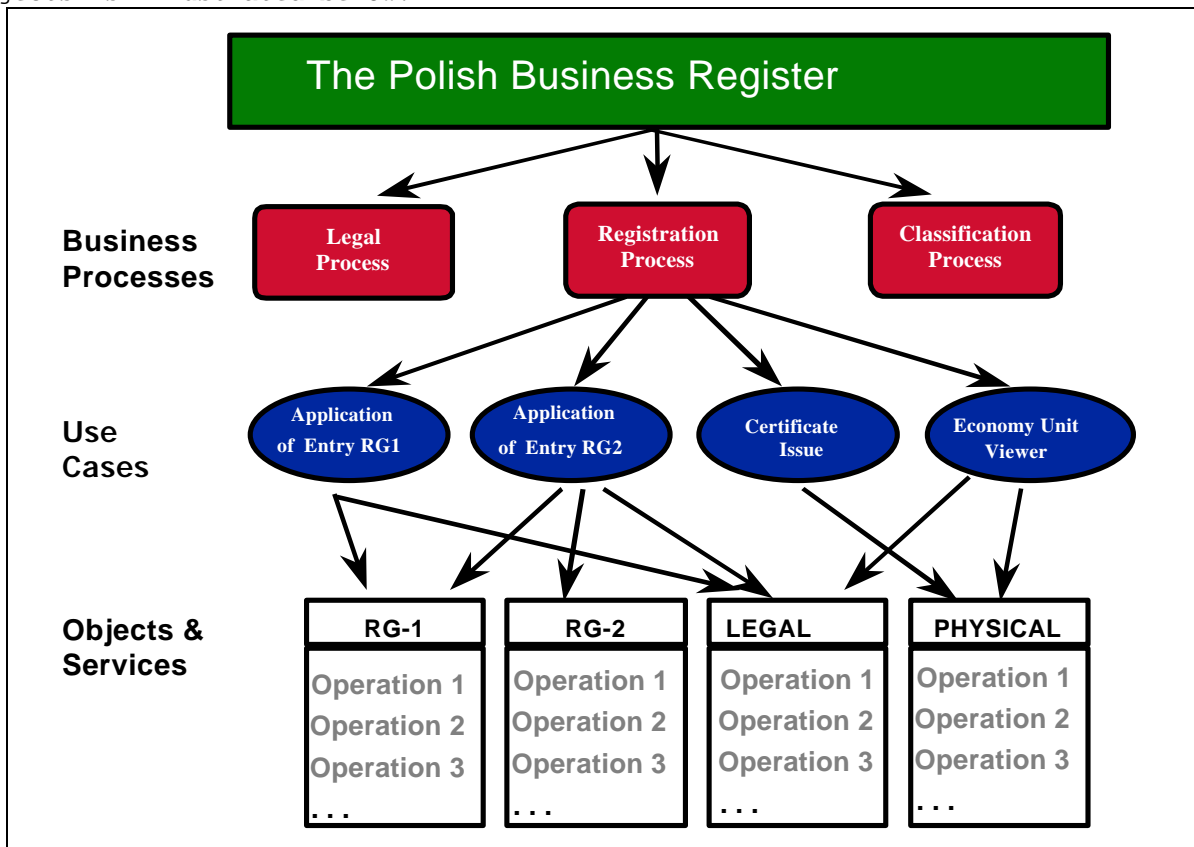


VI. MODELLING TECHNIQUES

9. Each software system has a specific functional, static and dynamic properties, which need to be precisely defined. Functional properties are defined using business process modeling (BPM) concepts, adapted from CSC Catalyst. Business Process Re-engineering (BPR) and Business Process Improvement (BPI) encompass a whole range of techniques, covering organizational changes and Business Process Modeling (BPM). The other techniques include I. Jacobson's use cases and interaction diagrams, J. Rumbaugh's object models and D.Harel's dynamic models. SELECT repository and diagrams are tightly integrated, allowing developers to choose whether they update models on screens or through the dictionary. Any change performed in any model immediately implies updating related models.



10. Modelling allows to design large complex systems, and helps users and developers to communicate. It permits tracing from the business model through to the final system, maintains a central repository for business rules, improves the overall software quality and reduces long term costs. The excerpt of an overall business process hierarchy with a subset of use cases and objects is illustrated below.



