Project no. FP6-028038

Palette

Pedagogically sustained Adaptive LEarning Through the
exploitation of Tacit and Explicit knowledge

Integrated Project

Technology-enhanced learning

D.INF.03 – First version of LimSee3 offering a template mechanism and a customized user interface

Due date of deliverable: March 31, 2007
Actual submission date: April 16, 2007

Start date of project: February 01, 2006 Duration: 36 months

Organisation name of lead contractor for this deliverable: INRIA

Project co-funded by the European Commission within the Sixth Framework Program	
Dissemination Level	
PU	Public

Keywords: Document authoring, structured documents, templates
Responsible Partner: INRIA

MODIFICATION CONTROL			
Version	Date	Status	Modifications made by
1	10-04-2007	draft	Jan Mikáč, Cécile Roisin
2	12-04-2007	draft	Nikos Karacapilidis, Jan Mikáč
3	13-04-2007	draft	Étienne Vandeput, Jan Mikáč
4	16-04-2007	Final	Jan Mikáč

Deliverable manager

Vincent Quint (INRIA)

List of Contributors

Jan Mikáč (INRIA), Cécile Roisin (INRIA)

List of Evaluators

Étienne Vandeput (ULg), Nikos Karacapilidis (CTI)

Summary

This document presents the first version of the LimSee3 multimedia authoring tool. We briefly recall the context in which it was designed, then we describe the main features of this version and its limitations and we illustrate its use through an extensive example.

TABLE OF CONTENT

1. Introduction

Multimedia documents have become part of the everyday life - we encounter them in entertainment, learning/teaching and more generally in exchange of knowledge and experience. As a matter of fact, multi-media is well suited for sharing practical information, since it provides a multi-modal way of expressing it, thus facilitating understanding. It is no wonder if users have largely adopted multimedia as consumers - it makes their lives easier.

Unfortunately, producing multimedia documents is not so easy a task, due to some intrinsic difficulties - i) multiple media types and formats have to be manipulated, ii) different components of a presentation have to be synchronized and iii) presentation should be interactive, which raises the level of intricacy. Existing authoring tools cope with this complexity in two ways: either they suppose a high level of expertise and leave it to the user to resolve those technical issues, or they are very domain- and application-specific, in which case they strongly guide the user in the production of a document with only a little possibility of customization.

Given that Community of Practice (CoP) members are interested in using multimedia for sharing their practical experience and knowledge, but that they often do not have expertise in the multimedia field, LimSee3 proposes an alternative approach to the authoring process, based on an extensive use of adaptable document templates. This approach promotes easiness of use, yet flexibility and is based on an original model of (multimedia) documents, which we defined and built atop XML.

The design principles of LimSee3 and its targeted features are presented in D.INF.01 [1], which defines the architecture of information services in Palette. More information about the document model can be found in [2], [3] and [5]. A user-oriented presentation of it appears in [4].

LimSee3 is developed as an entirely new open-source project. It is not a direct successor of LimSee2, since LimSee2 is a SMIL-authoring tool, while LimSee3 aims at multimedia authoring in general. Developing from scratch has the inconvenience of being lengthy, yet it allows us to be compatible with the principle of participative design, as we can integrate user comments and requests along the road. In particular, Palette partners are encouraged to provide feed-back about the first version of LimSee3 delivered with this document.

The software presented here is publicly available via the Palette Tools&Prototypes page (http://palette.ercim.org/content/view/13/30/), section LimSee3.

In the sequel, we present the first version of LimSee3. We discuss its main features and present an example of use. More technical matters like the installation procedure are given in an appendix.

2. Main features

This deliverable is the first release of the LimSee3 multimedia authoring tool. Its purpose is to demonstrate template-driven editing of documents, particularly in its convenient and easy-of-use dimension. The implementation is not complete yet (it is considered as version 0.5), but available functionalities are sufficient to make experiments and to gather comments from users.

In this release, the main features are:

- creation of a document by instantiation of a template
- media asset/media library handling
- customizable user interface

Simple document authoring is the basic functionality that is expected from a document editor. Currently, LimSee3 proposes to create new documents by instantiation of a template: a template can be seen as a "document with holes" and the instantiation process consists in filling the holes with media content. The spatial and temporal layouts, as well as interactive features of the resulting document are automatically provided by the template, so that the user does not have to worry about them. Thus, creating a document as instance of a template is an easy task, mostly achieved by drag-and-drop or copy-paste gestures.

LimSee3 organizes medias in *media libraries*. The purpose of a library is to provide a uniform view of some sets of local and distant media assets. Libraries do not contain the assets, they merely reference them: thus an asset present in several libraries is not physically duplicated; and a distant asset is not downloaded for local storing. Libraries allow users to organize assets into logical sets, independently of their actual locations, thus providing an abstraction level between the media files and the media objects manipulated in a LimSee3 document.

Finally, the adaptability of the graphical user interface has been one of the initial requirements of LimSee3. As part of the effort to achieve this requirement, functionalities supporting *customization* have been implemented: the user is free to organize the workspace at his/her convenience. In particular, visual elements can be moved, minimized, maximized, or detached (floating). Moreover, some elements (namely the media libraries) already provide a support for *accessibility*, that is the possibility to adjust their rendering accordingly to the visual performance of the user.

All the above mentioned points are illustrated through an example in section 4.

3. Current limitations and possible evolutions

The version of LimSee3 delivered with this document is not complete with respect to its expected functionalities. This version constitutes a basis on which developpers and users will build next releases. In particular user feed-back (or even active participation on the development) will provide useful insights and directions for future work. Currently, the following features are not available:

- Documents created by template instantiation can only be stored in the LimSee3 internal format. In particular, it is not yet possible to view the documents in some external viewer. We plan to provide support for exporting into SMIL within a few weeks.
- Currently, LimSee3 allows only basic instantiation: assets can be inserted into reserved places, but they cannot be further manipulated. A support for adjusting the spatial and temporal behavior will be included in some forthcoming release.
- Document templates cannot be edited yet in LimSee3. A limited support for template authoring will come along with the previous point, yet a full support is subject to some theoretical issues that are still under study. For the time being, document templates can be edited directly in XML (with a plain text editor), but only users familiar with the LimSee3 document model would be able to do so.
- LimSee3 manipulates all kinds of medias, but is able to render only texts and images; rendering sounds and videos is a mid-term aim. Notice that such a limitation is not very restrictive for document authoring, because all documents will soon be exportable into SMIL, so that they will be viewable in external players.
- The application communicates with the user in English. The current implementation already supports localization, but no translation of the dialogs is available yet. Localization in French and other languages will be available shortly. Moreover, a translation tool will be provided to allow users to produce other localizations on their own; these localizations can be integrated in LimSee3, thus making benefit of a participative design.
- User interface customization is not persistent yet: re-starting LimSee3 results in reloading the default interface layout. Customization persistence will be provided as part of a larger "user-preference saving" functionality.
- There is currently no user guide (or manual) to LimSee3. We will provide one shortly, under the form of a tutorial. It will be accessible through the Palette tool page (Palette > Tools&Prototypes > LimSee3).
- The user interface is far from being definitive. Comments from partners and in particular usability analysis will help us improve this part of the application.

4. Example of use

In this example, we will present the use of LimSee3 with the template of a slide-show. We recommend to view this section in the electronic document, for better rendering of included images.

The slide-show template allows to create a presentation formed by several slides that are designed to be viewed in a sequence. As we shall see in the sequel, each slide has a title, two navigation buttons (previous and next) and can contain several media objects. The precise number and layout of these objects is further specified by the means of sub-templates. There are currently three sub-templates, allowing the user to choose between a "cover" slide (one main media object), a "low populated" slide (two media objects) and a "populated" one (four media objects).

