ARCHITECTURE AND IMPLEMENTATION OF REPORTING MEANS IN ADAPTIVE DYNAMICALLY EXTENDED INFORMATION SYSTEMS

Vyacheslav Lanin

Abstract: This article describes architecture and implementation of subsystem intended for working with queries and reports in adaptive dynamically extended information systems able to dynamically extending. The main features of developed approach are application universality, user orientation and opportunity to integrate with external information systems. Software implementation is based on multilevel metadata approach.

Keywords: CASE-technology, adaptive information system, electronic document, query builder, report generation.

ACM Classification Keywords: D.2 Software Engineering: D.2.2 Design Tools and Techniques – Computer-aided software engineering (CASE); H.2: Database Management: H.2.3 Languages – Report writers; H.3.3 Information Search and Retrieval – Query formulation.

Introduction

Currently term «business intelligence» is understood as tools for data analysis, reports and queries building. These tools aid users to process great volume of data for extracting and producing desired information. Fetching of data processing results (contained in database or data warehouse) is one of the main functions of each information system. Information is entered and collected for further processing (OLTP systems), analysis, forecast and decision support (OLAP and DSS). Information contained in database must be not only processed but also visualized, presented in different document formats according to users informational needs.

Thus, when any information system is developed, tasks of creating means for query building, report design and generation appear. If it is possible, this means should not require user programming skills, vice versa it must be intended for users, who able to work in office program environment. «Business intelligence» means are practically components of most information system; its implementation determines whole system effectiveness.

Requirements to reporting subsystem of CASE-system METAS

Development of information systems, which allow

- capability of dynamical customization according to varying environment and user needs changing,
- functionality extending during system using,

requires special tools.

The CASE-technology METAS (METAdata System) is a base for creating adaptive information systems, which are managed by metadata [1]. This technology is intended for lowering labour-intensiveness of enterprise information systems development and increase its flexibility, scalability, adaptability immediately at runtime. The essential difference between under consideration CASE-system and any other CASE-systems which generate executive code using data domain specification is that our system uses this specification in run-time. In other words, metadata describes data and functions of information system built on METAS technology. It provides flexible application customization, conversion of data objects structure in system maintenance time; this capability may be used as a base of creation intelligent system that would adapts for user needs. METAS technology is intended for development of open distributed application. Therefore reporting subsystem must take into account these technology features [2].

Reporting subsystem must implement following functions:

- creation of queries to database in terms of data domain by user;
- reports generation on bases of prepared by user templates;
- report templates and queries passing between hosts of distributed system;
- integration with external systems.

Consider listed above requirements in detail taking into account METAS technology features and probable using of information system built on this CASE- technology.

A core requirement to reporting subsystem is providing a user interface that aids nonprogrammer users to prepare reports and queries in usual data domain terms. The reports can be quite complicated; it can include additional data processing, analysis and visualization. The system must provide for extension facility (creation of new reports and queries). Furthermore user does not have to use any program language or SQL queries, just the reverse; user must work in well-known environment. Necessary condition of desired requirement implementation is using of advanced metadata. These metadata have already presented in information system built on METAS technology; it describes data domain of a system. So, either data domain and created by user reports and queries can be described in the same terms.

It is necessary system to have means for storage of created queries and report templates (reporting objects); this means must allow to reuse, edit and stored queries and report templates use as a source for new queries. This requirement leads to necessity to include *reporting model* in the metadata structure, this model describes queries and report templates. Means for advanced data exchange between different subsystems are needed. Hence reporting objects and metadata descried it have to be simply portable.

Query building needs writing complex conditions and expressions. Usually so called “expression builder” is applied for this purpose; this component allows to build expression from core construction and ensure syntactical checking of expression. Build-in formula language is demanded for expression builder power increase.

Tools for report item positioning must be provided for user, when he creates new report. Requirement to accuracy of report item positioning may be enough critical, especially when reports are intended for external reporting or automation processing.

Means for integration with external software are needed for document view and edit direct from system. Especially close integration must be organized with e-mail systems as one of the main means of document exchange.

One more requirement is opportunity of using different database system.

Essential features of suggested approach are common engine of document processing from different sources, automation of data exchange processes with different information system, providing capability of texts and documents import from different format files and database. It is necessary to support opportunity of using OLAP и Data Mining for data analysis.

Implementation of Reporting subsystem of CASE-system METAS

As stated above, one of the main requirements to reporting subsystem is capability to create reports and queries by nonprogrammer user. Such requirement can be fulfilled only due to including additional semantic layer; a base of this layer can be metadata already presented in METAS system. Semantic layer allows users to operate with data in terms used in specific data domain, abstracts his mind from physical normalized data table structure. In suggested approach reporting subsystem consists of two components, such as “Query Builder” and “Report Generator”.

A user interface of query builder bases on similar Microsoft Access tool well-known to many skilled users. According to suggested approach a user chooses entities involved into a query and necessary relations between them. After then the user selects needed entity attributes and enters other parameters that affects to result sorting and grouping. As a result taking into account user conditions and metadata interpretation query builder generates SQL-query to information system database automatically.