11. Among modelling benefits we can distinguish:

- Direct link to business requirements;
- Minimalistic approach;
- Pragmatic, easy to learn;
- Integrated industry standard techniques;
 - CSC Lynx BPM, Rumbaugh OMT, Jacobson OOSE, UML.

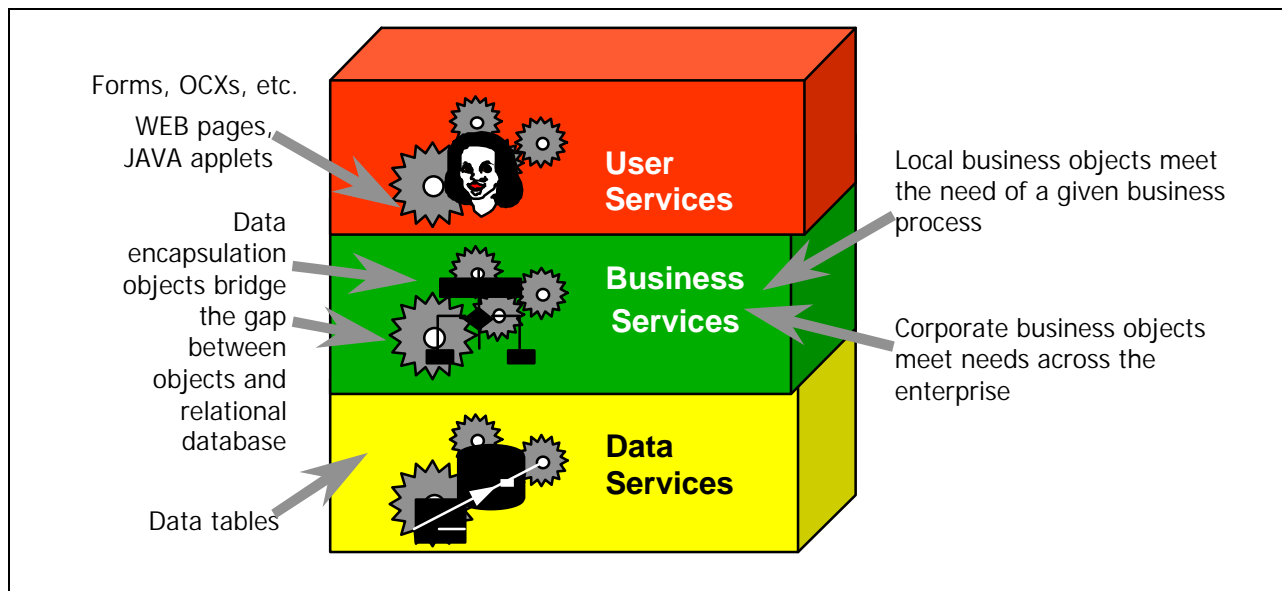
VII. ARCHITECTURE

12. The SELECT Perspective architecture works on the basis of the following service categories:

- User services are linked together with business services in order to deliver the business capabilities to users;
- Business services convert data received from data services and user services into information. A business service joins related business tasks with relevant business rules;
- Data processing is independent of the underlying physical storage implementation.

13. The advantages of such an approach are:

- The capability of changing is built into the system;
- Business logic and technological principles are different;
 - The system can grow and change without needing fundamental redesign;
- Can be directly mapped to multi-tier client/server environment;
 - System can be distributed without fundamental redesign;
- The enterprise-wide business needs are different from those of each business process
 - Identifiable re-usable components.



14. The interface objects allow interaction with the system. They constitute the system presentation layer. They can be implemented as screens, menus or dialogs, depending on the chosen technology. The design is always accomplished by prototyping, and can be implemented in Visual Basic/ PowerBuilder, etc.

15. The local business objects are project/application specific and define a set of business requirements. The local business objects can be:

- New objects - designed for that application;
- Objects derived from corporate objects by inheritance or aggregation.

16. Corporate business objects meet the needs of business processes across the enterprise. Their sources are:

- Top down modelling;
- Pilot projects;
- Legacy databases;
- Corporate data models;
- Legacy applications.

They can be implemented e.g. C++.

17. Storage objects allow to store data in persistent layer or corporate and local layers. They are implemented with the help of relational databases or object databases.