4.1 Basic document authoring

When launched, LimSee3 presents an empty interface (see figure 1). In fact, some output has already been produced as part of the internal logging system: this output is by default not visible, since it is of little interest to the user, but it may become important if a bug is submitted to the developing team. The log console can be brought forth with the bottom-left button (fig. 2).

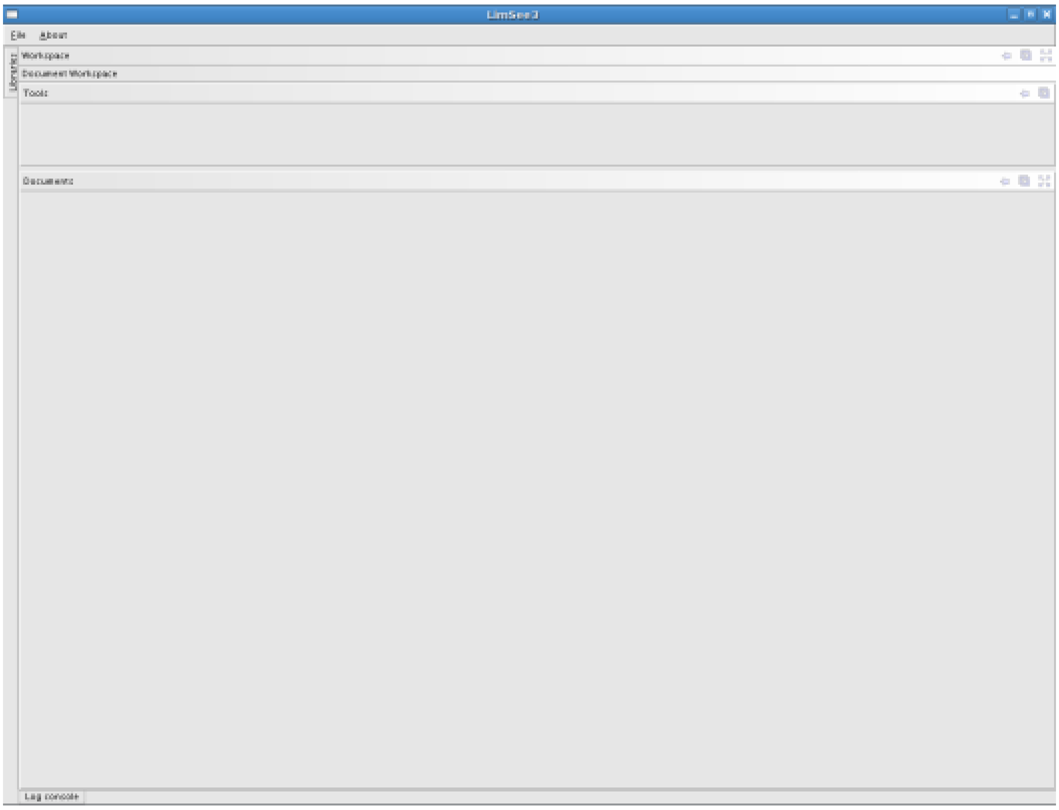
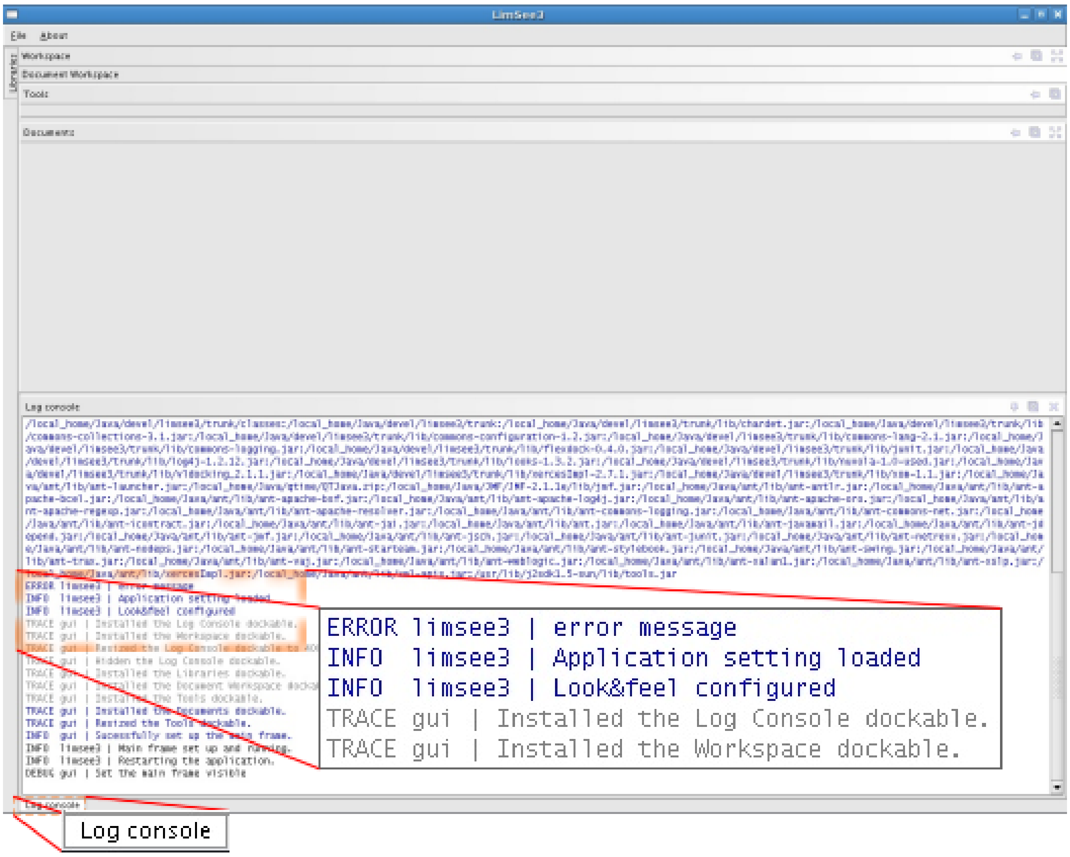


Fig.1: Initial screen



```
ERROR limsee3 | error message
INFO  limsee3 | Application setting loaded
INFO  limsee3 | Look&feel configured
TRACE gui | Installed the Log Console dockable.
TRACE gui | Installed the Workspace dockable.
TRACE gui | Resized the Log Console dockable to 400
TRACE gui | Hidden the Log Console dockable.
TRACE gui | Installed the Libraries dockable.
TRACE gui | Installed the Document Workspace docka
TRACE gui | Installed the Tools dockable.
TRACE gui | Installed the Documents dockable.
TRACE gui | Resized the Tools dockable.
INFO  gui | Successfully set up the main frame.
INFO  limsee3 | Main frame set up and running.
INFO  limsee3 | Restarting the application.
DEBUG gui | Set the main frame visible
```

Fig.2: Log console with zoomed opening button and a part of the content

All document-related actions are initialized from the File menu. The first possible action is to open an existing document. We choose to edit a document entitled "slideshow-1.xml" which is in fact a template document implementing the slideshow template. When opened, we can see the first (non-instantiated) slide and a dedicated slideshow toolbar (fig. 3).

