

# Regia: a Metadata Editor for Audiovisual Documents

1	Introduction.....	1
2	The metadata AVDocument structure.....	2
3	Terminology.....	3
3.1	The AVDocument.....	3
3.2	The GUID.....	3
4	Opening a AVDocument.....	3
4.1	Opening a AVDocument from the filesystem.....	3
4.2	Opening a document from the ECHO Database.....	4
5	Browsing/Editing the Bibliographic section AVDocument.....	5
5.1	Structure of the editor window.....	5
5.2	Browsing the AVDocument.....	6
5.3	Editing AVDocument.....	7
5.4	Setting the Assigned/Unassigned flag of the fields.....	8
5.5	Deleting an object and its children.....	9
5.6	Adding a new child entity.....	9
6	Browsing/Editing the Multimedia section AVDocument.....	9
6.1	Browsing a Version.....	9
6.2	Expression Tool.....	10
6.3	Introduction.....	10
6.4	Browse/edit metadata.....	11
6.5	Playing the Expression.....	11
6.6	Read the Video Transcript.....	12
6.7	Browse the Expression segmentation.....	12
6.8	Create, modify, and delete, segments of the Expressions.....	12
6.8.1	Create.....	12
6.8.2	Modify.....	12
6.8.3	Delete.....	13
7	Saving the AVDocument.....	13
8	Creating a new AVDocument.....	13
9	Exporting/Importing.....	13
10	CrossLanguage issues.....	13
11	Installing and configuring Regia.....	14
11.1	Installation.....	14
11.2	Configuration.....	14

## 1 Introduction

Regia is a prototype application for manual editing metadata for audiovisual content developed in the ECHO<sup>1</sup> project [1]. Regia allows the user to manually edit textual metadata and to hierarchically organize the segmentation of the audiovisual content. An important feature of this metadata editor is that it is not hard wired with a particular metadata attributes set; for this purpose the XML schema

---

<sup>1</sup> ECHO was funded by the IST programme of the European Commission under the V Framework; the project began in February 2000 and was completed in March 2003.

of the metadata model is used by the editor as configuration file. Moreover, Regia can be attached to an external program of shot detection which is also able to produce MPEG-7 descriptions of the keyframes corresponding to the detected scenes.

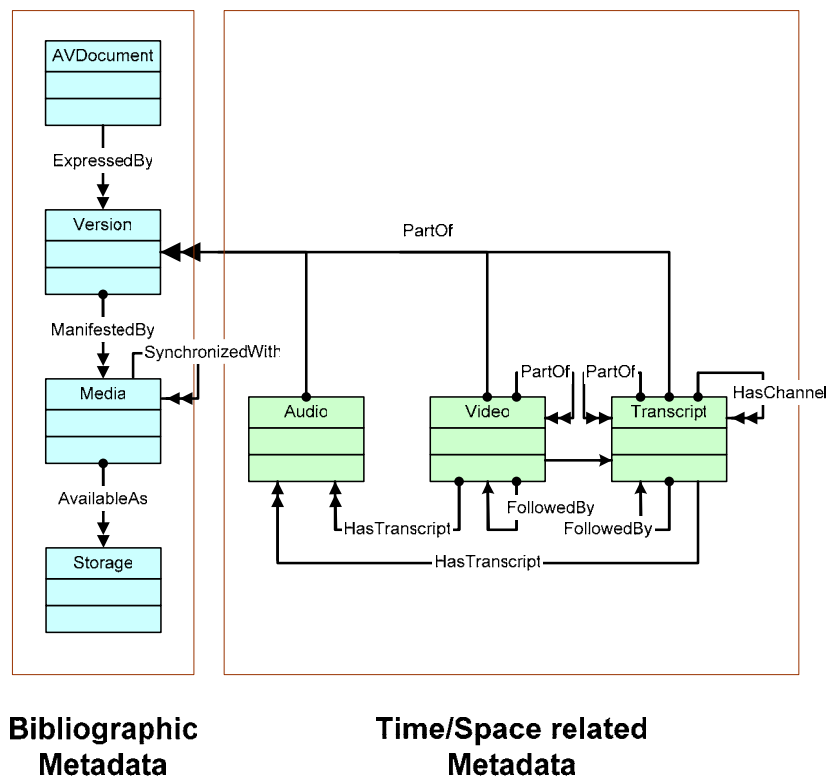


Figure 1 - Schematic representation of the Echo Metadata Model.

## 2 The metadata AVDocument structure

The metadata model of ECHO is based on the IFLA/FRBR model [2][3], a general conceptual framework used to describe heterogeneous digital media resources. This model allows to take into account different levels of specialization of the same abstract document. For instance the abstract idea of audiovisual object “The Lord’s of the Rings” can be realized through different versions: “Original Version”, “Extended Edition”, etc. Moreover, the “Original Version” can be embodied in different formats, such as: DVD, VHS, DivX, etc. The metadata model is composed of four levels describing different aspects of intellectual or artistic endeavour: *AVDocument*, *Expression*, *Media*, and *Storage*. The entities of the model are organized in a structure that reflects the hierarchical order of the entities from the top level to the bottom one. Figure 1 shows a schematic representation of the ECHO Metadata Model. As it is possible to see, the metadata which belong to different classes comprised in the model, are logically divided in two sets *Bibliographic Metadata* and *Time/Space related Metadata*. This classification is also reflected by the Metadata Editor interface.

