

## ABSTRACT

*The experience with authoring multimedia material for educational purposes shows a major problem: how to provide an easy and efficient way to handle multimedia objects in a manner that non-expert users (namely school teachers) can be able to design and build their own presentations? The creation of this presentations involves factors like storage, delivery, search and presentation of multimedia material (video in special). A basic infra-structure that stores and efficiently deliver the video data is needed. However, another important point is the organization of these data stored into the server in a way to facilitate the access to them from the users. In the system wich is the subject of this work, this is achived through the use of an interactive information management and retrieval system designed to facilitate the access to items (or parts of the items) stored in the server. The main characteristic of the system is the use of a metadata base which contains attributes of the videos stored in the server. Searches can be made by title, subject, length, author, content or, most important in the didatic multimedia material case, by a specific scene or frame. The system was built with JAVA programming language in a client/server way. The communication between clients and servers is realized through the use of the Visibroker 3.0, which is a Distributed Objects programming tool according to the CORBA standard. The data access from the metadata base use a PostgreSQL driver which follows the JDBC API. For evaluation purposes a playback tool was built using Java Media Framework (JMF). An analisys was carried out to verify the impact of the utilization of CORBA and JDBC technologies in the system. It was detected that JDBC technology utilization imposes a much more significate delay than the CORBA technology utilization. Another conclusion is that metadata utilization provide better interactivity searches, making the editing process faster and save storage space through the sharing of objects like videos, scenes and frames.*