Digital content authoring based on an open architecture approach

Bernard Lange, Cezary Mazurek
Poznań Supercomputing and Networking Center

Key words: authoring, authoring tools, digital content, open-architecture

Abstract:

Versatile authoring tools will help authors cope with the increasing number of standards and proprietary solutions. Such tools will be able to cooperate with numerous systems in order to exchange and reuse content. In this paper we present an open-architecture approach which could be followed during the design of the authoring systems, which, we believe, greatly facilitates the creation of adaptable and flexible solutions. As part of the research, a pilot system Contentum exploiting this approach was developed in Poznań Supercomputing and Networking Center. The tool was successfully integrated with the existing digital library solution. The possibilities of other usages are being investigated.

1 Introduction

The need for versatile and effective authoring tools becomes more crucial, as diversity in standards increases or new interoperable standards emerge. Vendor-specific systems are required to efficiently exchange, share and reuse content; however, the character of such cooperation is often unknown at the development stage. Thus, when it comes to the point where different tools are to exchange information, the existing solutions are unable to cooperate or cope with compatibility problems. Then, such a cooperation becomes real challenge requiring a significant amount of organisational and developmental effort.

Similarly, as the effort, complexity and the cost of implementation and deployment of one particular content management solution in the corporate or academic environments is considerably huge, it is unlikely for the tool to be changed and for the whole process to be repeated every time when new and better standards become known. Therefore, the tool's specialization for particular standards and media types becomes an obstacle, and more workaround solutions must be designed and implemented. Consequently, the more adaptable and open the tool, the lower the cost of such an introduction. In this paper we present a possible remedy to the problem, an open architecture-oriented approach which can be followed during the design and development of authoring tools. As a part of the research a pilot system which exploits the proposed architecture was developed in Poznań Supercomputing and Networking Center and integrated with the dLibra digital library framework [1], [2].

1 Open-architecture approach in authoring

Solutions characterized by high adaptability and versatility, are more likely to live on in the ever-changing environment. We tend to believe that exploitation of the open architecture design might help authoring tools to achieve such an adaptability and versatility. In fact, open-architecture design has proved to be successful in the software development world where e.g. the Eclipse platform [3] gives boundless possibilities for plugin creators to augment IDE with support for practically any programming language. The complexity and the cost of the creation of one plugin are far lower than the complexity and the cost of a design and implementation of the whole IDE.
In this context, by *open architecture systems* we understand systems characterized by four main features:

- framework-based core
- pluggable standards
- easy integration and communication with other systems
- platform independence

Consequently, the proposed open-architecture comprises of 3 layers (Figure 1, “Open architecture authoring tool”):

- cooperation layer
- frameworked core layer
- content servicing layer

At the bottom of the system lies the cooperation layer providing slots for different types of services responsible for content exchange, storage, searching, publishing. In general, adaptors available on this layer are responsible for interactions with external systems. As Figure 1, “Open architecture authoring tool” shows such interaction may involve e.g. different storage mechanisms (FTP, DBMS, WebDAV, CMS etc.). It may exploit implementation of particular protocols e.g. IMS DRI [4] making the system able to talk with other DRI-enabled systems. This layer assures that the system is able to cooperate with other standard-aware or proprietary systems. This obviously requires relevant adaptors to be created, but, the effort required to create such an adaptor is insignificant when compared with long-term benefits. Let us take a system which currently stores the created content on a company's WebDAV server. Assume that the company bought and introduced a new CMS system which is to be used as a main repository for the content from now. Assume that the CMS is DRI-enabled and is able to expose its assets to DRI-compliant clients. If the authoring tool the company currently uses can be augmented with the DRI adaptor, the integration is almost seamless and practically unnoticeable for the users. Still, they use the same tool to create their content, only the way it is stored changes. In contrast, if the tool is constrained and bound to the WebDAV support, the company must obtain a tool which will be able to interact with the CMS system. This increases costs and lengthens the deployment phase and forces users to learn how to use the new tool.

The frameworked core provides essential services like document management, internal representation, coordination and communication between modules. It acts as a basement in which the additional components can be plugged in and bridges the cooperation layer with the content adaptors. For example, if the user orders the system to store the created content, the core will present the possible ways to accomplish this task, basing on its knowledge about services registered in the cooperation layer. The core is also responsible for the provision of generic content management, validation and visualisation functions.

The top level is associated with particular content servicing. This encompasses issues like content styling, dedicated visualization, edition (special actions), in general the whole knowledge about a particular type of content whose core may be unaware of. As a type of content we understand different standards used to describe content, e.g. adapter for Docbook with sophisticated rich text styling and WYSIWYG edition can be provided while the adaptor for SMIL or NewsML, can provide completely different visualisation and edition environment. Such specialized adapters can be added any time, allowing inclusion of support for new content standards as they emerge.
The open architecture, the component-based approach results in high adaptability and flexibility of the overall system, allowing greater overall scalability and interoperability of the created authoring application. Authoring tools can benefit from such a development course. Once a framework is created, it can be augmented with satellite modules which can be used to serve different kinds of content. Configurable modules can be grouped to establish different environments reflecting different users. Owing to this modular structure, the tool can be tailored for the needs of any authoring process, according to the currently established company-wide authoring rules. Consequently, adaptation of such a system comes down to the addition of new modules. In fact, such modules can greatly increase the ability of the system to cooperate with other systems, making inter-system communication or integration with the existing solutions feasible.