According to the IFLA/FRBR methodology the AVDocument entity is the most abstract one; it provides the general intellectual or artistic view of the document.. For instance, let us suppose we want to describe a document about the Berlin Olympic Games in 1936. An AVDocument object will represent the abstract idea of the documentary film on the Games. A number of instances, of the abstract entity Version, could represent different language version of the same film, e.g., versions in Italian or in German. However, the Version entity does not represent any specific implementation of the film. This aspect can be represented by means of the manifestation level. For

instance, a Media object could represent a digital realization of the document in MPEG format. More than one manifestation of the same Version, e.g. MPEG, AVI, etc., may exist.

Nevertheless, the Media object does not refer to any physical implementation. For instance, the MPEG version of the Italian version of the Games can be available on different physical supports, each one represented by a different Storage object (e.g., videosever, DVD, etc).

The three entities of the internal metadata, which are *Video*, *Audio*, and *Transcript*, belong to an instance of the Version entity by means of the **PartOf** relationship. These entities are identified in the IFLA/FRBR as *Expressions* (since formally they actually belongs to the Expression level). This means that for each instance of the Version entity we can associate a Video, an Audio, and a Transcript Object, which correspond to the entire document from the temporal point of view. The most important aspect to remark is that each of these entities has a one-to-one **PartOf** relationship to itself. This relationship permits to associate a document sub-segment to the parent segment which contains it.

For instance, a Video object, which represents a specific scene Film document, is part of the Video object which represents the whole document. This allows to recursively partition the Video in sequences, scene, shots, etc. The relationship **FollowedBy** can be used to express the temporal succession of the document parts and the type of transition between a part and the following one (cut, fading, etc.).

## 3 Terminology

### 3.1 The AVDocument

Throughout this document, for simplicity, we refer to an instance of the model as “AVDocument”, which is composed of an instance of the AVDocument entity, and one or more instances of the other entities (Version, Video, etc). Furthermore, we use the word “AVDocument root” to refer to an instance of just the AVDocument entity.

### 3.2 The GUID

By means of the term GUID we refer to the unique identifier of an object of the model. When the object documents are stored in the disk (FileSystem) the GUID is the complete path filename of the XML document representing the object. When the object is stored in the Database it is a hexadecimal number of 64 digits.

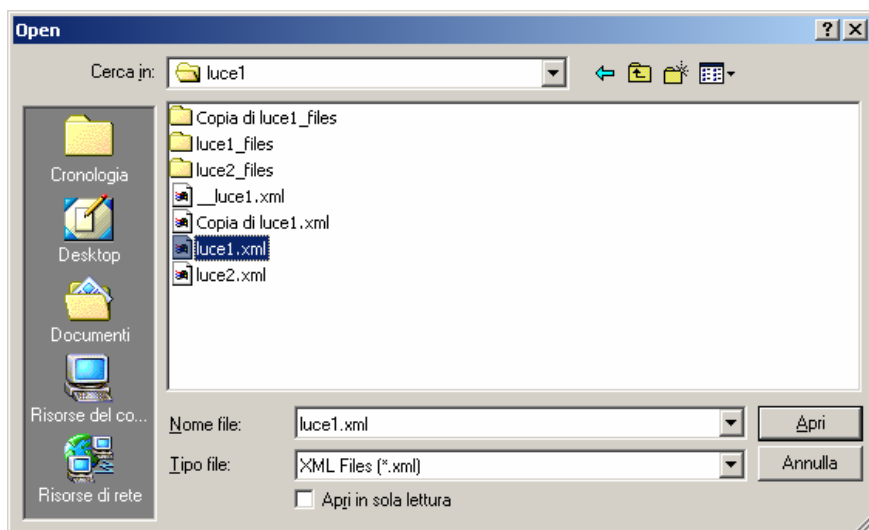
## 4 Opening a AVDocument

### 4.1 Opening a AVDocument from the filesystem

When an AVDocument is stored in the disk, it is composed of a certain number of XML files, each one correspondingly to an instance of an entity of the model, and a certain number of multimedia files (such as MPEG or AVI) referenced by the *Storage* objects.

In particular, there is one XML file representing the instance of the AVDocument root and an XML file for each instance of entities of the document which stored in a subdirectory located on the same location of the AVDocument root file. The name of this subdirectory is “xxx\_files”, where “xxx” is the name of the AVDocument root file. For instance if the document is called *doc1.xml* in the same directory where there is a directory called *doc1\_files*, which contains the rest of the XML files, that belong to the AVDocument.

In order to open an AVDocument, go to the menu File→Open and select a root XML file of the AVDocument which you wish to open (see Figure 2).



**Figure 2 - Opening an AVDocument from the disk.**

*Tip: it is possible to open an entity of the AVDocument selecting a file XML in the subdirectory of an AVDocument. In the case Regia will open the whole document and will directly open a dialog window for editing the selected object.*

#### **4.2 Opening a document from the ECHO Database**

The document stored in the Database can be open directly from the system Retrieval Interface, see Figure 3. Click on the **AV** icon and type on the Find text box the string of you wish to search for. Click on the title of one of the AVDocuments of the result set (left bottom of the window) and then click on the *Cataloging* icon (on the top). The editor will be launched and the corresponding AVDocument will be loaded. Regia communicates with the Echo Datamanager in order the retrieve the metadata of the selected AVDocument. The network address of the host of the Echo Datamanager to use can be set in the *ConnectionDialog*; for further details see the Section 11.2.