VIII. PROCESS MANAGEMENT

18. The basic paradigms of process management philosophy are:

- Iterative and incremental process development based on DSDM (Dynamic Systems Development Method) MSF (Microsoft Solution Framework);
 - Uses prototyping and JAD (Joint Application Development);
- RAAD (Rapid Architected Application Development) based on components;
- Component re-use;
- Compact project teams with active user involvement;
- Timely delivery of both business systems and components;
 - Time estimation built into the development process.

IX. COMPONENT MANAGEMENT

19. The component is an executable unit of code that encapsulates specific services. The services can only be accessed through a consistent interface based on an interaction standard. It must be possible to link components (through communication interface) to form a larger group. It is a key to reusability. We can provide an analogy:

- Applications provide use cases
- Components provide services
- Objects provide operations

Applications are composed of components, which are composed of objects.

20. Good component management requires a repository. The repository includes a definition of component interfaces and the services supplied through those interfaces. The repository also allows to identify models that are associated with components, and that can be composed, extended and adapted through the modelling process. The repository contains all project information which is consistent and sharable. It helps developers and analysts communicate. It facilitates re-using all components, and helps to manage the code, components and the whole model. It also enables effective process control.

21. Component-based development allows enterprise systems to be composed rather than constructed. According to the Gartner Group, by 2001 at least 60% of all new applications development will be based on assembling Componentware

22. Component management benefits include:

- Components provide higher levels of re-use based on business objects;
- Components provide the glue for legacy integration;
- Supports the "re-use before you buy; buy before you build" philosophy;
- Implemented by industry standard technologies - OLE/COM and CORBA;
- Facilitates composition as opposed to construction;

- Provides a focus for re-use within IT organizations.

x. LEGACY COMPONENTS MANAGEMENT

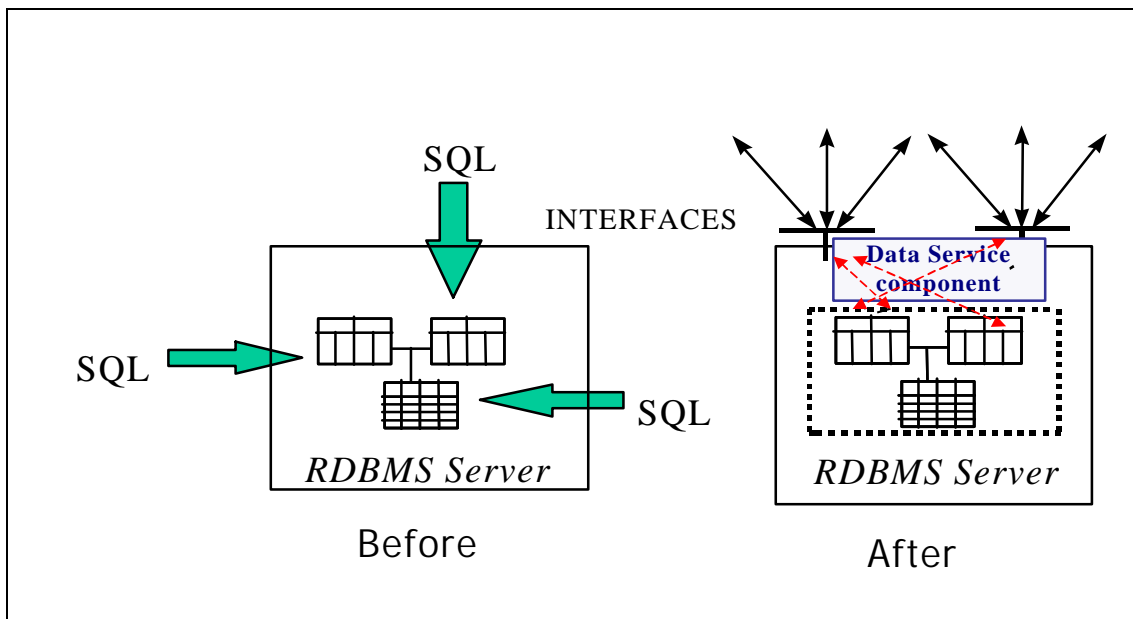
23. In Polish statistics there are at least 150 software products, developed on the basis of older technologies, which are so vital to statistics that they cannot be replaced or disrupted without a major impact on the statistics. These legacy models (such as data models) are often associated with the legacy asset. Legacy assets come in various forms: for example, function libraries, programmes, programme fragments, data structures, database interfaces, data models etc. SELECT offers wrappers that provide an object-oriented interface to non-object oriented software. Wrapping is a technique which takes legacy assets and encapsulates them inside components. Wrappers provide three key advantages:

- Insulating new component-based software from legacy code, in particular from the effects of changes;
- Re-using the code in a component-based setting;
- Migration to component technology while protecting investments in existing code and at the same time providing new innovative solution to statistics problems.

