A Solution for Document Bases and Relational Databases Integration Problem

Žilvinas Binisevičius
Vytautas Magnus University, Kaunas, Lithuania
zilvinasb@kaunas.omnitel.net

Eugenijus Telešius
Vytautas Magnus University, Kaunas, Lithuania
etelesiu@ssoft.net

Abstract

Every business today is investing in advanced computing architectures and platforms, exploiting robust databases, document bases and operating systems in distributed networks. More and more companies and organizations express their interest in the next generation information systems. The development process of the next generation information system can not start using a 'clean slate approach'. A large portion of the data that is critical to many of an enterprise's business processes is stored and managed by traditional transaction–oriented systems. One of the most formidable challenges that information systems designers and managers face is the need to integrate innovative document and workflow management solutions with the traditional transaction–oriented information systems. In this article we consider the main differences between these two types of environments, the main problems that must be solved in the integration process and describe one of the possible solution.


1. Traditional business management environments

Traditional information systems are designed to achieve the best performance and effectiveness and to be able manage large and very large relational databases. Data input (or renovation) is managed using transaction integrity rules. These rules protect enterprise data from damage and/or unauthorized access and implement the greatest part of business processes. Relational database management systems (RDBMS) provide a common interface which allows joining and combining data elements using integrity rules.

A lot of big companies already have implemented integrated information systems managing all the corporate data, business rules and technological processes. These large complex systems are difficult to use and to modify. They typically have separate systems for budget control, requisitions, logistics, technological process control, etc. The integration of such a heterogeneous environment, having no system approach, is highly complicated.

2. Groupware environments

To adapt the changing business requirements more and more companies and organizations implement next generation information systems [1] (often called document management systems – DMS). The core technology for the next generation information systems is groupware environment. It means that all the business processes in an organization are considered as communications among agents (people or software), not through merely data flows around an organization. Using this approach
the next generation information system project and business process reengineering (BPR) project has to be seen as the multidimensional concept. This is very important, because it means a transition from a set of task–centered workstations to a single process–centered environment based on relations between people in the organization. Another one general characteristic of DMS is that information unit is not a record in database table, but a compound document in document base. This document can contain information objects such as structured information, text, graphics, sound and voice, video or entire other document.

3. Can we use these environments together?

The answer to this question is simple – YES. To achieve the effect as big as possible, we need to integrate these different environments and use them together. Our goal is the integration of different technological solutions under the groupware infrastructure 'umbrella' [2]. In most cases there is a need to use technological process data in documents used by company managing staff. The integration problem arises also when we need to pass data from document management system to relational database management system (back–office).

RDBMS and DMS both are used for information resources management and they are completely different, because they are designed to manage different kinds of information. For example cellular phone operator in Lithuania servers about 10 million calls per month. It is easy to calculate the amount of calls per year and these numbers are still growing. There is no need for explanation that such amount of data can be managed only by special system. We can not do it with document management system.

4. Completely different systems

One of the most difficult problems that must be solved in integration process – how to compare two or more completely different systems and find common principles and interfaces, which enable to develop a new integrated heterogeneous environment. Such different systems are document base management systems and relational database management systems. Table below looks at the main differences between DMS and RDBMS technologies [3; 4]:

<table>
<thead>
<tr>
<th></th>
<th>DMS</th>
<th>RDBMS</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Data Characteristics</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Type</td>
<td>Various data types: rich text, images, other documents.</td>
<td>A set of simple data types: integer, char, float, etc.</td>
</tr>
<tr>
<td>Store</td>
<td>Distributed document base.</td>
<td>Relational database.</td>
</tr>
<tr>
<td>Scope</td>
<td>Distributed document bases within workgroups.</td>
<td>Centralized database.</td>
</tr>
<tr>
<td>Integrity</td>
<td>Data integrity is managed at the client application level.</td>
<td>Data integrity is managed using data integrity rules.</td>
</tr>
<tr>
<td>Structure</td>
<td>Unstructured or semi–structured data.</td>
<td>Structured data.</td>
</tr>
<tr>
<td><strong>Process Characteristics</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Work Unit</td>
<td>DMS</td>
<td>RDBMS</td>
</tr>
<tr>
<td>-----------</td>
<td>-----</td>
<td>-------</td>
</tr>
<tr>
<td>Research area exists which try to incorporate transactional features in document management systems.</td>
<td>YES</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Completeness</th>
<th>The complete set of information is not needed to ensure the full functionality of DMS</th>
<th>A complete set of data is needed to run RDBMS</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Performance</th>
<th>Users interact lightly with the system. Performance is not a key factor.</th>
<th>Performance is a key factor for system acceptance.</th>
</tr>
</thead>
</table>

These differences and absence of integration standards introduces additional problems to system integrators and software developers. They need to investigate the individual cases and find the optimal integration solution, unfortunately this solution often is not the best one.

5. Most popular solutions

The greatest part of today used solutions has one common characteristic: to ensure one feature (for example performance) other features (for example security and comfort) must be lost. Lets take a short look to a few integration solutions.

5.1 Terminal emulation

Using terminal emulation it is required, that all the users of document management system understand the syntax and are familiar with the relational database management system. It led to increasing of the training costs.

Also we need to remember, that information transfer using emulated terminal usually requires some form of manual intervention, which may introduce inefficiencies and inaccuracy. One other problem is that integration processes in this case must be implemented on the client side, what is not acceptable and not cost effective for large and medium organizations.

