HAL
open science

Multimedia Authoring for Communities of Teachers

Agnès Guerraz, Cécile Roisin, Jan Mikác, Romain Deltour

To cite this version:

Agnès Guerraz, Cécile Roisin, Jan Mikác, Romain Deltour. Multimedia Authoring for Communities of Teachers. International Journal of Web-Based Learning and Teaching Technologies, 2007, 2 (3), pp.1-18. inria-00189319

HAL Id: inria-00189319

https://inria.hal.science/inria-00189319

Submitted on 20 Nov 2007

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Multimedia Authoring for Communities of Teachers

Agnès Guerraz and Cécile Roisin and Jan Mikáč and Romain Deltour
INRIA Rhône-Alpes
655 avenue de l'Europe
38334 Saint Ismier, France
{Agnes.Guerraz,Cecile.Roisin,Jan.Mikac}@inria.fr

One way of providing technological support for communities of teachers is to help participants to produce, structure and share information. As this information becomes more and more multimedia in nature, the challenge is to build multimedia authoring and publishing tools that meets requirements of the community. In this paper we analyze these requirements and propose a multimedia authoring model and a generic platform on which specific community-oriented authoring tools can be realized. The main idea is to provide template-based authoring tools while keeping rich composition capabilities and smooth adaptability. It is based on a component-oriented approach integrating homogeneously logical, time and spatial structures. Templates are defined as constraints on these structures.

Introduction

We are involved in a multidisciplinary project the aim of which is to support the activities of Communities of Practice (CoP) in pedagogical environment. This project will provide tools for document production and for document reuse in heterogeneous applications. The objective is to reduce the current limitations caused by the proliferation of data sources deploying a variety of modalities, information models and encoding syntaxes. This will enhance applicability and performances of document technologies within pedagogically consistent scenarios. In this paper, we will focus on the authoring needs of teacher communities and propose a new authoring model, LimSee3.

In the educational context, there exists a large variety of authoring tools, see (Brusilovsky, 2003) for an extensive review. The main objective of these systems is to provide adaptive educational hypermedia thanks to well-structured hyperlinked content elements that are mostly static content. In (Hoffmann & Herczeg, 2006) the created documents are video-centric, providing a way to add timed hot-spot embedding additional media and interaction facilities in the resulting hypervideo. The time structure is therefore straightforwardly given by the video media, while the time model of our approach (given by the SMIL time model) is much more general. In our project, we want to provide educators with a way to take advantage of multimedia synchronization to offer more lively pedagogical material. But it is worth noting that multimedia brings a higher order of complexity for authors. In order to reduce this complexity, we propose a multimedia authoring model that will provide similar authoring services than formed-based hypermedia systems (Grigoriadou & Papanikolaou, 2006).

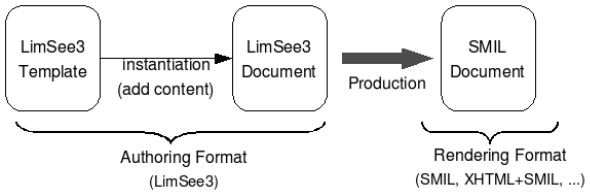

Figure 1. The authoring process in LimSee3

The LimSee3 project aims at defining a document model dedicated to adaptive and evolutive multimedia authoring tools, for different categories of authors and applications, to easily generate documents in standard formats. Our approach is to focus on the logical structure of the document while keeping some semantics of proven technologies such as SMIL (SMIL, 2005). This provides better modularity, facilitates the definition of document templates, and improves manipulation and reusability of content. The LimSee3 authoring process is given on figure 1: a document is created from a template by adding content in an application-guided way. The obtained LimSee3 document can be exported into one or several presentation documents suitable for rendering.

This paper is organized as follows: next section presents a scenario example that will be developed throughout the paper and thereby analyzes CoPs requirements for authoring multimedia documents. We then define the main concepts on which multimedia authoring tools are based and we classify existing approaches in the light of these concepts. After that, we introduce the LimSee3 document model and show how it can be used for the development of authoring tools tuned for specific CoPs. Last section presents the current state of our development and our perspectives.

This work was partially supported by the European project FP6-2005-IST-028038 Palette.

A Learning-Oriented Example of Authoring

Multimedia Story-Telling for Enhanced Learning

Educators have integrated practice into their curriculum to different degrees; the Figure 2 shows this continuum and how LimSee3 can be naturally used to enhance authoring multimedia documents.

Edward Bilodeau (Bilodeau, 2003) illustrated that moving towards full immersion requires substantial changes to course design. Careful consideration must be given to the optimal location for student learning to occur on this continuum. Using templates in LimSee3 authoring tool for pedagogical approach allows production process during this continuum. It gives a way of making things simpler and faster to teachers and writers. It focuses on pedagogical issues. It produces practical Units of Learning (UoL).