24. The component provides a set of services and benefits that include:

- Hiding data structure/storage from application;
- Allowing distribution of data from application;
- Re-using of services across organization.

An illustration of wrapping is provided below.



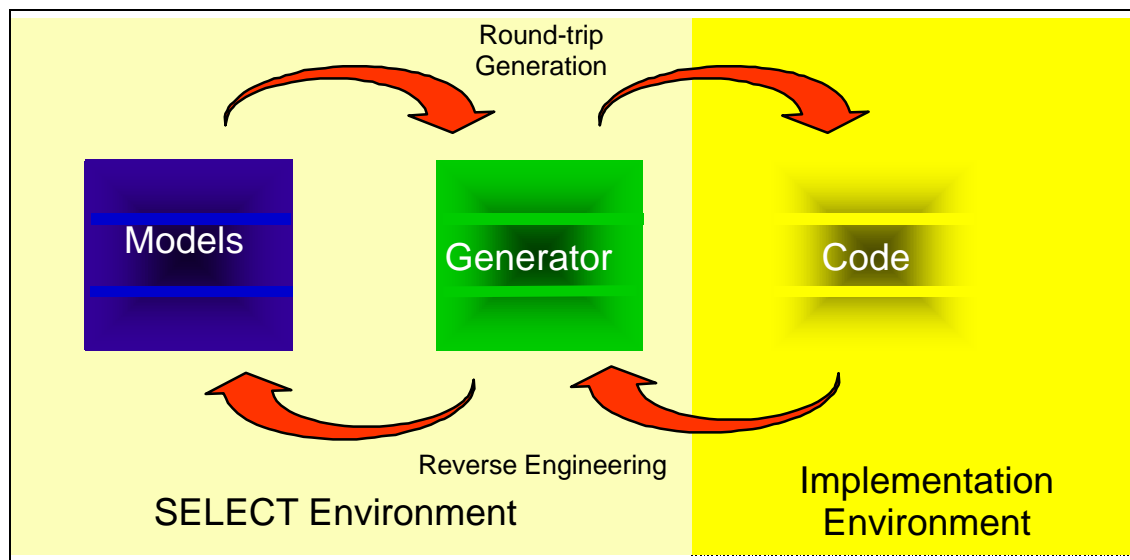
XI. GENERATING OF NEW SOLUTIONS

25. Without automated round-trip engineering, rapid iterative development of multi-tier multi-platform applications would be almost impossible. Maintaining a scalable application

architecture in both the model and the code is essential. The SELECT Enterprise facilitates this mapping with code generators which act as the link between the model and the code. The generator extracts model information from the repository and performs the necessary translation to target implementation language, in our case to Visual Basic. Once the code has been generated, any changes made can be reflected back into the model. Furthermore, when the model is updated and regenerated, hand-crafted changes to the code will not be overwritten. The SELECT Enterprise also offers mapping from an object model to entity-relationship-type storage model. It also offers a generator for Erwin (Logic Works). We have used Erwin for database creating and maintaining. Both tools are bi-directional.

26. Architectural Engineering provides the following benefits:

- Finished applications reflect business need described through models;
- Identification of code re-use through modeling;
- Code and model keep pace
 - Models can reflect changes to code through reverse engineering;
- Business and technology architecture remain synchronuous;
- Modeling identifies aggregation of code to form components;
 - Facilitating greater code re-use through components.



XII. CONCLUSIONS

27. We have briefly presented the structure of the Polish Business Register system, with a three-tier client-server architecture, designed in the Central Statistical Computing Centre. We also explained the general properties of the methods and tools that we are successfully applying in IT system development. Our system is now in the final testing stage and will be fully delivered soon. The methodological background and the technology is complex. Therefore, we need to pay great attention to the education and training process. The object-oriented methodology is promising but the approach is different from the design based on structural techniques. Therefore it requires essential changes in the way of thinking. A strong advantage of The SELECT Perspective approach is that the user has to actively participate in the development process. The majority of the models are understandable to users and allow for their participation in the design process. This approach guarantees that systems will be scalable, maintainable, reusable and understandable.