Report generator is a special tool for report developing. Microsoft Word documents and Excel workbooks may be used as report templates. Also Report generator has a function of reports and queries import/export for exchanging with other hosts of a distributed information system.

According to suggested approach user action order on development new report is following:

1) preparation of necessary queries with “Query Builder”;
2) preparation of document template (including into template static information such as appearance items, formulas, diagram and etc.) and its markup (including into template information about ranges, where query execution result will be inserted when document based on template will be generated);
3) linking of queries and corresponding document range, that will be used for data paste;

4) saving created template in metadata base for hereafter using.

Suggested approach has some important advantages. First of all, programming and writing SQL-query is not needed for development a new report. Second, if analytic facility of query is not enough additional data processing can be performed in Microsoft Excel workbook. Third, report template is a part of metadata because template is kept in metadata base; it allows replicating template with its queries between hosts of information system therefore software update is not needed in contrast to traditional system.

A report generation sequence is following:

1) report template is extracted from metadata base and new documents are created on its base;
2) all queries are run linked with templates;
3) query execution results are inserted into report ranges; if it is necessary additional data processing run;
4) created report is saved into database as a document, it can be printed or passed by network;

Due to the fact that METAS implements electronic document storage in the database created report becomes a part of information system data. Therefore analysis of different time period documents allow creating new combined report.

Approach to development reporting means described above can be implemented in the system which has architecture shown on Figure 1.

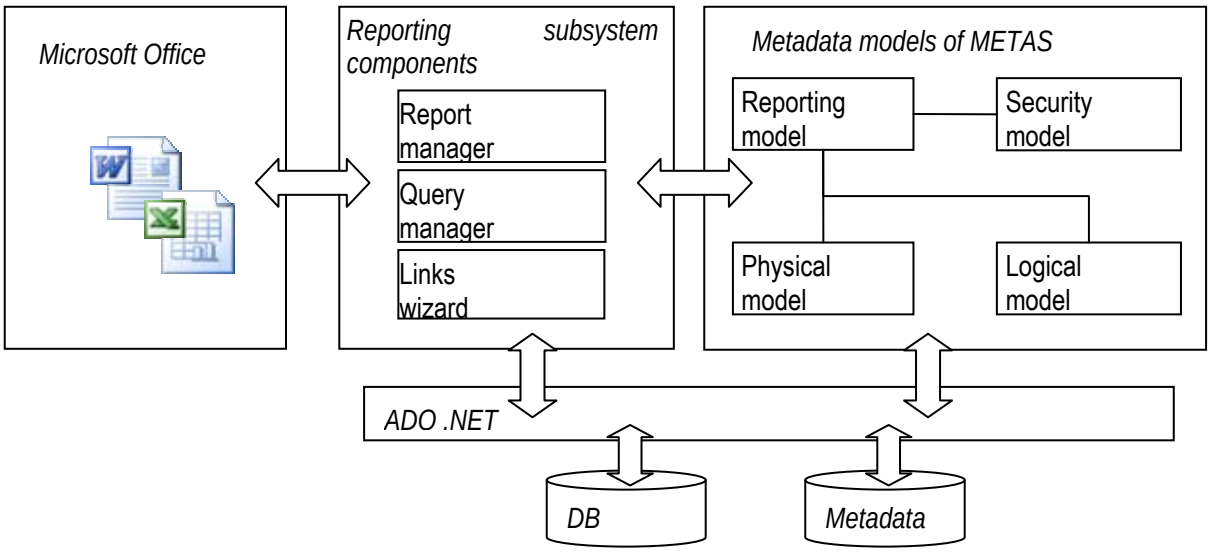

Figure.1. Reporting subsystem architecture

Developed reporting model includes two sub models: query model and report model. This sub models are entirely integrated to common metadata conception of METAS technology, it relies on functionality of physical, logical and security models. Program interface of reporting model is used for:

- execution, addition, deletion and editing query in information system database; this query is formed in terms of logical model of metadata kernel i.e. in terms of entities and its attributes;
- report templates management: generation of report on base of certain template, addition, deletion and editing of report template.

The query model is intended for storage information about queries created by user, editing already existing queries, and its export/import capability, while report model holds information about available reports. A feature of report model is direct storage of report template in metadata.

Program component “Query manager” is used for query handling by user; it is interface for query model. “Report manager” is user interface for report model. Component “Link wizard” is intended for binding report template ranges to queries with aid of graphical interface.

An algorithm of query generation

As described above query created by user is storaged in reporting model of METAS metadata; while data of information system holds in relational database. Hence task of query compilation from object model to SQL appears.

A user forms his query in terms of entities; he expresses constraint and needed fields with aid of expression where logical model terms are used (entities, attributes, etc.). Because of result of generation query process must be SQL query to database of information system, it is obvious name of entities and attributes can not be used. So, compilation from terms of logical model to database tables and fields is necessary. But solution of this task is very complicated; powerful expression analyzer must be created or user must be limited in possibility to build complex expression.