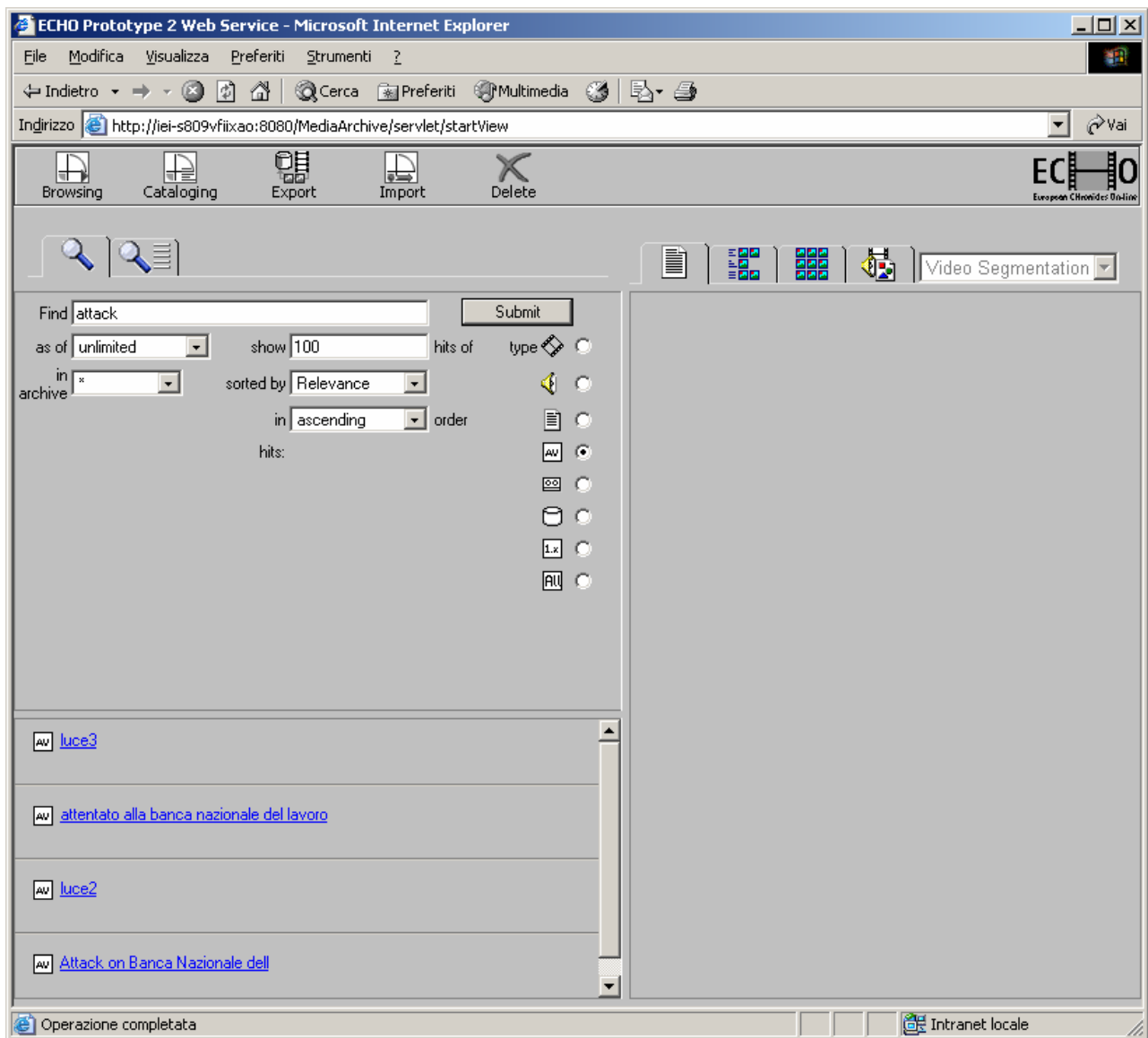
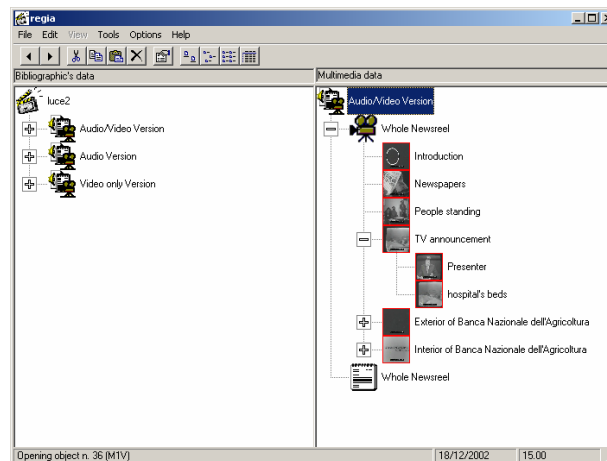


Figure 3 - Echo retrieval interface.