Researchers such as Dolores Durkin (Durkin, 1961), Margaret Clark (Clark, 1976), Regie Routman (Routman, 1988)(Routman, 1991), and Kathy Short (Short, 1995) have found evidence that children who are immersed in rich, authentic literary experiences become highly engaged in literature and develop literary awareness. Their studies revealed that positive and meaningful experiences with books and written language play a critical role in the development of literacy skills. Other researchers have found that students acquired reading and thinking strategies in literature-based programs that included teacher-led comprehension instruction (Block, 1993) (Baumann & Ivey, 1997) and (Goldenberg, 1992).

Story-telling in the learning process.

Stories are basic: we start telling our children about the world by telling them stories. Stories are memorable: their narrative structure and sequence makes them easy to remember. "What happens next?" is a very basic form of interest and engagement. Stories are everywhere: very rarely in real life do we set out to convey ideas in terms of hierarchies or taxonomies or bullet-points. Instead we tell stories. Teaching is one of the predominantly professional activities which do habitually communicate by means of stories and also use elaborated language codes.

We want to go deeper inside the multimedia process taking into account the advantage of creating multimedia in the immersion process. As an example, U-Create (Sauer, Osswald, Wielemans, & Stifter, 2006) is an authoring tool centered on 2D and 3D scenes targeted to non-programmers who want to easily produce story documents. The tool is based on predefined structural elements (from story, scene, to action, stageSet and asset) and associated dedicated GUIs.

In this paper we consider a group of teachers that are working together - a CoP in our terminology - to create and share course materials based on tale story-telling.

Tale Learning Example

Little Red Riding Hood example.

Little Red Riding Hood [1] is a well-known and well-loved fairytale dating back to over three centuries ago when it was first published by Charles Perrault in his Histoires ou Contes du temps passé in 1697 and based on oral European tales. Since then, Little Red Riding Hood has been retold in a variety of forms and styles, as Big Books and Lift-the-flap books, as poems and plays, and whilst some details may have changed, many of the essential elements have stayed the same. Little Red Riding Hood makes a great literature teaching unit theme for elementary school.

A general synopsis follows:

Act I.1 Setting at the edge of a forest; the Little Red Riding Hood goes off to take a basket to her ill grandmother, her mother warns her not to dawdle in the woods or to talk to strangers.

Act I.2 A place inside the forest. Woodcutters can be heard chopping wood. Little Red Riding Hood comes out of some bushes. As she pauses to pick some flowers, and the wolf catches sight of her. On the path he stops her and makes up a story about a shortcut the grandmother's house. When he challenges her to see who will get there first, she agrees, and both of them run off in different directions as the woodcutters resume their work.

Act II.1 The chorus explains that the wolf has not eaten for three days and was able to get to grandmother's house first. The wolf, pretending to be Little Red Riding Hood, manages to get into the house and swallow grandmother. He takes her place in the bed before Little Red Riding Hood arrives. In several questions she expresses her surprise at how differently grandmother looks now, and the wolf swallows her.

Act II.2 Some hunters and woodcutters, who have been tracking the wolf, come by and enter the house. They find the wolf in his sleep and open his belly to let grandmother and Little Red Riding Hood out. After they sew up the wolf again, he repents and is permitted to live in the forest as long as he lives up to his promise to be good.

In the learning process, it is possible to exploit this story in different approaches (*Francparler.org – un site de l'Organisation internationale de la Francophonie*, 2006), for instance:

Story and the Time: a) After reading of the tale and giving explanations of difficult points and incomprehensions, to work on the chronology from drawings. b) Try and feel the knowledge of the terms "before", "later", during this time line.

Oral expression: a) Drawing images and explain them. b) Playing dialogues without written support. c) Reciting a rhyme or poem, singing a song.

A variety of resources, ranging from texts, illustrations, media presentations to computer based interactive materials for students are available for use in classroom. Based on these materials, a teacher can propose:

[1] As a coincidence, the Little Red Cap (Riding Hood) tale has also been the reference story in the workshop organized by U. Spierling and I. Iurgel on *The Authoring Process in Interactive Storytelling*, December 2006, 3rd.