An original solution is suggested, it do not limit user in expression complexity and do not demand labor-consuming programming. Sources of data will not be database tables but preliminarily adduced to 1NF entity view. METAS logical model already has means for generating such views. Because of this approach same view of data sources and expressions are taken.

Taking into account query model and operation features of METAS kernel framework it is necessary to present query to following view.

SELECT query_fields
FROM ((query_for_Entity1) Alias_for_ Entity1
INNER|*LEFT*|*RIGHT JOIN* (query_for_ Entity2) Alias_for_Entity2
    ON join_condition_Entity1_and_Entity2, …
INNER|*LEFT*|*RIGHT JOIN* (query_for_EntityN) Alias_for_EntityN
    ON join_condition_EntityN_and_EntityM),
        join_entity_tables_M:M, (…) Alias_for_Entity(N+1),…,
        (…)Alias_for_Entity(N+n),
WHERE join_entity_conditions_M:M, user_conditions
GROUP BY group_by_expression
ORDER BY order_expression [ASC | DESC]

The key algorithm stages are following.

1. Creating a list of fields which will be output as a result of query execution.
2. Separation of entity set used in query to three subset:
   - linked with aid “left join”, “right join” or “inner join” (*JoinLinked*);
   - linked with relation M:M and not included in first subset (*MultiLinked*);
   - other entities used in query (*NotLinked*).
3. Call to logical model for SQL query for each used entity (logical model generate query which retrieves table entity view in 1NF).
4. Generation of “FROM” query part:
   - merge of queries obtained on step 3 of this algorithm for each entity from set *JoinLinked* with aid of indicated join type (LEFT JOIN, RIGHT JOIN, INNER JOIN), assignment of aliases to query results;
   - enumeration of queries for entities from set *MultiLinked*, for each entity assignments alias and auxiliary tables organized relation M:M;
   - enumeration of queries for entities from set *NotLinked*, assignment of aliases to query results.
5. Generation of “WHERE” query part:
   - addition for each entity in *MultiLinked* relation condition;
   - addition user condition to entity attributes.
6. Generation of “ORDER BY” query part by merge ordering items.
7. Generation of “GROUP BY” query part by merge grouping items.

Described sequence of actions leads to SQL query generation in terms of tables and fields of information system database. This query will be performing directly to database without call to metadata models.

It is obvious generated query is not optimal by its structure, because inconsistency requirements were prompted. However taking into account all modern database systems have internal query optimization means and speed of system response is not critical important suggested approach suits main requirements. It is important to notice source of optimization consist in logical model. Logical model retrieves 1NF as a response of query included all attributes same of them can be not obligatory for query execution. Thus text query size and execution speed can be decreased by addition special functionality to logic model.

Conclusion

Suggested approach to development of reporting subsystem of CASE-system METAS allows to create flexible, user-oriented systems for reports and queries management. Using metadata allows avoiding writing queries on SQL when user needs to create new query. Developed reporting subsystem is an important part of integrated approach to electronic document management in CASE-system METAS.

Bibliography

[1] Лядова Л.Н., Рыжков С.А. CASE-технология METAS // Математика программных систем: Межвуз. сб. науч. трудов / Перм. ун-т. Пермь, 2003. С. 4-18.

[2] Бакланов Д.М., Варламов А.А., Ланин В.В., Лядова Л.Н. Подсистема репортинга программного комплекса MDK METAS // Математика программных систем: Межвуз. сб. научных трудов / Перм. ун-т. Пермь, 2003. С. 19-34.

[3] Ланин В.В. Подсистема управления документами CASE-системы METAS // Математика программных систем: Межвуз. сб. науч. трудов / Перм. ун-т. Пермь, 2006. С. 135-146.

Author's Information

Vyacheslav Lanin – Perm State University, post-graduate student of Computer Science Department; Russia, 614990, Perm city, 15, Bukireva st.; e-mail: lanin@psu.ru

AN APPROACH TO REPRESENTING THE PROCESS OF INFORMATION BUSINESS MODELING

Nadezhda Filipova, Filcho Filipov

***Abstract**: The compact and visualized documenting of information business modeling is a major prerequisite for comprehending its basic concepts, as well as for its effective application and improvement. The documenting of this process is related to its modeling. Thus, the process of information business modeling can be represented by its own tools. Being based on this thesis, the authors suggest an approach to representing the process of information business modeling. A profile for its documenting has been developed for the purpose.*

***Keywords**: business modeling, SPEM, UML profile*

***ACM Classification Keywords**: I.6.5 Model Development - Modeling methodologies*

Introduction

Information business modeling (infBM) aims at visual representation of the business processes of the target organization by means of methods for information system development [Filipova, 2003]. This is a methodology for analysis and reengineering of organizations, as well as for developing adequate integrated computer information systems (CIS). Hence, infBM is a common business process, through which one could represent business processes, performed in organizations, including these for analysis and design, CIS development methodology, etc. This means that infBM is a process for modeling and representing other processes, i.e. this is a metaprocess. This basic feature of infBM emphasizes its importance, and the necessity to understand its components, tools and mechanisms.