## 5 Browsing/Editing the Bibliographic section AVDocument

### 5.1 Structure of the editor window








Figure 4 shows the structure of the main window of the editor. The left side of the window is thought for containing the object of the bibliographic entities; while the right side of the window contains the multimedia objects (i.e., the time/space related metadata).

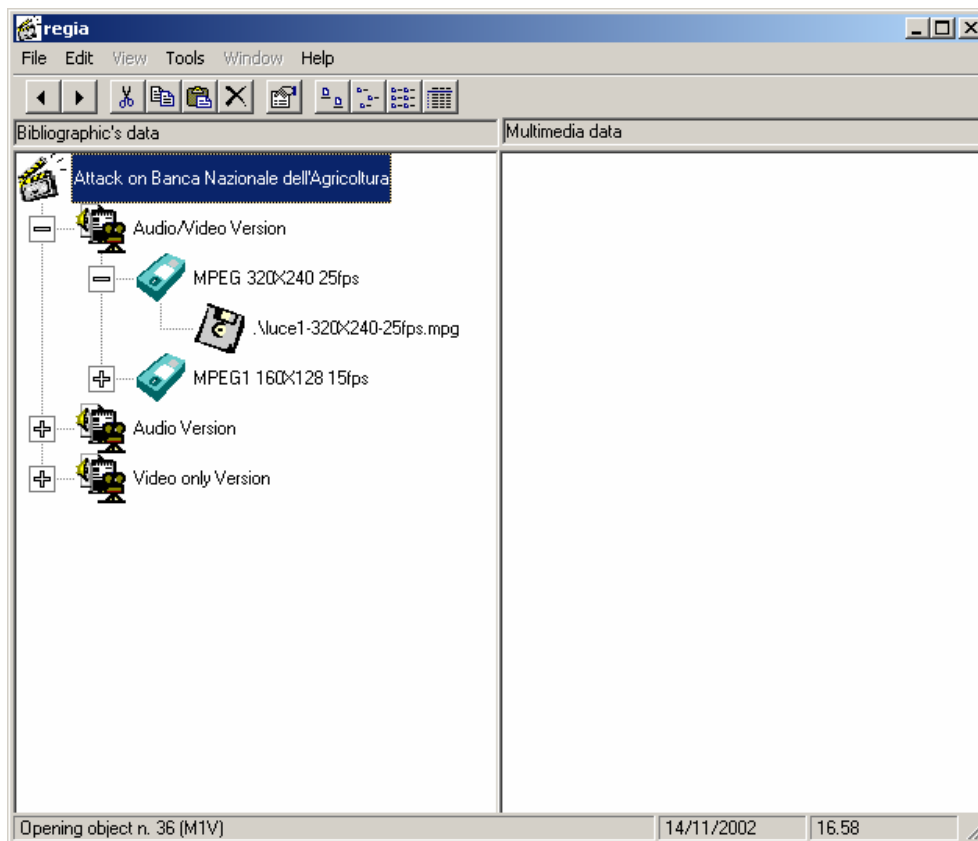


**Figure 4 - The editor structure.**

## 5.2 Browsing the AVDocument

Once you have opened an AVDocument it is possible to browse its structure just double clicking on the icons on the left side of the main Regia windows. The structure is organized as a tree and each node represents an entity object of the AVDocument structure. Each entity is associated to a different icon; the icons are the followings (see also Figure 5):

- AVDocument root..... 
- Version..... 
- Media..... 
- Storage..... 
- Video..... 
- Audio..... 
- Transcript..... 



**Figure 5 - Main window of Regia**

### **5.3 Editing AVDocument**

In order to edit the metadata fields of any object it is sufficient to select its corresponding icon and to click on the right button of the mouse. It will appear a menu; click on the “Edit Metadata...” entry. A dialog window containing all the metadata of the selected object will appear. Figure 6 shows an example of the edit window.