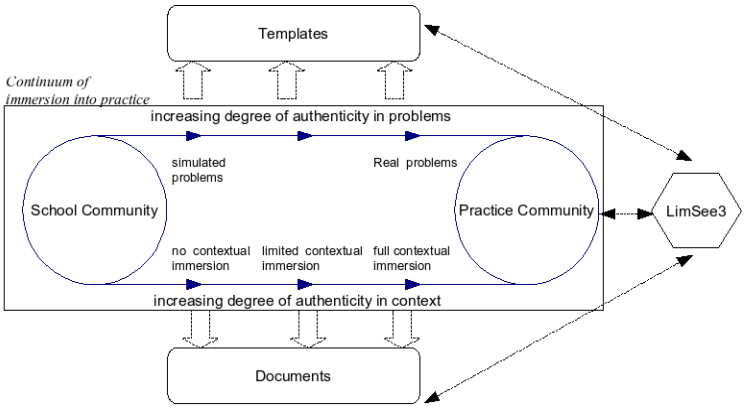

Figure 2. Continuum of immersion into practice adopted from Hogan (Hogan, 2002) and LimSee3 use

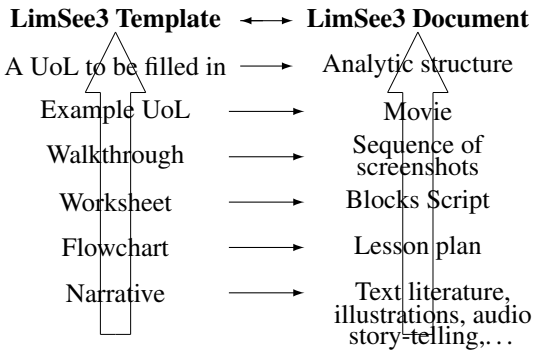

Figure 3. LimSee3 template and document links

- *Story* - tell the story, watch and comment movies
- *Songs* - organize some spoken drill type activities
- *Handcraft activities* - propose drawing, folding, coloring of sceneries, puppets, etc.
- *Play* - study and put on stage a personalized version of the story

Basically, the Units of Learning that are exchanged in this CoP of teachers are multimedia story documents that are composed of sequences of story steps where data elements are heterogeneous and multimedia. The challenges are to enrich information with the synchronization of data elements (for instance an activity with the corresponding material) and to provide a document structure enabling knowledge sharing and reusability (of stories).

The CoP of teachers needs templates for making things simpler, faster, for being focused on pedagogical issues, for producing practical Units of Learning. The Fig. 3 shows the structural link between LimSee3 Template and LimSee3 document contents. At the lower level, a narrative part inside the template corresponds to text literature and/or illustration and/or audio story telling. At a higher level, a template walkthrough corresponds to a sequence of screenshots. The first level is offered by the BNF (BNF, 2001) which for instance gives out textual contents and illustrations. To fully instantiate upper levels, we show a possible making of the tale with a logic modeling [template and document] from which we can extract levels to enhance associated authoring.

From the continuum of immersion into practice, represented inside the rectangle of the Fig. 2, we learn that the greater the degree of authenticity of the learning activities, the more the students will be able to be integrated into the practice. Different programs and courses benefit from different levels of immersion.. Moving towards full immersion requires substantial changes to course design. Teachers need authoring tools to set up these types of pedagogical materials. Careful consideration must be given to the optimal location for student learning to occur in this continuum. In this CoP, a number of teachers will create templates to build this type of very specialized tools. Such a modeling will naturally emerge from CoP work, using LimSee3 inside this continuum (see Fig. 2).

Basic Requirements for CoP-Oriented Learning

In order to be useful, the cooperative services to be provided to the CoPs must have the two following basic features: (i) authoring tool of stories dedicated to teachers ; (ii) access tool to read the existing stories. Looking more closely at the ways in which CoPs participants are producing multimedia information, we can identify some requirements for the authoring and presentation platform:

1. Simple and efficient authoring paradigms - because CoPs members are not (always) computer science technicians.
2. Easy and rapid handling of the authoring tool - because new members can join CoPs.
3. Modular and reusable content - because multimedia information results in a co-construction process between members.
4. Evolutive structuring of documents - because of the dynamic nature of CoPs objectives.
5. Use of standard formats - because CoPs need portability, easy publishing process and platform-independence.

Basically, our approach proposes a template mechanism to cope with requirements 1 and 2, a component-based structuring enabling requirements 3 and 4, and relies on proven

standard technologies to ensure the last requirement. Before further stating our authoring model, we present in the next section the main concepts and approaches of multimedia authoring on which this work is based.

Multimedia Documents and Multimedia Authoring

In traditional text oriented document systems, the communication mode is characterized by the spatial nature of information layout and the eye's ability to actively browse parts of the display. The reader is active while the rendering itself is passive. This active-passive role is reversed in audio-video communications: active information flows to a passive listener or viewer. As multimedia documents combine time, space and interactivity, the reader is both active and passive. Such documents contain different types of elements such as video, audio, still-picture, text, synthesized image, and so on, some of which having intrinsic duration. Time schedule is also defined by a time structure synchronizing these media elements. Interactivity is provided through hypermedia links that can be used to navigate inside the same document and/or between different documents.

