

Requirements for Multimedia Authoring Tools: Workshop Paper

CIDOC – Multimedia Working Group – June 2001 (revised December 2002)

Authors:

Jan Van der Starre –Cap Gemini

Ann Borda – Science Museum

The Multimedia Working Group (MMWG) of the Commission on Documentation (CIDOC) of the International Council of Museums (ICOM) met on 3 July 2001 in Barcelona for a workshop on requirements of multimedia authoring tools. In that meeting a discussion took place on the various aspects of multimedia authoring tools. The results of this workshop are reported here. It is the intention of the MMWG to use this as a starting point for a requirements list.

1. User Community

The community of museum developers can be divided into several groups according to the matrix:

	Large Museum	Small Museum
Large Project		
Small Project		

This community should have every opportunity to influence the functionalities of authoring tools, in close cooperation with the producer of the tool. This is the more so important when tools have a high degree of fixed structure (see underneath).

2. Structure

2.1 Structure and complexity

Following this matrix, both structure and complexity in general can be represented as follows:

Fixed Structure	Large Museum	Small Museum
Large Project	--	-/+
Small Project	-	+

The degree of fixed structure in an authoring tool should be dependent on the position in the matrix (- and + indicates degree of fixation). Of course there may exist exception from this 'rule', in that large museums may build small applications with fixed structure.

The same matrix could be used to indicate the degree of allowed complexity in a Multimedia application:

Allowed Complexity	Large Museum	Small Museum
Large Project	++	-/+
Small Project	+	-

Complexity can be expressed in linear, hierarchical and networked (hypertext), with all mixed forms allowed.

Ideally an application development tool or authoring tool should allow for all levels of structure and complexity. However in practice this can never be achieved and only a few are possible.

Authoring tools therefore should be positioned clearly in one or two areas in the matrix. The development of applications with no fixed structure (fully networked) and a high degree of complexity should probably better not be done by the museum itself, but should be given over to a professional developer.

2.2 Structure and metaphors

Structure, complexity and metaphors should be content driven, i.e. the choice to be made is dependent on the content and structure of the message which has to be conveyed. Metaphors are also important because of the differences in culture.

2.3 Structure and application building

Structure gives guidance to application builders, i.e. it sets clear limits to what is possible with a particular authoring tool, at the same time allowing museum staff involved to express whatever (form of) message needed. Curators are thus limited but not constrained in telling the museum story. Seemingly, this is contradictory with the use of metaphors, as described in the previous paragraph. However, ideally an authoring tool allows for the use of a multitude of metaphors, which is possible to achieve, even within a fixed structure.

2.4 Structure, complexity, metaphors and the museum visitor

It is assumed that, given enough variations in the use of metaphors, museum visitors will benefit from a standardisation of the functionalities of interfaces. They do not have to learn to use each application individually, but may use their knowledge of a particular application to access another application. This reduces the learning curve visitors have to go through, will allow them to enjoy more the content of applications, and will give them the feeling to be in control.

3. Scaling

An authoring tool should allow for:

- Scaling (growth of an application over time)
- Portability (transferring to another platform, e.g. other languages, other operating systems and hardware (Mac, Windows, Linux))

3.1 Local language versions

Local language versions of authoring tools are needed, but the mechanism to control these versions should be in the hands of the supplier. Local users may have at their disposal translation tools, but finalisation is in the hands of the supplier.

3.2 Desktop and Web development

Ideally an authoring tool should allow for development both for in-house applications (exhibition area) and stand-alone applications outside the museum (kiosk type applications). Furthermore, applications should also be transferable to other environments, such as the Web, either as products developed for the Web, or as products ported to the Web.

4. Exchange platform

An exchange mechanism is needed both between the supplier of the tool and the museum community and between museums among themselves. The first is the responsibility of the supplier and the second of the museums. However, in the latter case the supplier could very well play a facilitating role, and may even should do that. Such a platform ideally should take the form of a web application in order to be accessible for the community at large.

4.1 Supplier platform

The supplier platform should provide facilities for:

- Updates of software
- Distribution of local language versions
- Distribution of user guides and other supportive material
- Feedback or user forum
- Evaluation

4.2 Museum platform

The museum platform should allow for:

- Exchange of whole multimedia presentations
- Exchange of components of presentations
- Exchange of the cultural assets contained within presentations
- Cooperative developments
- Evaluation

5. Cost and functionality

An authoring tool should show a balanced trade-off between cost and functionality. In the best tradition of trade this should mean as much functionality for as low cost as possible as most museums lack economic power. Museums however which wish to build highly sophisticated and multi-functional applications should be aware that they have to pay a price: either have a professional developer build the system, or build it themselves while having to go through a steep learning curve and having to solve many technical problems.

6. Standards and formats

Authoring tools for museums should stick to accepted and proven standards in all respects. It should not be leading edge technology, and exchange and reuse of material is only possible when standards are in place and adhered to. In the case of reuse XML awareness is of prime importance.

7. Media types

Depending on the required content and complexity of an application, all media types should be allowed. This includes:

- Still and moving images
- Sound effects
- Music
- 3-D type images
- Database records and access to databases
- Embedded documents (e.g. PDF)

For special applications even a text-to-speech translation tool can be necessary.

8. Specific requirements of authoring tools

In addition to the more general and functional requirements listed in the preceding paragraphs, a few more specific ones should be considered:

- The tool should allow the author to provide equivalent or alternative information (e.g., provide text equivalent for video or image content).
- Ensure that functionality related to 'best practice' authoring is naturally integrated into the overall look and feel of the tool.
- Promote accessibility and best practice authoring in help and documentation features of the tool.
- Allow the author to edit all properties of each element and object in an editing window without altering the underlying programming language; unless, of course, a committed 'save' is made by the author.
- Ensure that the editing view allows navigation via the structure of the document in an accessible way
- Ensure that the tool preserves all accessible content during authoring, transformations, and conversions, i.e. a basic back-up functionality, version control and undo options.

- Provide basic functionality for managing, editing, and reusing multimedia objects similar or better than standard Windows functionalities, without affecting the document markup or language.
- Ensure that the authority tool has built-in functionality that allows for the user/author to generate a multimedia product which is stable and supported by standardised language for delivery and reuse on several platform types. Therefore the tool must be able to support (and correct) error handling, incorrect sequencing, etc. without the user/author relying on a knowledge of programming language. Integrated help messaging/documentation is an essential part.