**Figure 6 - The edit window.**

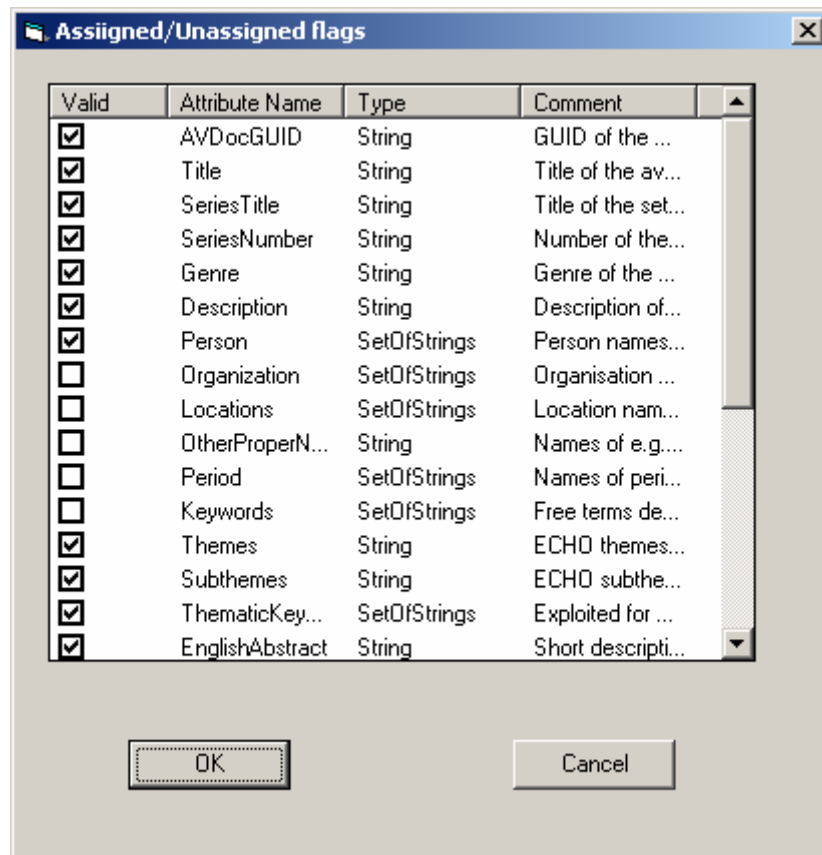
At this stage it is possible to modify any metadata field of the corresponding object. For convenience, the fields are grouped in different tabs windows called “set 1”, “set 2”, etc. Note that, some fields are read-only, (for instance the GUIDs). Clicking on the button “Cancel” the modifications will be discarded. Clicking on “OK” the modifications will be kept.

#### ***5.4 Setting the Assigned/Unassigned flag of the fields***

Sometimes, only a subset of the available textual fields contains meaningful values. However, the value contained in the corresponding XML element may be not able to indicate a null or not assigned field. For instance a boolean element must be **true** or **false** no other value is accepted by the XML parser. The solution adopted is to allow the XML element of an “unassigned field” to not appear in the document, which, for convenience, it is not made accessible for editing on the interface. In order to change the status of a field from **unassigned** to **assigned** and viceversa, we use the separate dialog window of Figure 7, which appears selecting an icon object by the right



button of the mouse, and clicking on “Edit Assigned/Unassigned flag...” If the checkbox is checked then the corresponding field is **Assigned** (and will appear in the edit window) if checkbox is unchecked then the corresponding field is **Unassigned** (and will not appear in the edit window).



**Figure 7 - The Assigned/Unassigned dialog.**

If the checkbox is checked then the corresponding field it is Assigned (and will appear in the edit window) if checkbox is unchecked then the corresponding field is Unassigned (and will not appear in the edit window).

### **5.5 Deleting an object and its children**

By selecting an object icon and typing the “Del” button on the keyboard it is possible to delete an object and its children.

### **5.6 Adding a new child entity**

It is possible to add a new child object by selecting its parent, clicking on the right button of the mouse, and then clicking on “Add child...”. It will appear an input box asking for the title of the new object. Note that, for the Storage entity the title corresponds to the content of the **collocation** field.

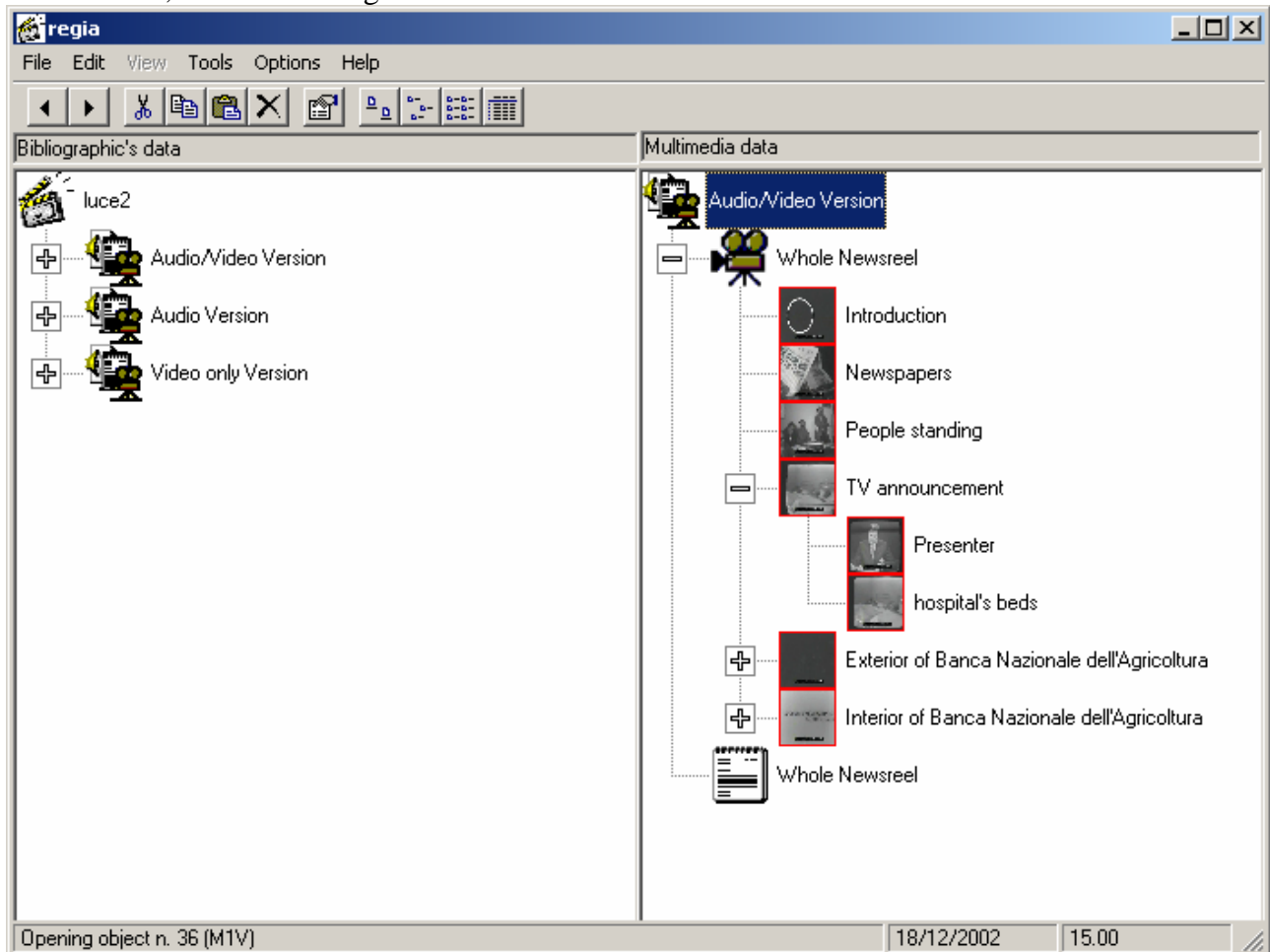
## **6 Browsing/Editing the Multimedia section AVDocument**