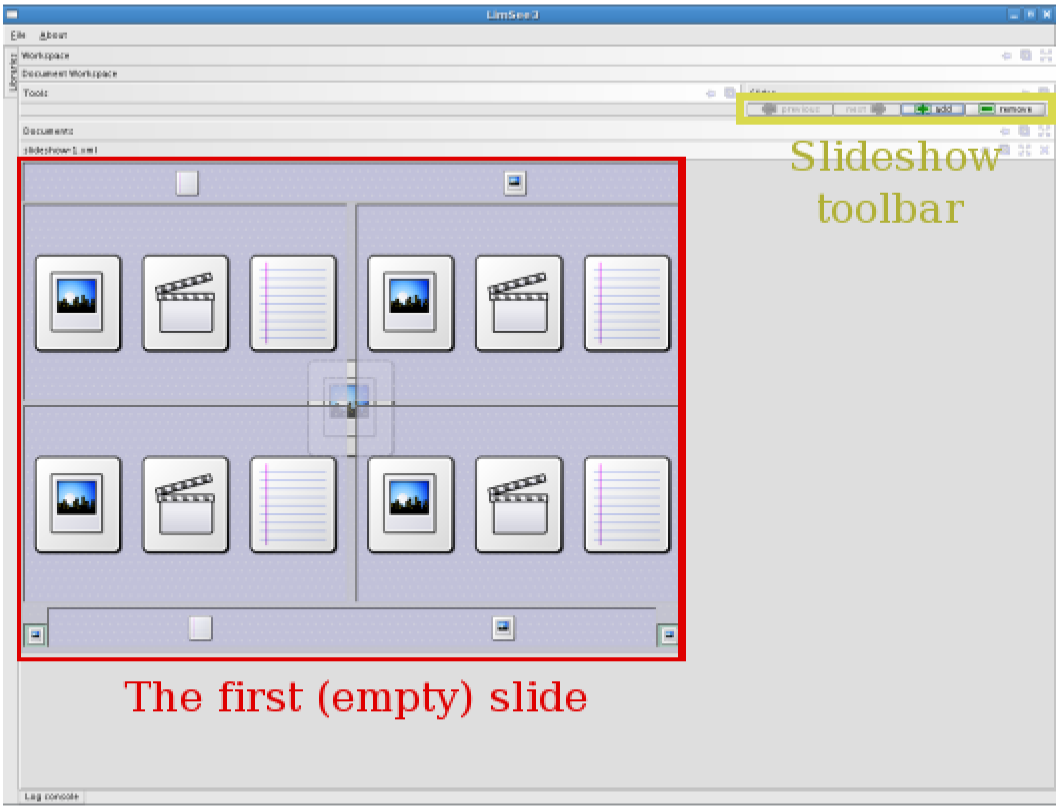


Fig.3: Opening an empty document

Let us have a closer look at the template visualization. It is composed of several rectangular regions with different icons inside. Every region represents one "hole" or "place-holder" in the template that can be filled with a media asset. The kind of assets that can be used for a particular region is specified by the template and is depicted by the icons. Thus, for instance, the four main regions can receive an image, a video or a text (fig. 4). We can notice an "image" icon in the middle of the slide that is partially hidden by other elements: it belongs to a region that is situated under the other regions and that features the background of the slide.

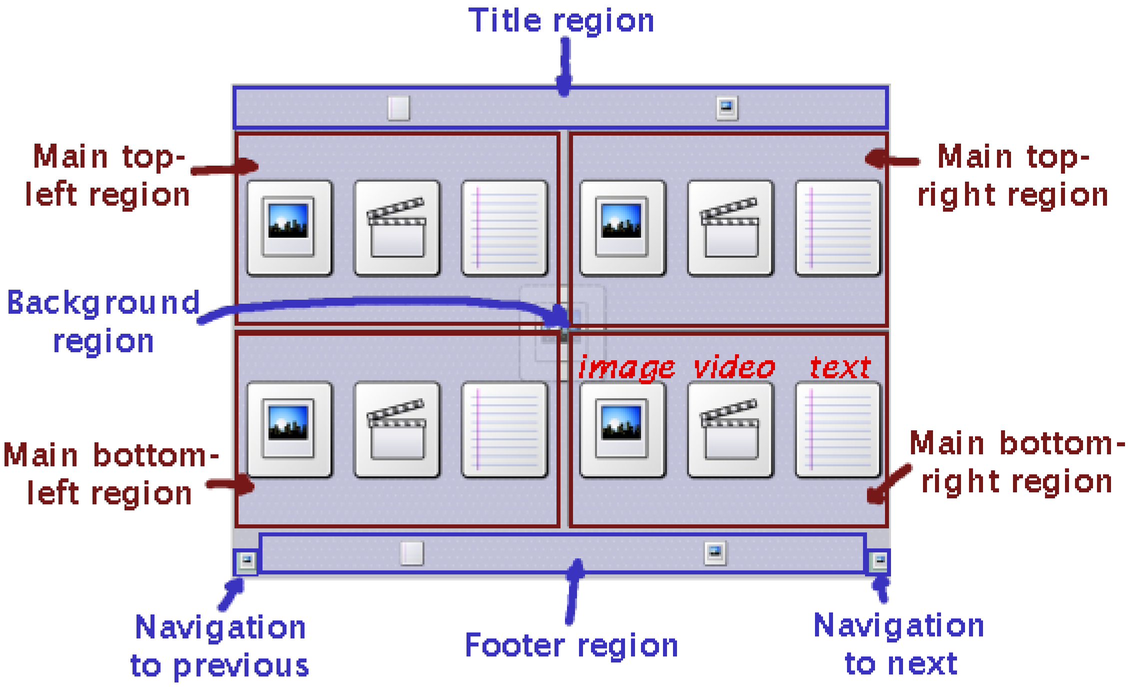


Fig.4: Detail of the slide template: blue regions are present on all slides, brown ones are specific to this particular sub-template

To instantiate the template, we need access to some media objects. In LimSee3, media objects are organized in asset libraries: we can open one (or several) libraries through the File menu. The library appears on the left part of the screen (fig. 5). Media libraries provide several ways of asset viewing (on figure 5, the snapshot view is used) and allow asset manipulations that we will see in more details in section 4.2.

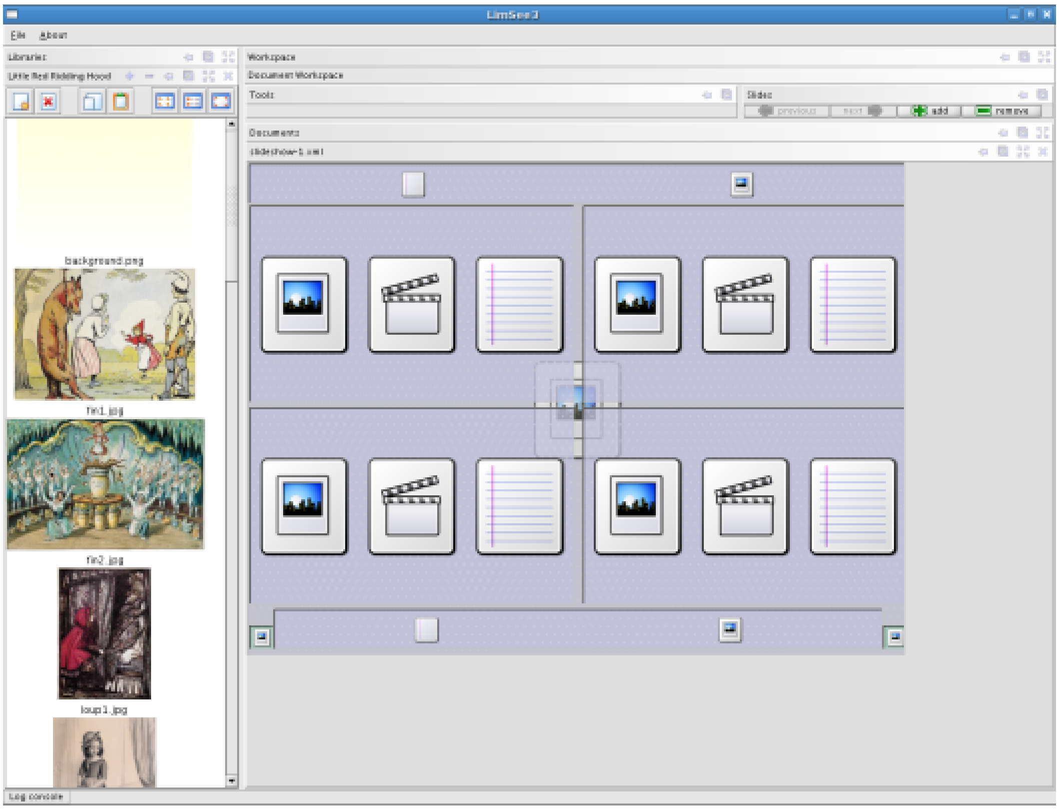


Fig.5: Opening a media library

It is easy to instantiate a region of the slide template with an asset from the library: the user can either use a mouse gesture (drag&drop) or a copy&paste action (those commands are accessible when right-clicking on a an asset, resp. a region). Figure 6 presents the result of instantiation of the background and the main top-left region.