5.2 Copy data stores

The main point of this approach is creation of temporary data store, which duplicates the real operational data. This data store is filled executing large-scale queries against the relational database. It allows the document management system users to have direct access to data in a familiar environment and did not require additional personnel training. But there are some hidden problems: duplicated data immediately falls out of synchronization with back-end system (RDBMS in our case). So this is acceptable for catalog or general information or decision support systems only. For operational data it is not acceptable, because it may lead to data integrity problems.

5.3 Direct data access

This solution ensures that the information in heterogeneous environment is as new as possible and that all document management system users has direct access to operational data. But this approach
requires significant investments in middle tier software and security mechanisms. The most dangerous problem in this case is – to break data and transaction integrity rules. Direct data access solution may lead to security, system recovery, and synchronization problems. So it break the most of relational database management system requirements.

6. The three tier architecture

Here we describe one possible DMS and RDBMS integration problem solution.

![Diagram of integration](image)

Figure 1. Integration

The heterogeneous environment shown in the figure is divided into three levels:

1. Document management system (DMS) level;
2. Relational database management system (RDBMS) level;
3. Integration server level.

All the data conversion mechanisms are implemented in the integration server (middle level/tier). Integration server is a special independent process working on the same machine as the document management system server or on separate dedicated machine. Dedicated server should be used in the case when integrator meets with very large data amounts. Server is waiting for messages from document management system and relational database management system. After the message arrives, integration server needs to take data from document management system and pass the query (data) to RDBMS. After the query against the relational database is completed, integration server needs to take results (if any) from RDBMS and update DMS document with this data. There is also a forth element in the entire system – control data store. It is a database (in RDBMS) or a document base (in DMS) where the configuration parameters of integrator are stored. Here also are stored the data conversion instructions for integration server.

6.1 Work scenario

Here we describe one possible application scenario in order to illustrate the work of such integrated heterogeneous environment.

Let’s say we have a fictitious marketer of mobile phones with a head quarter in Kaunas and branches in Vilnius and Klaipeda. In headquarter the SQL compatible relational database management system is installed, which stores and manages all the structured operational data. All organizational units of the company are using groupware environment for document management applications (for example Lotus Domino).

The sales representative in the company prepares an ordering document containing the ordered mobile phone and customer information. This document is automatically placed into document base (in Lotus Domino). The document base is all the time monitored by special DMS process. The new document in document base has changed the state of this base. This event fires the trigger that sends the
message to integration server. The message wakes up integrator. According the information stored in the message integration server establishes connection with DMS and RDBMS (if it was not established before).

After the connection is already initiated, integration server reads the information stored in a new ordering document. Then it also reads a set of data manipulation instructions from control data store. The instructions describe actions to be performed with data stored in DMS document. Then according these instructions integrator performs necessary type conversion and/or modification routines on DMS data and passes it to RDBMS server. When all these operations are done, integrator may close connection to RDBMS and DMS and go to sleep.

RDBMS gets a query with data from integrator and executes it against the database. All the data modifications and transformations are made using the business logic stored in relational database. When all the necessary modifications are made, RDBMS sends a message to integration server and informs it that queries processing are done and data may be accessed and transferred back to DMS.

After this message from RDBMS is received, integration server takes data from RDBMS, performs necessary type modifications and transfers it back to DMS. This finishes the data transfer process.

And now let's imagine the same organization with no integrated document management and online database management environment. The sales representative, in this case, files the new ordering document and places it into document management system. So now all the necessary ordering information exists in DMS and all the authorized DMS users are able to access it. But what about our back–office system, which manages all the critical enterprise information? There are no data transition mechanisms between these completely different environments. In this case we should have an employee which does nothing, just manually takes data from document management system and enters it to back–office (RDBMS) system. Of course, this ‘solution’ leads to all the problems we have mentioned before in this article.

6.2 Goals of successful integration

Summarizing our findings, we generalize the key factors for successful integration:

- Unified object model;
  If we are using unified object model [6], we are concerned with a system, that is logically compatible and stable, not a casual set of application software. Users need only to learn one object model (API) and use it. Many applications from different vendors use the same software components.

- Application development using programming languages;
  Each strategy should allow to develop integration applications using one or more programming languages. That is not important what a powerful integration mechanism exists, there is situation where additional programming is necessary.

- RAPID application development;

- Language independent integration;
  Software developer should be able to use more than one programming language (C, C++, Java etc.) Application programming interface (API) should be designed and developed with steady interface for all programming languages.

- Possibility to integrate completely different (incomparable) environments;

- Enterprise level integration.
  There should be a possibility to do enterprise level integration. The integrated environment should be able to serve the large amount of clients in a large organization. It
also includes multiplatform support, multiprocessing, remote administration, multithreading and support for national character sets.

6.3 Advantages and disadvantages of our solution

The main advantage of this approach is the possibility to implement it in a corporate system. Such realization allows to perform exchange of large data amounts. Also very important are other aspects of this integration model:

1. This solution allows to access RDBMS data not using ODBC. Integration server may use native interfaces to connect to RDBMS and access data with significantly increased speed;

2. System is platform independent. The server may run on one operating system (for example digital UNIX), and client part may work on completely different operating system (for example Windows NT or Macintosh). Also separated parts of heterogeneous system may be platform independent if we have implemented it using object oriented technology.

3. Systems communicate using special protocol, it is possibility to develop our protocol or to use an existing one. For example we may take ‘Open Server’ architecture TDS protocol. It describes data exchange between RDBMS client and server and allows expanding the protocol depending on user needs [6].

4. It is possible for one integration server to serve several document management system servers, working with many RDBMS servers.

5. We can place all the parts of the system on separate (dedicated) servers and implement the duplication of functions, in order to increase data security and reliability.

7. References


