

MDFS 2006 - preliminary info

The MDFS stands for Multi-Dimensional File-System, which is not really a files-system, but a data storage engine with variety of functions defined in it. The closest match to MDFS in today technology would be the combination of native XML databases and object oriented databases.

Storage Engine

The MDFS is in the first place intended for data storage. The system is built to store highly structured data like XML, but there is also a possibility to store other kind of data, in legacy systems is this form known as BLOB (binary large object). In MDFS the structured data is stored directly, and unstructured - binary data could be stored in one of the dimensions.

Dimensions

The cool thing about the MDFS is, that you store there your (structured) data in the form of XML and you can retrieve it not only as XML, but also in another forms. Each dimension has an identifier and a type, where the type information is compatible with the MIME standard. Example usage of dimensions can be for generating source code (C, Pascal, JavaScript, PHP and many more) and for visualization purposes (ER and class diagrams).

Relational data

The first version of MDFS had a poor support for data linking, so this new specification defines not one, but directly two types of data linking - one in system level and one application level. The application level linking is an old method, where the application is responsible for linking the proper data and it requires two types of nodes - one is the object and one is the reference.

The better method could be the system level linking, which uses only one types of data - the object, and for every node it defines if it is a link or not. In the case that the node is a link, only those data have to be present, which are the part of the primary key. The system then could find the proper object node, with full data, and the application does not notice the difference.

Indexing service

From the experience with the first implementation of the MDFS it is clear, that the performance of this system was not so good. The solution offered to speed up node lookups is indexing. The index is created over a set of nodes which have are of the same type (this means they have same name and namespace). There is expected a little slowdown, for managing the index when the nodes are created, modified or deleted, but the speedup of node lookup will be much faster.

Triggers

Triggers introduced in the new version of the MDFS should bring easier implementation of the business logic tier on the top of the data tier. There are four types of triggers - on creating, deleting, updating and even on reading of data. The reading trigger can act for example as an access counter. Each of these triggers is separated to two parts - whereas the first one can deny the whole operation and the second one is run after the operation is completed. The triggers are node type related (as indexes).

Stored procedures

While the triggers are rather part of the system than the higher level application, the stored procedures can replace the most of the business logic. When we imagine a stored procedure in the object database, we get methods - which in MDFS can be both static (class related) and non-static (object related).

Access control

The last, but not less important feature of the new MDFS is that we had improved it in the question of security. Now on, the user of the MDFS can set hierarchical privileges in the structured data which says who can do what (operations are reading, deleting, changing, creating sub nodes, creating nodes).