Due to this time dimension, building an authoring tool is a challenging task because the WYSIWYG paradigm, used for classical documents, is not relevant anymore: it is not possible to specify a dynamic behavior and to immediately see its result. Within the past years, numerous researches have presented various ways of authoring multimedia scenarios, focusing on the understanding and the expressive power of synchronization between media components: approaches can be classified in absolute-based (Authorware, 2004), constraint-based (Buchanan & Zellweger, 1993), (Jourdan, Layaïda, Roisin, Sabry, & Tardif, 1998), event-based (Sénac, Diaz, Léger, & Saqui-Sannes, 1996) and hierarchical models (SMIL, 2005), (Rossum, Jansen, Mullender, & Bulterman, 1993). Besides, to cope with the inherent complexity of this kind of authoring, several tools (Authorware, 2004), (Producer, n.d.), (Hua, Wang, & Li, 2005) have proposed limited but quite simple solutions for the same objective. Dedicated authoring, template-based authoring and reduced synchronization features are the main techniques to provide reasonable editing facilities. But we can notice that these tools generally also provide scripting facilities to enrich the authoring capabilities and therefore loose in some way their easiness.

Beside timelines, script languages and templates, intermediate approaches have been proposed through "direct manipulation" and multi-views interface paradigms. IBM XMT authoring tool (XMT, n.d.) and SMIL tools such as LimSee2 (LimSee2, 2003–2006) and Grins (Grins, n.d.) are good examples. In LimSee2, the time structure of SMIL is represented for instance in a hierarchical timeline as shown in of Fig. 2 (4). Time bars can be moved or resized to finely author the timing scenario. This kind of manipulation has proven very useful to manipulate efficiently the complex structures representing time in multimedia XML documents.

However even if XMT and SMIL are well-established languages, the above-mentioned tools are too complex for most users because they require a deep understanding of the semantics of the language (e.g. the SMIL timing model). Moreover these models generally put the time structure at the heart of the document whereas it does not always reflect exactly the logical structure in the way it is considered by the author. Our approach instead sets this logical dimension as the master structure of the document, which is a tree of modular components each one specifying its own time and spatial structures. Additionally, the document can be constrained by a dedicated template mechanism.

A template document is a kind of reusable document skeleton that provides a starting point to create document instances. Domain specific template systems are a user-friendly authoring solution but require hardly extensible dedicated transformation process to output the rendering format. We chose on the contrary to tightly integrate the template syntax in the document: the template is itself a document constrained by schema-like syntax. The continuum between both template and document permits to edit templates as any other document, within the same environment, and enables an evolutive authoring of document instances under the control of templates. There is no need to define a dedicated language to adapt to each different use case.

We believe that the combination of document structuring and template definition will considerably help CoPs in (i) reusability of materials, (ii) optimization of the composition and life cycle of documents, (iii) development and transmission of knowledge, (iv) drawing global communities together effectively.

The LimSee3 Authoring Language

Main Features

In the LimSee3 project, we define a structured authoring language independently of any publication language. Elements of the master structure are components that represent semantically significant objects. For instance a folktale can be seen as a sequence of scenes. Each step is composed of several media objects and describes a phase of the story (departure from home, encountering the wolf,...). Components can be authored independently, integrated in the document structure, extracted for reusability, constrained by templates or referenced by other components.

The different components of a multimedia document are often tightly related with one another: when they are synchronized or aligned in space, when one contains an interactive link to another, and so on. Our approach, which is close to the one proposed in (Silva, Rodrigues, Soares, & Saade, 2004) is for each component to abstract its dependencies to external components by giving them symbolic names. This abstraction layer facilitates the extraction of a component from its context, thus enhancing modularity and reusability.

Finally, the goal is to rely on existing proven technologies, in both contexts of authoring environments and multimedia representation. The timing and positioning models

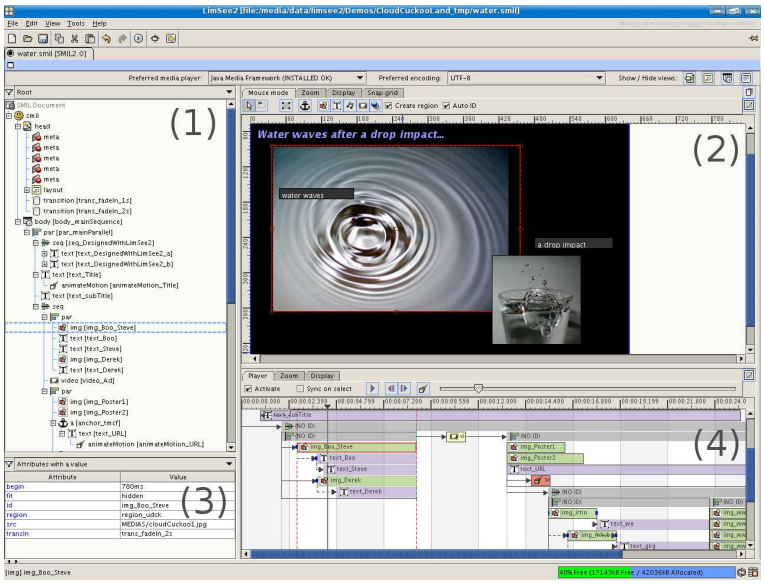

Figure 4. Multiview authoring in LimSee2

are wholly taken from SMIL. Using XML (XML, 2006) provides excellent structuring properties and enables the use of many related technologies. Among them are XPath (XPATH, 1999), used to provide fine-grained access to components, and XSLT (XSLT, 1999) , used in templates for structural transformation and content generation.