### **6.1 Browsing a Version**

As explained in Section 2 an AVDocument can maintain one or more Version objects. By clicking a Version icon of the Bibliographic data tree, it is possible to browse the corresponding Expressions structure composed of Video, Audio, and Transcript objects. In particular, a Version object can be

associated to exactly one Main Video Expression, one Main Audio Expression, and one Main Audio Expression. A Main Expression represents in general the whole movie document and its children its temporal segmentation in term of sub-Expression. For Instance, the Main Video represents the whole Video Track of the indexed movie; while, its children represent the scenes in which it has been partitioned.

By clicking on a Version icon the structure of the Version object will be shown on the Multimedia data section of Regia (on the right side). It is possible to browse the Version by means of the tree-view control, as shown in Figure 8.



**Figure 8 - The Version structure.**

Note that, each Expression and sub-Expression has its own distinct setting of metadata values, which can be independently edited. Therefore, the browse, creation, and update, of the Expressions and sub-Expressions works as explained in the previous sections.

**Tip:** it is possible to delete an Expression (and its sub-Expressions) directly by deleting the corresponding XML file. This operation does not damage the AVDocument consistence.

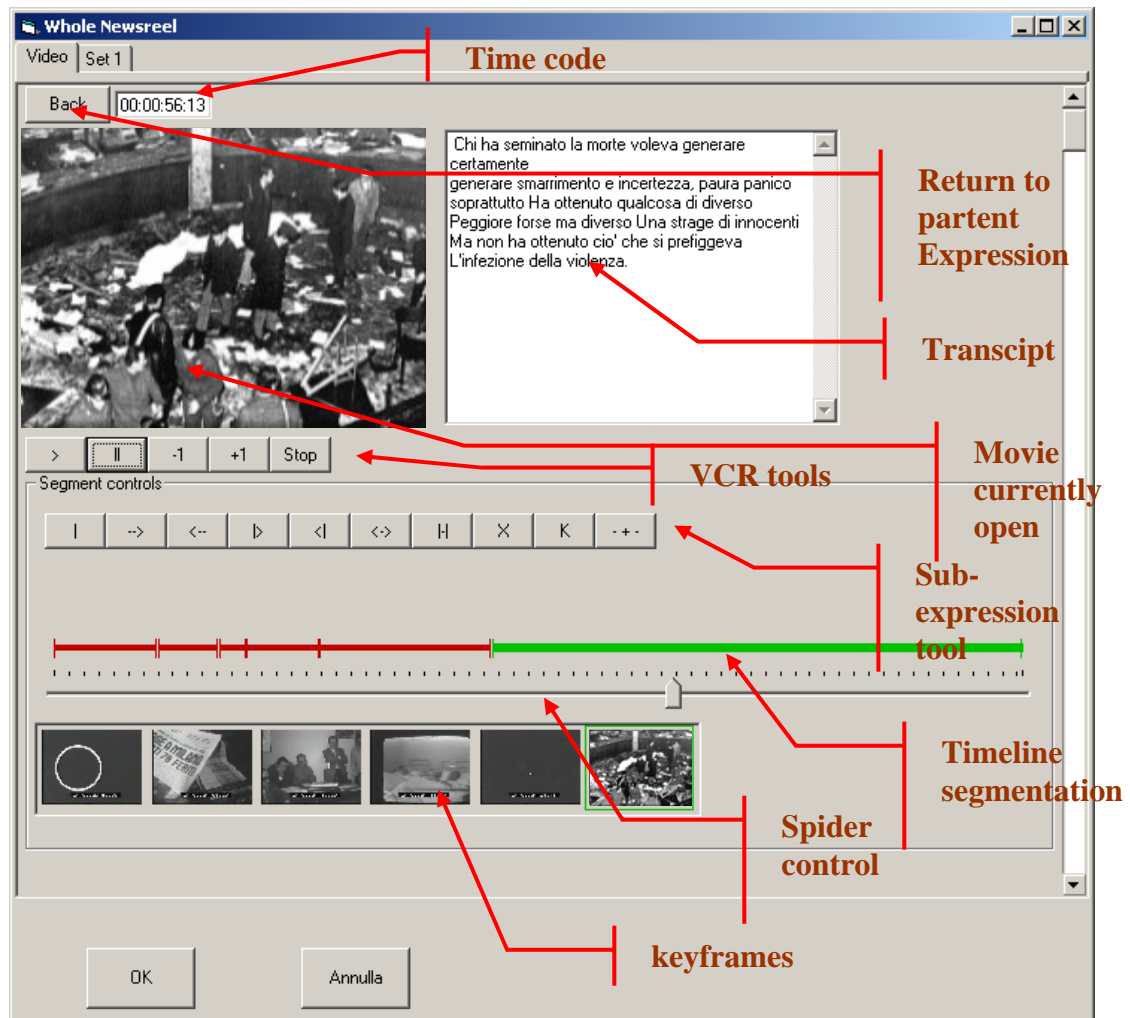
## 6.2 Expression Tool

### 6.3 Introduction

By means of the Expression Tool (see Figure 9) it is possible to:

1. browse/edit the metadata associated to the Expressions.
2. play, stop, pause, step frame by frame, the movie associated to the current Expression object.

3. read the transcript (if any) synchronized during the Video playing.