2 Pilot system

In order to observe and assess the usability of such an approach and identify possible difficulties during development, an exemplary pilot system the authoring tool called Contentum was implemented as a part of research on digital information management techniques carried out in Poznań Supercomputing and Networking Center.

The main development goal was to make the system as versatile and flexible as possible. Thus, the open architecture allowing inclusion of new modules was exploited. As a result the Contentum accepts any XML based content description standard. In fact, the majority of the standards in the areas of e-Learning, digital publishing, and content management is based on XML. The system encompasses support for popular standards (like e.g. DocBook, SMIL etc.), yet seamless inclusion of standards that will appear in the future is possible.

Therefore, the tool is not bound or constrained only to one particular content description schema. In fact, support for different standards could be plugged as specialized modules. Such modules can take responsibility for content visualisation, styling, edition, leaving the storage, and internal presentation issues to the core. The core itself is also responsible for the provision of the environment, standard workspace within which the content can be authored. In addition to that, the system encloses off-the-shelf support for any content which internal structure is described in XML. (Figure 2, “Contentum off-the-shelf visualization”) This encompasses support for such assets as e.g. text, media.
However constrained, the system can be aided with components which can be used to serve different kinds of assets embedded in the content. For example, the application cannot handle Macromedia Flash objects, but this can be changed if a proper handler is provided. Such handlers can be bundled with adapters for new standards or can be added separately. Then, adapters can be combined to create dedicated sets that adhere to the needs of e.g. a school or publishing company. This results in a highly configurable environment.

In the next part of our research, we decided to make the Contentum system able to publish content into a digital library created with the dLibra digital library framework \cite{1}\cite{2}. To achieve this, the storage adaptor was created. It contacted the library and passed the content in the digital-library manner. The experiment succeeded and the Contentum system was successfully integrated with the dLibra digital library framework. Consequently, owing to the component-oriented architecture approach, integration in the area of these two completely independent information systems may be possible. The benefit will be much greater if the communication between two systems is done by means of a standardized protocol. Then, the adaptor is reusable and can be used to cooperate with another systems. The experiment continues and the possibilities of collaboration with the tools used for the preparation of content for interactive television are currently being investigated.

3 Conclusion

The future of authoring tools lies in their flexibility and high adaptability. We tend to believe that the open-architecture approach can help to achieve this goal. The implementation of the pilot system confirmed our beliefs, though, it simultaneously revealed that the development can cope with several problems.

Firstly, the design of a good core which would serve well in collaboration between content and cooperation layer is crucial. The set of messages that can be sent across layers should be designed in a rather general way, defined in terms of functionalities like e.g. repository access, permission and access management and so on. The rules of modules inclusion and definitions of module sockets should be well defined. This can be facilitated by means of standards, e.g OKI OSID's (Open Service Interface Definitions).
Next, the other standards must be in place and used. Authors of the authoring tools should promote the existing content description standards (e.g. MathML, SMIL etc.) along with content sharing and exchange standards (including the static content description (e.g. SCORM[5]) and dynamic communication protocols (e.g. IMS DRI)). Once the standards are supported and promoted, the interoperability of the tool increases.

In terms of initial effort, complexity and cost associated with the development of one open-architecture and proprietary authoring tool, we can assume that they are similar. However, in a long-term perspective, the open-architecture-based systems are more likely to scale effortlessly.

Finally, the proposed approach can result in universal systems, that can be suited and configured to the needs of a wide variety of users. This adaptability combined with the longevity of the tool itself can dynamize the content creation process, making it easier and more accessible.

3.1 Related work

In the area of open eLearning systems a great deal of work has been done by the IMS Global Learning Consortium [6] with the definition of new standards for data exchange and cooperation among LMS systems. IMS has defined DRI[4], whose purpose is to provide "recommendations for the interoperation of the most common (digital) repository functions. These recommendations should be implementable across services to enable them to present a common interface" (IMS site). This work is complemented by OKI (The Open Knowledge Initiative) [7], fostered by MIT, which has produced a series of Open Service Interface Definitions (OSID's) - architectural elements determining the abstract layer between the programmer and the enterprise infrastructure systems. OKI aims at simplifying and enhancing the development of the educational applications.

References:


Author(s):

Bernard, Lange, MSc
Poznan Supercomputing and Networking Center, Network Services Department
61-704 Poznań, ul. Zwierzyniecka 20
Bernard.Lange@man.poznan.pl

Cezary, Mazurek, MSc
Poznan Supercomputing and Networking Center, Network Services Department
61-704 Poznań, ul. Noskowskiego 10
Cezary.Mazurek@man.poznan.pl