The authoring language is twofold: it consists in a generic document model for the representation of multimedia documents, and it defines a dedicated syntax to represent templates for these documents.

In this section we describe the main features of the LimSee3 language and we illustrate their syntax with short excerpts of the story-telling example.

Document Model

A *document* is no more than a `document` element wrapping the root of the object hierarchy and a `head` element containing metadata. This greatly facilitates the insertion of the content of a document in a tree of objects, or the extraction of a document from a sub-tree of objects.

A *compound object* is a tree structure composed of nested objects. Each compound object is defined by the `object` element with the `type` attribute set to `compound`. It contains a `children` element that lists child objects, a `timing` element that describes its timing scenario, a `layout` element that describes its spatial layout and a `related` element that abstracts out dependencies to external objects.

The value of the `localId` attribute uniquely identifies the component in the scope of its parent object, thereby also implicitly defining a global identifier `id` when associated with the `localId` of the ancestors. In Example 1, the first child of object `scene1` has the local id `title` and hence is globally identified as `scene1.title`.

The timing model, and similarly the positioning model, is taken from SMIL 2.1. The `timing` element defines a SMIL time container. The timing scenario of a component is obtained by composition of the timed inclusions defined by the `timeRef` elements, whose `refId` attributes are set to local ids of children.

A *media object* is actually a simple object that wraps a media asset, i.e. an external resource (such as an image, a video, an audio track, a text...) referenced by its URI. It is defined by the `object` element with the `type` attribute set to either `text`, `image`, `audio`, `video` or `animation`. The URI of the wrapped media asset is put into the `src` attribute. Example 2 shows an image media object with local id `right-button` which wraps the media asset identified by the relative URI `./medias/right-arrow.png`.

Area objects inspired from the SMIL `area` element can be associated to media objects. They are used for instance to structure the content of a media object or to add a timed link to a media object. An area is defined as an `object` element with the `type` attribute set to `area`. For instance, in Example 2 the media object `right-button` has a child area which defines a hyperlink.

Relations of dependency between objects are described independently of their semantics in the document. External dependencies are declared with `ref` elements grouped inside the `related` child element of objects. The value of `refId` of a `ref` element is the id of the related element and the value of `localId` is a symbolic name that is used within the object to refer to the related object. For instance, in Example 2, object `right-button` provides a clickable image that links to the object `story.scene2`, by first declaring the relation in a `ref` element and then using this external object locally named `target` to set the value of the `src` attribute of the

```
<document xmlns="http://limsee3.gforge.inria.fr/ns"
  xmlns:template=".../ns/template"
  xmlns:smil="http://www.w3.org/2005/SMIL21/">
  <head><!-- some metadata --></head>
  <object localId="scene1" type="compound">
    <children>
      <object type="text" localId="title">
        ...
      </object>
      <object type="image" localId="illustration1">
        ...
      </object>
      ...
      <object type="compound"
        localId="navigation-bar">
        ...
      </object>
    </children>
    <timing>
      <smil:par dur="30min">
        <timingRef refId="title" />
        <timingRef refId="illustration1" />
        ...
        <timingRef refId="navigation-bar" />
      </smil:par>
    </timing>
    <layout height="600" width="800" />
  </object>
</document>
```

Figure 5. **Example 1** A simple scene LimSee3 document

link, using `attribute` and `value-of` elements taken form XSLT.

```
<object localId="right-button" type="image"
  src="./medias/right-arrow.png">
  <related>
    <ref localId="target" refId="story.scene2"/>
  </related>
  <children>
    <object type="area" localId="link">
      <attribute name="src">#<value-of
        refName="target" select="@id" />
    </object>
  </children>
  ...
</object>
```

Figure 6. **Example 2** A LimSee3 object with an external dependency relation

Templates

Template nodes aim at guiding and constraining the edition of the document. In order to have better control and easy GUI set up, the language includes two kinds of template nodes: media place holders and repeatable structures.

A *place holder* is a template node that defines a reserved place for a media object. It is represented by an `object` element, whose `type` and `src` attributes are not (yet) set. It specifies the kind of media resources it accepts in a special `template:types` attribute (the values can be `text`, `img`, `audio`, `video`, `animation`, or a list of these types). The author can also specify content that will be displayed to invite the user to edit the media zone with the `template:invite` element (of any media type). For instance Example 3 shows a media place holder `title` for a text, with textual invitation. During the authoring process place holders are filled with media objects inserted by the user.

