Preservation by Migration to XML

Dirk Roorda; Data Archiving and Networked Services (DANS); Anna van Saksenlaan 51, 2593 Den Haag, Netherlands; **Dirk.Roorda@dans.knaw.nl**

There are three key approaches to preserving digital information against software obsolescence: emulation, migration, doing nothing.

Emulation preserves the methods to access the archived data, so that the archived data can remain untouched. These access methods need to be emulated on ever newer platforms over time. Migration converts the archived data, so that newer applications can be used to access it. These conversions have to be adapted and repeated over time. Doing nothing leaves everything as it is, and puts the burden of future accessing data to ... the future.

All methods have distinctive use cases where they are to be preferred, and others where they are to be deprecated. Moreover, emulation and migration can be done straightforwardly or intelligently, and in the latter case the effort of these methods can be greatly reduced. Emulation is handy where you have to preserve materials that have an important, unique look and feel that is part of the worth of the data: art expressions, multimedia materials, everything that performs. Migration is handy where the meaning of the data is not dependent on the precise form, and where you want to aggregate data in new ways: this holds for research data.

The focus of this paper is a project, carried out at DANS, Netherlands, to implement an economical migration strategy for research data. The name of the project is Migration to Intermediate Format for Electronic Data (MIXED). It is a two-year project with a 2M€ budget.

MIXED is meant to handle tabular data, that is data that comes in the tables of databases and spreadsheets. This data is created by popular office applications or database systems during research projects, and when it comes to archiving, they would quickly become hard to access and to aggregate because of the ever changing vendor formats in which the data is packed.

By defining a generic XML format for tabular data, called M-XML, and converting such data to M-XML upon ingest, and converting M-XML back to desired vendor specific formats upon dissemination, we actually implement a very efficient migration strategy. Instead of carrying out numerous different migrations on many different formats, we factorize the complexity of preservation into a synchronic dimension and a diachronic dimension. Synchronically, we define and implement the conversions between vendor formats and M-XML. Diachronically we implement the conversions between successive versions of M-XML.

The advantage is that the synchronic conversions do not have the difficulty of bridging time, and the diachronic conversions do not have the complexity of working with many vendor formats.

Last but not least: the archived material is now in a kind of normal form, M-XML, and can be aggregated irrespective of its original vendor format.