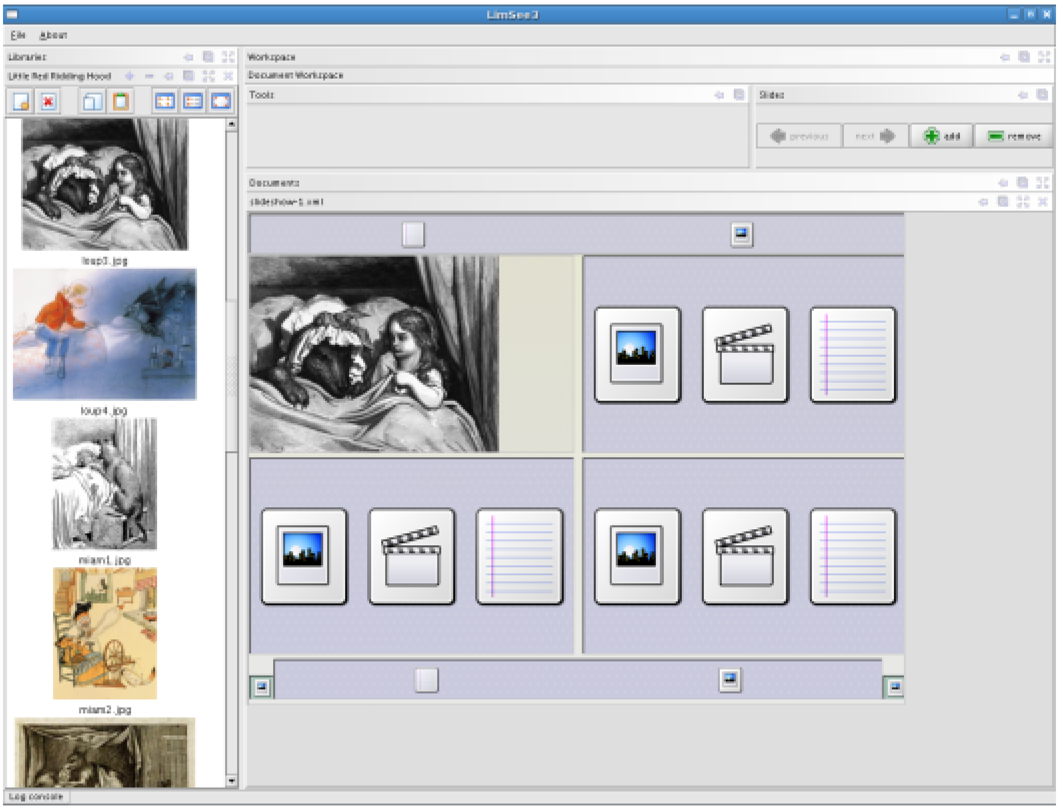


Fig.6: Instantiating from a media library

Text is a special kind of media, since it can be directly created by the user. This is why the right-click on an empty region proposes (besides pasting a media) to start a new text. On figure 7, we show how the slide title can be edited: a new toolbar appears that allows to modify usual text attributes, such as color, size, alignment etc.

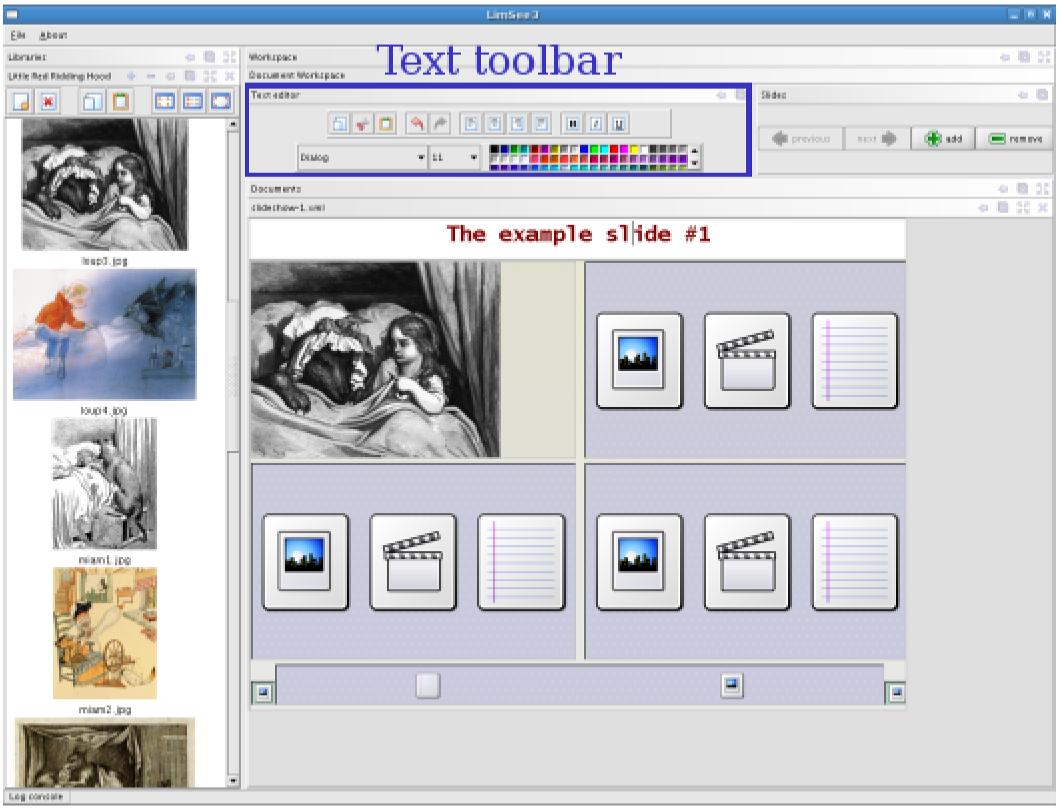
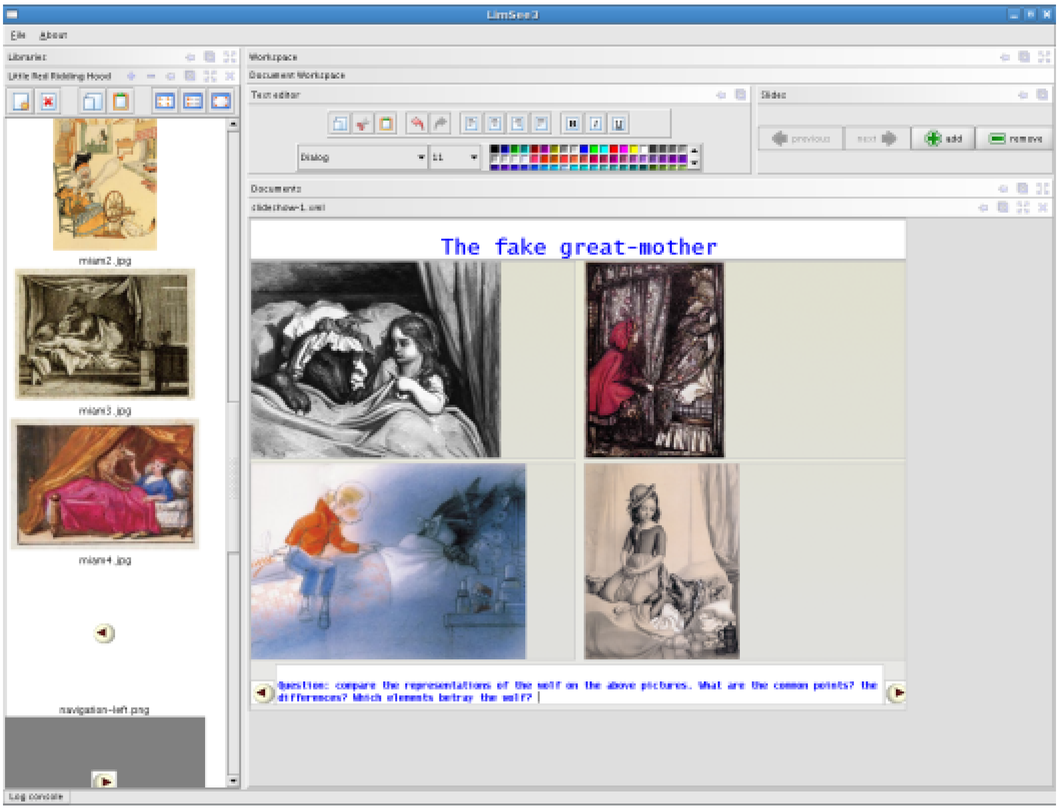