A *repeatable structure*, represented by the `template:model` element, is a possibly complex template node that defines a homogeneous list of objects. Each item of the list matches the model. The cardinality of the list can be specified with the `min` and `max` attributes. Example 3 shows a tale scene template named `tale-scene`: this complex model is composed of several place-holders (`title`, `questions`), an embedded model (`illustrations`) and the `navigation` object.

Finally, our model makes it is possible to lock out parts of a document with the `locked` attribute, to prevent authors from editing these parts. This allows for instance to guide more strongly inexperienced users by restricting their access to the only parts of the document that make sense to them.

Authoring with LimSee3

The LimSee3 model defined in the previous section aims at being an internal kernel of our authoring system as can be seen on figure 8. This model is hidden from the user by several abstraction layers: experienced users interact with a full-featured generic platform (LimSee3 core) that enables them to finely tune all document properties, however at the cost of some technical overhead, while basic users interact with template-specific simplified interfaces that allow them to author documents with less effort. In the case of a teachers CoP that wants to create and share multimedia materials for tale-telling, the benefits that could be achieved with LimSee3 are the following:

- *to describe the author's vocabulary* by structuring basic medias into author-defined logical multimedia structures ("tale scenes" viewed as collections of "illustrations" and "narrations", rather than mere "pictures" and "texts")
- *to adopt the author's vocabulary in the authoring process* by leveraging the logical coherence between the document to produce and the way to produce it (the template structure reflects the logics of the presentation, not its technical needs)
- *to facilitate document reuse in the CoP* by easily extracting, adapting, merging documents, applying alternate layouts for different pedagogical purposes

Figure 9 shows the different steps of the production of tale-tellings by the member of teachers CoP. First, an experienced author creates a tale story from scratch using the LimSee3 core platform (flow (1) in the figure 9) in order to define le logical structure of this pedagogical material that will allow a fruitful use in classroom. Eventually this multimedia tale will be refined thanks to inputs from other teachers of the CoP. When a consensus is reached, this teacher can use

```
<template:model name="tale-scene" min="1">
  <object>
    <children>
      <object localId="title" template:types="text">
        <children>
          <template:invite type="text">Fill in the title of this scene</template:invite>
        </children>
      </object>
      <template:model name="illustrations" min="1" max="5">
        ...
      </template:model>
      <object localId="questions" template:types="text">...</object>
      <object localId="navigation" type="complex">...</object>
    </children>
    <timing>...</timing>
    <layout>...</layout>
  </object>
</template:model>
```

Figure 7. **Example 3** A scene template

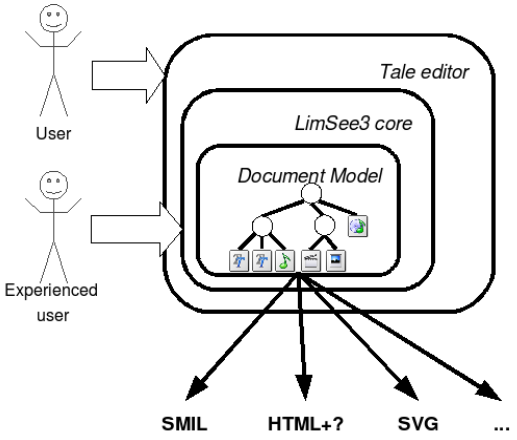

Figure 8. The LimSee3 three-layer architecture

the LimSee3 core to extract a template document from this instance (flow (2) in the figure 9). The main structure of the document, in this case a sequence of scenes, can be constrained by template nodes such as repeatable structures. The result of this step will be a dedicated authoring interface that other teachers can use (flow (3) in the figure 9) to create new multimedia tale stories. This is a typical example of participative design leading to the development of a dedicated tool based on the LimSee3 generic platform.

The figures 10 and 11 illustrate this last step of authoring with a dedicated GUI:

- Figure 10 shows how the place-holders defined in the template structure can lead to simple drag-and-drop authoring actions.
- Figure 11 illustrates the advantages gained with the separation of logical, spatial and time information. It allows the authoring and rendition of several scenarios of the same content: thanks a direct manipulations in the timeline view, an author has defined a sequential display of the illustrations instead of the default parallel one.

Finally, the proposed application can evolve to take into account new needs of the CoP members. For instance, a teacher wants to register his/her course, using a camera that films her/him while (s)he gives a talk illustrated with the multimedia tale document. In order to easily synchronize the video with the different parts of the tale document, the authoring tool is enriched with a simple control panel, as can be seen on the left part of figure 12.

