IMPLEMENTATION OF CORPORATE INFORMATION TECHNOLOGIES USING LOTUS NOTES/DOMINO

S.G. Tolouzakov

Computer Science Department
Tomsk State University,
Lenina av., 36, Tomsk,
Russian Federation, 634025
phone: (3822)426096, 426097
e-mail: chief@csd.tsu.ru

Lotus Notes/Domino is a well-wide groupware product and a standard de facto of document flow, e-mail and collaborate products. It has a lot of advantages making it suitable to implement a corporate information system.

There is a strategy made, containing three basic aspects: description of workflow and document flow within an organization, methods of involving Lotus Notes/Domino means and features to make an open, scalable and effective system, and a sequence of steps to make to build a basic structure of an information technology, providing the two previous aspects are taken into consideration.

1. Originally, description of an existing workflow can be done, using common language: just a set of statements, describing some facts of real document flow and processes with the organization. Further, it can be unified, using a specific language (specific terms, diagrams, etc.).

There is a family of methodologies that help a developer of an information technology build a comprehensive description of business processes to be automated. One of them is IDEF3 (Integration DEFinition, Workflow diagramming) that provides an analyst with terms and symbolic signs to describe a process-oriented model.

The first aspect is a process-oriented (or subject-oriented) description, that is based on some lingual form written. A process-oriented description begins from the upper level of decomposition, where we have the only process, that can be associated with the organization's mission (in a functional context). This process is divided (or decomposed) into several (3 to 7) subprocesses, etc. At the last level of decomposition we get the "atoms" (or the processes, that can't be further divided), that are the database elements to be designed.

The second view is an object-oriented view. It helps an application designer to find out, what kinds (or types, etc.) of documents should fulfill the future storage. Documents can be created on other document's (or document collection's) basis. This makes us create a "document deriving" structure, showing how the documents appear, modify and get off the storage.

The third fact, that's going to be taken in consideration, is the fact, that documents appear, modify and are removed from the storage due to some certain events. This makes us to build a temporal model (or event-oriented model), that shows the consequences of event happening and the document changing. Here we can have a "document life cycle" diagram, that has two dimensions: event happening and document changing.

2. To use the means of Lotus Notes/Domino one should know the features, the effective functions, mechanisms and, of course, disadvantages, conditions that bring a complexity and inefficient results to the application and, therefore, to the end users.

Lotus Notes has a wonderful feature of manipulating of big numbers of documents - the full-text search. This feature should be accounted, when an application is being developed. Search within views becomes faster, when a database is fulltext indexed.

The great feature of replication (and clustering) can be used for distributed storages, when a corporation has several offices around the world. The management in this case is decentralized, but the information access is centralized, each replica in each department has the full information (got during certain time, of course). In order to use this feature, a database designer should change the design, that is important.

Nonetheless, the means of Lotus Notes have disadvantages: documents with changing links are not supported by the system, a programmer should find the means of creating a relational Lotus Notes design and to support the link changing;

there is no support of distributed on-line document work, what brings a developer of a corporate system a serious trouble;
there is no transaction tracking system that can lock documents and fields, being edited, from other users, what makes a developer write some special programs, containing the strategy of getting rid of document on-line save conflicts.

there is not a "pure" client-server architecture, allowing only to find, read (or write) a document in a database, but complex calculations over big numbers of documents are not available.

These disadvantages can be removed with the help of some tricks, implemented by non-standard programming. For example, to keep links actual, we need to design special view that would "see" the changed documents, then to make agents that would change the related documents.

To make a "pure" client-server, we make a special "agent runner", that keeps all the agent within all databases intact and allows a synchronous run of a client and a server application, moving all the complex calculations to the server, and sending the output to the client.

3. Learning the description of busyness processes within the organization and basic concepts of Lotus Notes/Domino system guides one to the third step - building the system itself. It can be accomplished through several steps (or stages).

1) Define, what kind of dictionaries (or classifiers) should reside in a system. The number of records in each classifier should be considered as well, because if the classifiers are big enough and related with each other, they shouldn't be implemented as a Notes storage. Relational DBMS is more suitable and efficient. The best way to use big classifiers is to implement them in a RDBMS and to get access to them from Lotus Notes applications using existing interfaces (ODBC, NotesPump, etc.).

2) Design the dictionaries and define the means of access to them. It can be a standard Lotus Notes means (if a classifier resides within a Lotus Notes storage), or some other means (interface ODBC, NotesPump, etc.).

3) Design the forms according to the description of the busyness process. These forms should include all the basic requisites of the document, as well as they should include some assistant fields, such as document status and the route the document should pass through.

4) Design the views, needed for the end users, as well as some additional hidden views, used for effectiveness of the application. The views used by the end users should be designed according to the object-oriented model built on previous stages during the process description.

5) Design navigators, that are necessary for each automated workplace. They should include links to the basic views, designed on the previous stage. They should include some useful actions, needed to the automated workplaces, as well.

6) (Optional) Create the additional navigators and views, containing graphics, add (or modify) forms of the application to make an Intranet application. This stage can play significant role, when an application is going to be "rolled out" into the Internet.

Head of research - Assistant Professor, D.R., S.P. Souschenko

Serge Tolouzakov was born in Alma-Ata, Kazakhstan, on August 31, 1975. He graduated from secondary school with advanced English learning in 1991, then entered Tomsk State University in Russia. His employment experience included Tomsk State University, Tomsk City municipal administration, Tomsk City Department of Finance. His diploma work was implemented and devoted to electronic document flow and groupware within Tomsk City administration and Tomsk City Department of Finance.