Fig.7: Editing a text

By combining the two approaches (instantiation from a library and direct editing), the whole slide can be created (fig. 8).

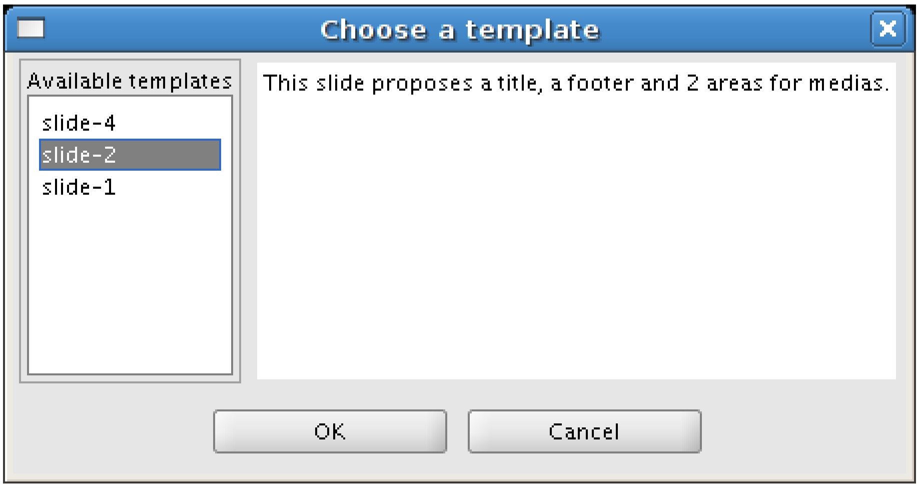


The fake great-mother

Question: compare the representations of the wolf on the above pictures. What are the common points? the differences? Which elements betray the wolf?

Fig.8: Example of a fully instantiated slide

Of course, a slide-show would contain more than one slide. One can create more of them thanks to the slideshow toolbar. When adding a new slide, the user can decide where to insert it (before or after the current slide) and then, (s)he is given a choice of the sub-template to use (fig. 9, fig. 10). Our slideshow template proposes three different slide layouts, differing mainly in the number of central regions.



Choose a template

Available templates

- slide-4
- slide-2
- slide-1

This slide proposes a title, a footer and 2 areas for medias.

Fig.9: Dialog for choosing a sub-template

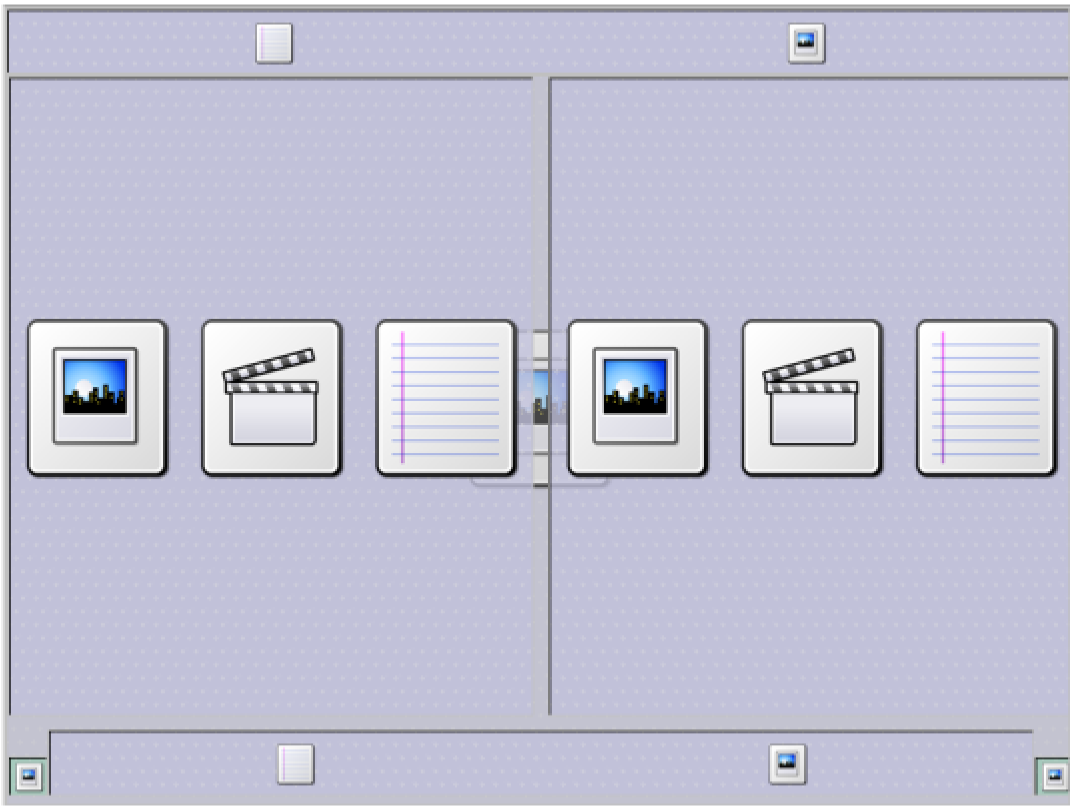


Fig.10: Detail of the "slide-2" sub-template

4.2 Media libraries

Media libraries allow users to organize the assets they (intend to) use into logical sets. In this section we propose a closer look on the functionalities that libraries provide.

Multi-modal viewing of assets: there are three different manners of displaying the contents of a media library. Besides the snapshot mode that we have used on the previous examples, there exists also a simple list mode (list of asset names) and a detailed list mode (list of detailed asset information). The actual mode can be chosen thanks to the three trailing icons of the library toolbar (fig. 11). Figure 12 presents the three different views of the previously used library.