Conclusion

The LimSee3 model leads to the development of authoring tools that fit the requirements stated at the beginning of the paper. The LimSee3 core is currently under development as a cross-platform open-source Java software: we provide this generic platform with widgets to manipulate all the elements defined in the model (documents, compound objects, timing and layout details, relations...). It provides features based on the proven authoring paradigms such as multi-views, timeline, structure tree an 2D canvas.

The model presented in this paper develops a practice-

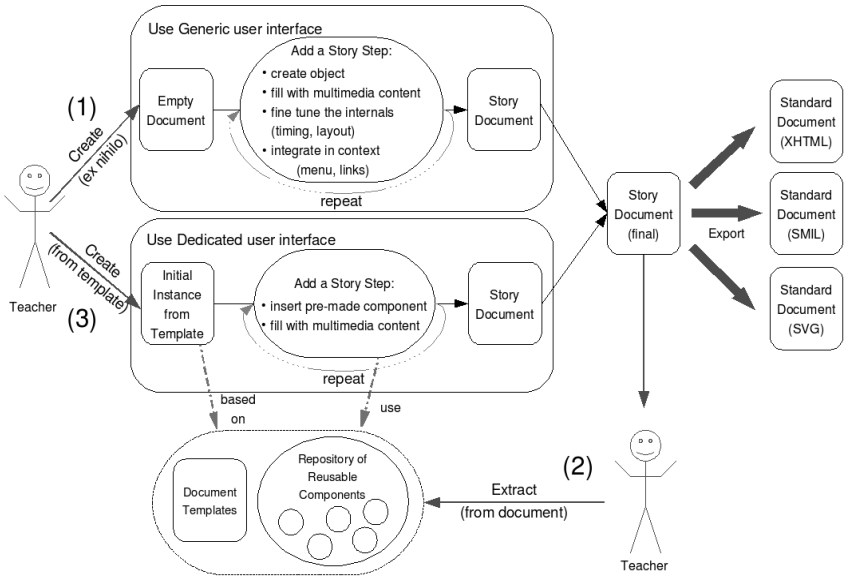

Figure 9. Authoring with LimSee3

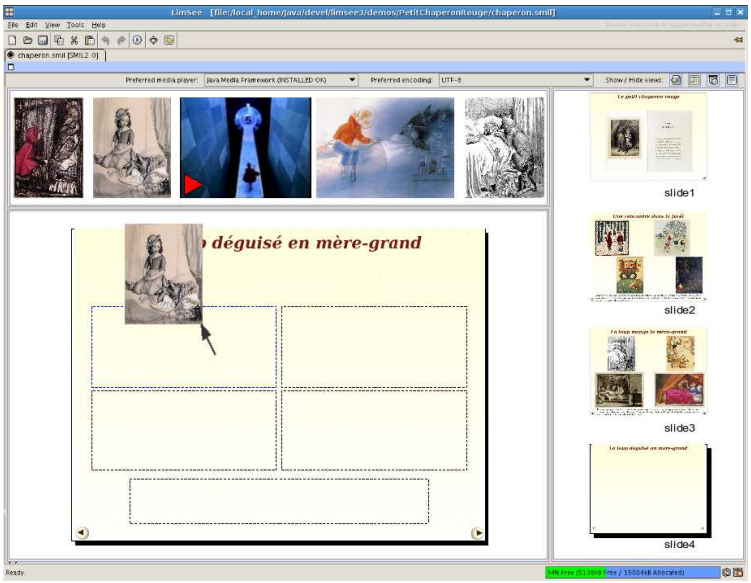

Figure 10. Instantiating a template document by drag-and-drop

based approach to multimedia authoring dedicated to communities where collaborative and participative design is of high importance. It improves reusability with template definitions and with the homogeneous structuring of documents. In the context of Palette, we will use this model to develop dedicated authoring tools for pedagogical CoPs.

References

Adobe Authorware 7 and Director MX 2004. (2004). http://www.adobe.com/products/.

Authoring in XMT. (n.d.). http://www.research.ibm.com/mpeg4/Projects/AuthoringXMT/.

Autour du Petit Chaperon rouge. (2001). http://expositions.bnf.fr/contes/pedago/chaperon/.

Baumann, J. F., & Ivey, G. (1997). Delicate balances: Striving for curricular and instructional equilibrium in a second-grade, literature/strategy-based classroom. *Reading Research Quarterly*, *32(3)*, 244.

Bilodeau, E. (2003). *Using communities of practice to enhance student learning: examples and issues, written by edward bilodeau from mcgill university*.

Block, C. C. (1993). Strategy instruction in a literature-based reading program. *Elementary School Journal*, *94(2)*, 139–151.