4. browse the Expression segmentation, by means of a coloured timeline control and the keyframes corresponding to the Expression segments.
5. create, modify, delete segments of the Expressions.



**Figure 9 - The Expression Tool.**

## 6.4 Browse/edit metadata

Each Expression object has its own metadata, and exactly as for the other entities, there is a metadata edit window (called Expression Tool), which is divided in several tabs. The first tab window shows the hierarchical Expression segmentation. The additional tab windows allow the user to edit the textual metadata associated to the Expression.

## 6.5 Playing the Expression

When the Expression window is opened, the corresponding video is loaded and positioned to the start time of the Expression. On top of the window where the movie is played, the current timecode is indicated. The format of the timecode is Hours:Minutes:Seconds:Frames. Underneath the movie a set of typical Video Recorder buttons allow the user to navigate through the movie. Moreover, by means of the timeline cursor it is possible to reach any position of the movie inside the interval defined in the current Expression.

## 6.6 Read the Video Transcript

During the video playing on the right side of the Expression window there is a text box where the speech Transcript (if any) associated with the Expression is shown. The transcript extracted from the movie is shown synchronized with the movie speech.

## 6.7 Browse the Expression segmentation

By means of the coloured timeline control, the user can navigate the Expression and visit its child Expressions (which are Expression objects, as well). In order to exploit this feature of the tool, it is worth understanding the behaviour of the slide segmentation. Figure 10 explains this characteristic.

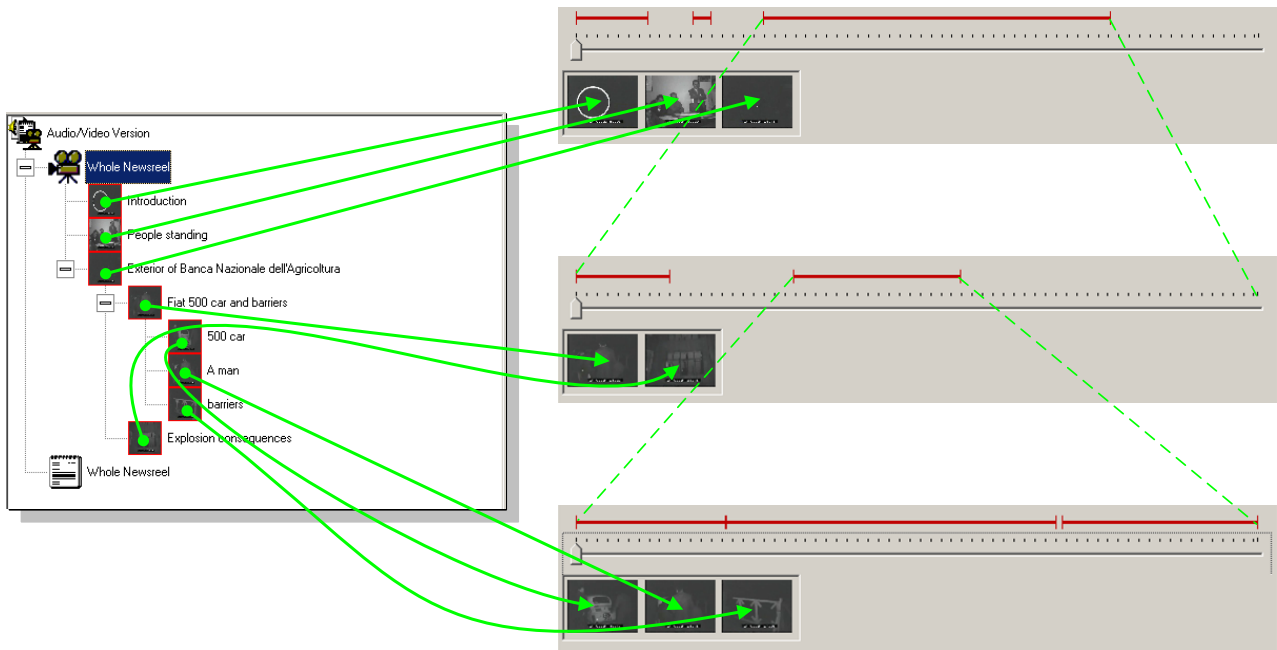


Figure 10 - Segment and keyframes association.