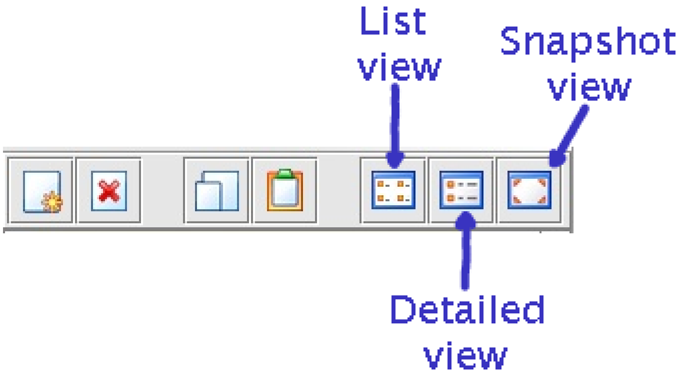
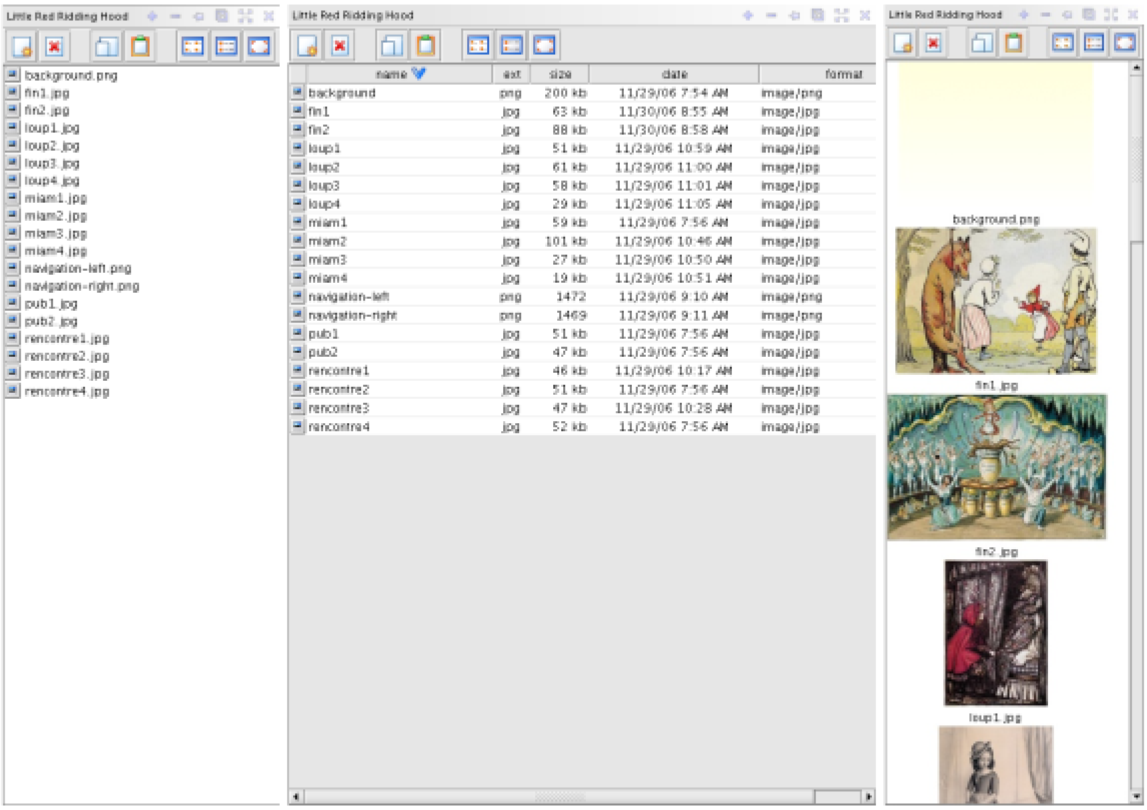


Fig.11:View switches



Little Red Ridding Hood

background.png
fin1.jpg
fin2.jpg
loup1.jpg
loup2.jpg
loup3.jpg
loup4.jpg
miam1.jpg
miam2.jpg
miam3.jpg
miam4.jpg
navigation-left.png
navigation-right.png
pub1.jpg
pub2.jpg
rencontre1.jpg
rencontre2.jpg
rencontre3.jpg
rencontre4.jpg

Little Red Ridding Hood

name	ext	size	date	format
background	png	200 kb	11/29/06 7:54 AM	image/png
fin1	jpg	63 kb	11/30/06 8:55 AM	image/jpg
fin2	jpg	88 kb	11/30/06 8:58 AM	image/jpg
loup1	jpg	51 kb	11/29/06 10:59 AM	image/jpg
loup2	jpg	61 kb	11/29/06 11:00 AM	image/jpg
loup3	jpg	58 kb	11/29/06 11:01 AM	image/jpg
loup4	jpg	29 kb	11/29/06 11:05 AM	image/jpg
miam1	jpg	59 kb	11/29/06 7:56 AM	image/jpg
miam2	jpg	101 kb	11/29/06 10:46 AM	image/jpg
miam3	jpg	27 kb	11/29/06 10:50 AM	image/jpg
miam4	jpg	19 kb	11/29/06 10:51 AM	image/jpg
navigation-left	png	1472	11/29/06 9:10 AM	image/png
navigation-right	png	1469	11/29/06 9:11 AM	image/png
pub1	jpg	51 kb	11/29/06 7:56 AM	image/jpg
pub2	jpg	47 kb	11/29/06 7:56 AM	image/jpg
rencontre1	jpg	46 kb	11/29/06 10:17 AM	image/jpg
rencontre2	jpg	51 kb	11/29/06 7:56 AM	image/jpg
rencontre3	jpg	47 kb	11/29/06 10:28 AM	image/jpg
rencontre4	jpg	52 kb	11/29/06 7:56 AM	image/jpg

Little Red Ridding Hood

background.png
fin1.jpg
fin2.jpg
loup1.jpg

Fig.12: Different views - a) simple list view, b) detailed list view (the window has been resized), c) snapshot view

Asset transfer: we have used a library as a source of assets, to instantiate a document. The same functionalities (drag&drop, copy&paste) are available for transfer of assets between libraries open at the same time, or for transfer of assets from a document to a library. The copy and paste commands are located in the central part of the library toolbar (fig. 13).

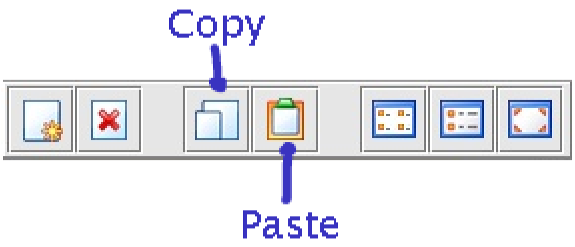


Fig.13: Copy and paste commands

Asset organization: finally, the leading toolbar icons (fig. 14) allow to import new assets and erase existing ones. Notice that these actions concern the current library only, not the (physical) assets themselves: erasing an asset removes it from the library, not from the disc; importing a distant asset does not have it downloaded.

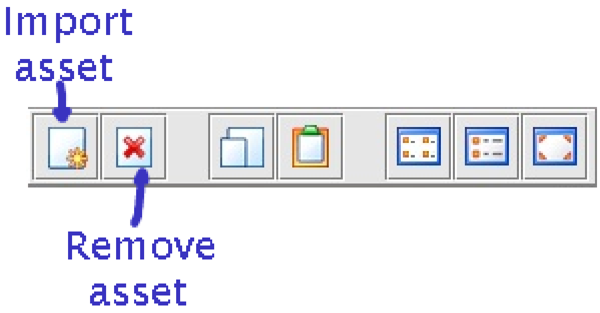


Fig.14: Import and remove commands

- *Importing assets* requires currently several steps from the user:
    1. When the "Import assets" button is clicked, a new dialog window opens (fig. 15).