Brusilovsky, P. (2003). Developing adaptive educational hypermedia systems: From design models to authoring tools. In T. Mur-

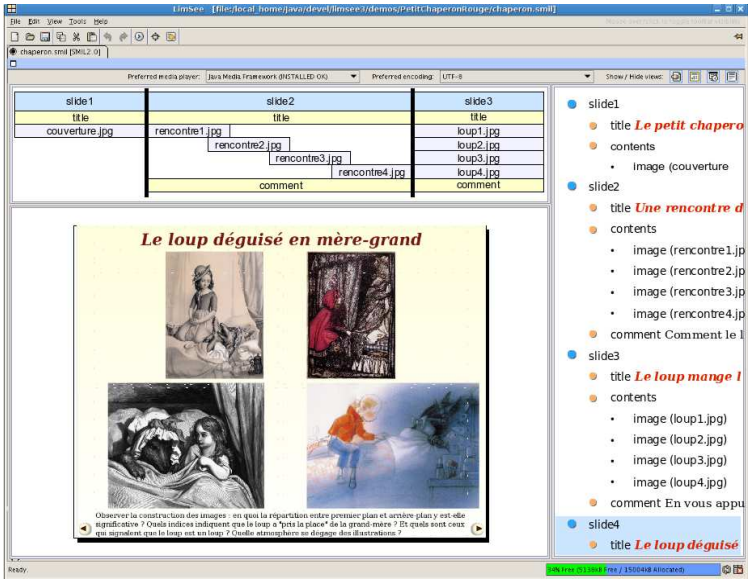

Figure 11. Modifying the timeline

Figure 12. Synchronizing a video with a story document

ray (Ed.), *Authoring tools for advanced technology learning environments: Toward cost-effective adaptive, interactive and intelligent educational software.* Kluwer Academic Pub.

Buchanan, M. C., & Zellweger, P. T. (1993). Automatic temporal layout mechanisms. In *Computer graphics (multimedia '93 proceedings)* (pp. 341–350). Addison-Wesley.

Clark, M. (1976). *Young fluent readers: What can they teach us?* Heinemann.

Durkin, D. (1961). Children who read before grade one. *The Reading Teacher*, *14*, 163–166.

Extensible Markup Language (XML) 1.1. (2006). http://www.w3.org/TR/xml11/.

Francparler.org – un site de l'organisation internationale de la francophonie. (2006). La communauté mondiale des professeurs de français.

Goldenberg, C. (1992). Instructional conversations: Promoting comprehension through discussion. *The Reading Teacher*, *46(4)*, 316–326.

Grigoriadou, M., & Papanikolaou, K. (2006). Authoring personalised interactive content. In *Proceedings of the first international workshop on semantic media adaptation and personalization (SMAP'06)* (pp. 80–85). Los Alamitos, CA, USA: IEEE Computer Society.

Hoffmann, P., & Herczeg, M. (2006). Interactive narrative systems -

hypervideo vs. storytelling integrating narrative intelligence into hypervideo. *Lecture Notes in Computer Science*, *0*(4326), 37–48.

Hua, X., Wang, Z., & Li, S. (2005). *LazyCut – Content-Aware TemplateBased Video Authoring.*

Jourdan, M., Layaïda, N., Roisin, C., Sabry, L., & Tardif, L. (1998). Madeus, an authoring environment for interactive multimedia documents. In *Acm multimedia'98.*

LimSee2. (2003–2006). http://limsee2.gforge.inria.fr/.

MS Producer for PowerPoint. (n.d.). http://www.microsoft.com/office/powerpoint/producer/.

Oratrix Grins. (n.d.). http://www.oratrix.com/.

Rossum, G. van, Jansen, J., Mullender, K., & Bulterman, D. (1993). CMIFed : a presentation environment for portable hypermedia documents. In *Acm multimedia'93.*

Routman, R. (1988). *Transitions: From literature to literacy*. Heinemann.

Routman, R. (1991). *Invitations: Changing as teachers and learners k-12*. Heinemann.

Sauer, S., Osswald, K., Wielemans, X., & Stifter, M. (2006). Story authoring - U-Create: Creative authoring tools for edutainment applications. *Lecture Notes in Computer Science*, *0*(4326), 163–168.

Short, K. (1995). *Research and professional resources in children's literature: Piecing a patchwork quilt*. International Reading Association.

Silva, H., Rodrigues, R., Soares, L., & Saade, D. M. (2004). NCL 2.0: integrating new concepts to XML modular languages. In *Acm doceng'04.*

Sénac, P., Diaz, M., Léger, A., & Saqui-Sannes, P. de. (1996). Modeling logical and temporal synchronization in hypermedia. *IEEE J. on Sel. Areas in Comm.*, *14*(1).

W3C SMIL 2.1. (2005). http://www.w3.org/TR/SMIL2.

XML Path language (XPath) 1.0. (1999). http://www.w3.org/TR/xpath.

XSL Transformations 1.0. (1999). http://www.w3.org/TR/xslt.