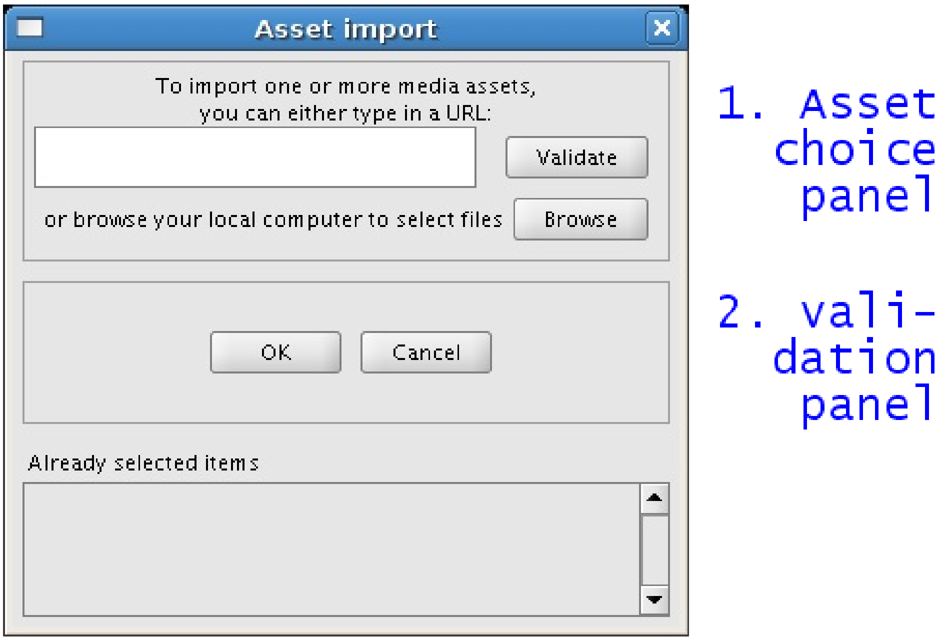
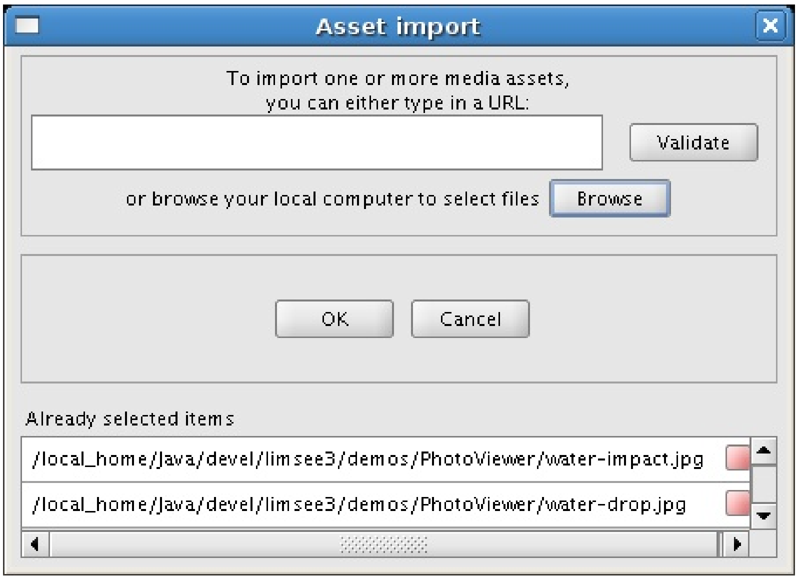

Asset import
To import one or more media assets, you can either type in a URL:
Validate
or browse your local computer to select files
Browse
OK
Cancel
Already selected items
1. Asset choice panel
2. validation panel

Fig.15: Import assets dialog window

2. Selection of asset(s) takes place in the top section of the window. User can either a) type in (or copy) the URL of an asset into the text field and validate it (with the Validate button) or b) browse the local disc(s) and select local asset(s).
3. As assets are selected, they appear in the bottom section of the window (fig. 16). At this stage, it is possible to remove them from the list by clicking on the trailing red button.



To import one or more media assets, you can either type in a URL:

or browse your local computer to select files

Already selected items

/local_home/Java/devel/limsee3/demos/PhotoViewer/water-impact.jpg

/local_home/Java/devel/limsee3/demos/PhotoViewer/water-drop.jpg

3.item panel

Fig.16: Situation after selecting two assets

4. Steps 2 and 3 can be repeated several times, so that multiple assets can be selected for import. When the list of assets is complete, the importing procedure can be started via the OK button in the middle section of the dialog (or totally aborted by the Cancel button). The import dialog closes.

- *Removing assets* is a straightforward process:
    1. User selects the assets to remove. This can be done for instance by clicking on the assets while holding the CTRL key, which allows to make a discontinuous selection, or the SHIFT key for a continuous one.
    2. User clicks on the "Remove assets" button. (S)He is prompted for a confirmation and then the assets are removed from the library.
- Notice however that library modifications introduced by importing or removing assets become permanent only after saving the library. Saving is proposed when the library (or the whole application) is about to be closed.

4.3. User interface customization

LimSee3 provides several possibilities to customize its user interface.

Docking: LimSee3 makes its users benefit from a full docking support. In general, a dockable element is a visual component with a dedicated title bar (fig. 17). The title bar is mouse-aware, so that the whole dockable component can be dragged&dropped into a different screen area. Mouse gestures also allow to organize dockables into tabs or to detach them altogether from the main work space. Moreover, the docking bar provides commands for minimizing and maximizing the dockables.

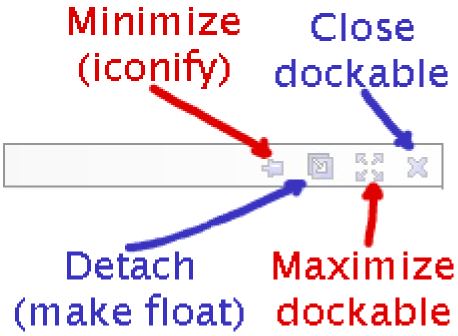
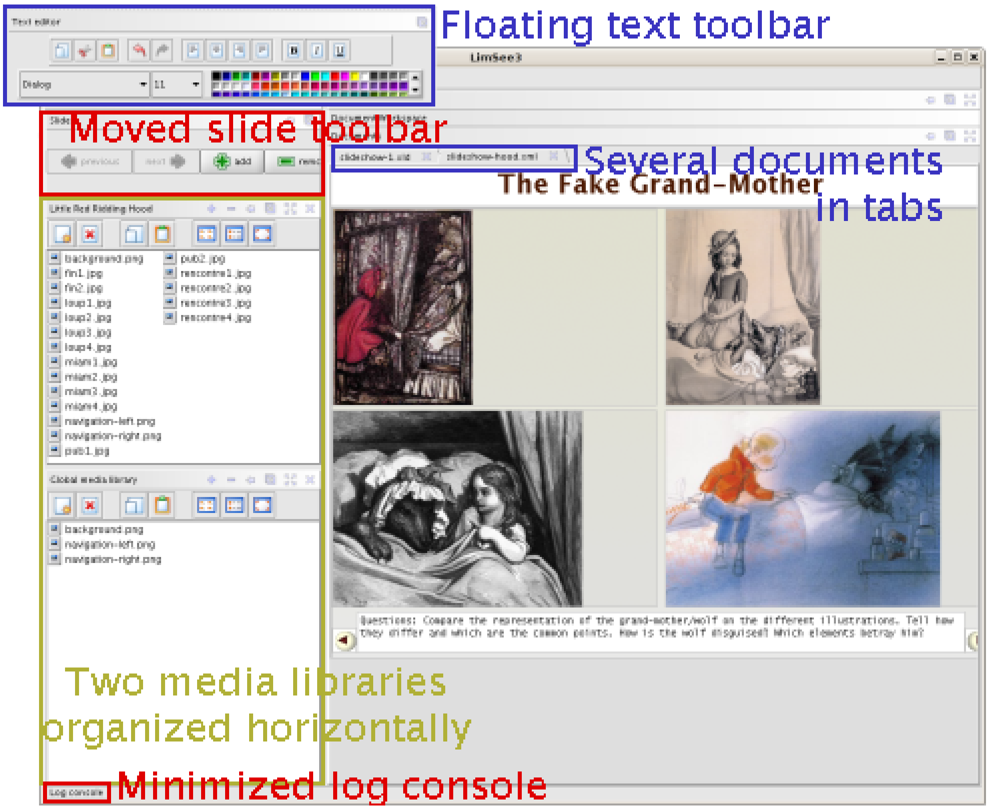


Fig.17: Docking bar

Figure 18 shows an example of a differently organized workspace. In particular, it presents: a floating (un-docked) component, tabbed components, a minimized (iconified) component and some moved ones.



Questions: Compare the representation of the grand-mother/wolf on the different illustrations. Tell how they differ and which are the common points. How is the wolf disguised? Which elements betray him?

Fig.18: Example of a different workspace organization

Accessibility: As a test, media asset libraries provide a support for accessibility. The "plus" and "minus" buttons on the docking toolbar allow to increase or decrease the size of the rendered elements, icons, texts etc. (see fig. 19) This functionality allows users to adapt their workspace to their eyesight ability. Progressively, all other interface components will integrate this scaling possibility.

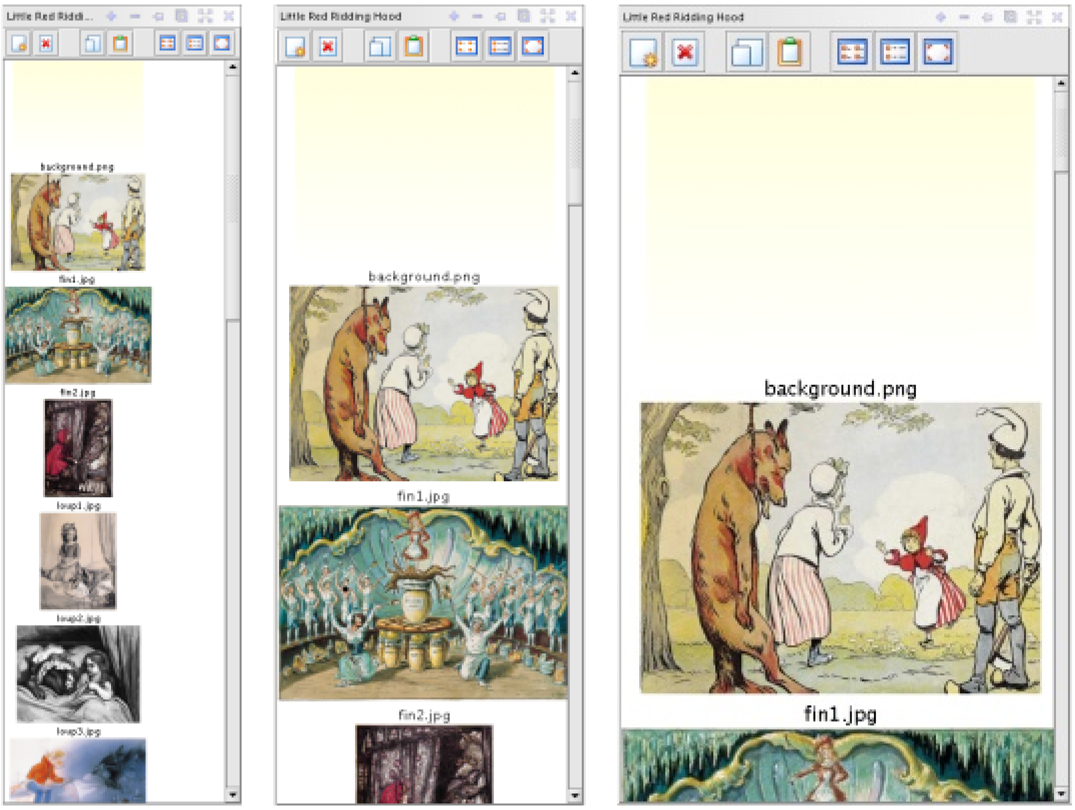


Fig.19: Scaled renderings of an asset library

5. References

[1] A. Boukottaya, F. Campoy Flores, R. Deltour, V. Quint, Ch. Vanoirbeek, I. Vatton, K. Zouba, *D.INF.01: Report on the design of extension mechanisms for creating templates, using templates for editing and customizing the user interface, and of extensions to be integrated in the information reuse tool (M3)*, 15 May 2006 (pdf, html)

[2] R. Deltour and C. Roisin, *The LimSee3 Multimedia Authoring Model*. DocEng2006 *ACM Symposium on Document Engineering*, 10-13 October 2006, Amsterdam, The Netherlands, pp. 173-175 (pdf)

[3] R. Deltour and A. Guerraz and C. Roisin, *Multimedia Authoring for CoPs*. TelCoPs'06, 1st International Workshop on Building Technology Enhanced Learning solutions for Communities of Practice, Crete, Greece, 2 October 2006, pp. 60-69 (pdf)

[4] A. Guerraz and C. Roisin and J. Mikáč and R. Deltour, *Multimedia Authoring for Communities of Teachers*. IJWLTT, International Journal of Web-Based Learning and Teaching Technologies, 2(3) (to appear in Summer 2007)

[5] J. Mikáč and C. Roisin and R. Deltour, *LimSee3 Document Model*. http://limsee3.gforge.inria.fr/public-site/docs/LimSee3-document-model.html, Internal PALETTE report, December 2006.

6. Appendix: Installation

6.1. LimSee3

The following points explain how to obtain and run LimSee3.

- **Prerequisite:** LimSee3 is written in Java. This means that for running the application, a Java virtual machine (version 1.5 or later) has to be installed on your system. On most systems, this is already the case; if not, a Java Runtime Environment can be obtained from Sun.
- **Download:** LimSee3 is distributed under the form of one executable archive, limsee3.jar. Download this file and store it somewhere on your computer. Do not unpack the archive.
- **Running:**
  - On Windows systems, a double-click on the `limsee3.jar` file is generally sufficient for running the application. If this is the case, no more steps are required from you. Otherwise, download the limsee3.bat script in the same folder as `limsee3.jar` and double-click that script to run LimSee3.
  - On other systems, either invoke "`java -jar limsee3.jar`" from a console, or download the limsee3.sh script which does it for you.

This installation procedure, as well as the current version of LimSee3 are available at http://limsee3.gforge.inria.fr/public-site/software.html.

6.2. Slideshow example

For convenient testing, we have bundled the resources used in all above examples into a ZIP archive: http://limsee3.gforge.inria.fr/public-site/example.zip. Should you be interested in using these resources, please download the archive file and decompress it **into the same folder** you have installed LimSee3 to. You can erase the archive after unpacking it.

Now your LimSee3 directory contains a number of images (the actual assets) and two XML files: `slideshow.xld` (the LimSee3 document) and `slideshow.xll` (the associated media library). If you open both in LimSee3 (menu *File* > *Open document* and *File* > *Open media library*), you can create a slideshow on your own.