The figure shows the correspondences between the Video on the hierarchical tree view and the keyframes associated with the sub-Expressions on the Expression Tool window. By double-clicking on the keyframe corresponding to the segment or by double-clicking the segment of the timeline bar it is possible to visit the corresponding sub-Expression. In order to return to parent Expression, click on the “Back” button on the top-left of the Expression Tool window.

## 6.8 Create, modify, and delete, segments of the Expressions

### 6.8.1 Create

In order to create a new sub-Expression (a video shot for instance) by means of the Expression Tool, there are two ways of proceeding:

1. move the cursor of the timeline and place it at the start time of the shot you want to create, push the button “[|”. The timecode and a small line on the timeline start blinking. Move the cursor and place it at the end time of the new segment, push again the button “[|”.
2. Push the button “[|-]”. A dialog box will allow you to select the start/end timecodes boundaries of the new segment.

### 6.8.2 Modify

In order to edit a sub-Expression, use one of the following methods:

1. Push the button “|>” (“<|”). This will set the end (beginning) of the selected segment to the beginning (end) to the (preceding) following segment.
2. Push the button “<->”. A dialog box will allow you to modify the start/end timecodes boundaries of the selected segment.
3. By selecting the segments by means of the mouse button and pressing shift button, it is possible to select more than one segment. The selected segments are highlighted by means of a thicker line on the timeline, and the keyframes contained are shown in a coloured frame. By pushing the button “-+” all the selected segments are merged together. Note that, the new created segment will inherit the textual metadata of the first selected segment (highlighted in green).

### 6.8.3 Delete

Select a segment and push the button “X” or the key “Del” of the keyboard.

*Tip: moving the mouse over a segment of the timeline or over a keyframe, a TooTipText showing the title and the start/end time of the segment appears.*

## 7 Saving the AVDocument

All the changes made on the AVDocument as creation, update and deletion, are not stored permanently in the disk or in the Database until the user save it. Exactly as a textual editor it is possible to save the document (menu File→Save) or to save a new copy of the document by means of the “Save As” command (menu File→SaveAs...).

## 8 Creating a new AVDocument

In order to create a new AVDocument from scratch, click on the menu entry File→New. A new AVDocument will be created. Proceed creating the other entities of the AVDocument structure.

## 9 Exporting/Importing

These features are useful when you wish to save the document from the disk to the Database and viceversa. Suppose you have already opened a document from the disk, click on the menu item “File→Export to the Database” to copy the whole AVDocument in to the Database. In the same way, having opened a document from the Database, click on “File→Import to the Filesystem”, in order to copy the whole AVDocument into the disk.

When the export/import process starts, the editor asks the user for the new collocations for all the Storage present in the AVDocument. Usually, the collocation of a Video in the Database is a remote path as for instance:

`\\pc-echo\ma\data\48534D2D4C4F434154494F4E2D312E305B5530928B7CCBBAF100000004ADDE23`

where the last string of characters is the GUID of the movie file. The collocations of an AVDocument stored in the disk correspond to path of the movie files.

## 10 CrossLanguage issues

All the AVDocuments of the collection are classified on the basis of three categories: *Themes*, *Subthemes*, and *Thematic Keywords*. For each category a closed list of possible values has been specified. Moreover, these categories are organized in a tree structure: each Theme includes a disjoint set of Subthemes and each Subtheme includes a disjoint set of Thematic Keywords. Each AVDocument is then associated to exactly one Theme, one Subtheme and one or more Thematic Keywords as shown in Figure Figure 11. In order to enable the cross-language browsing/editing, the text associated to these categories is translated in five languages (English, Italian, Dutch, French, and German). For a specific AVDocument, Regia uses the English entry for actually storing the value of these three categories, but uses the current user language for presentation. In other words, the English is used as pivot language.

When the user changes the Theme field of an AVDocument, the editor resets the combo boxes of the Subthemes and Thematic Keywords fields. Moreover, when the user changes the Subtheme of an AVDocument, the editor resets the combo boxes of the Thematic Keyword fields.

The keywords list is stored in the file ***ThematicKeywords.xml*** located in the directory where Regia is installed. Do not edit the ThematicKeywords.xml file, since a misalignment between this file and the stored documents can deteriorate the consistence of the documents. If there is an inconsistency between the file ThematicKeywords.xml and the stored metadata, Regia shows an error message and does not allow the user to edit the AVDocument metadata fields. However, to get over this problem click on the menu item “Tools→Language→Disable CrossLanguage”; Regia will allow the user to edit all the CrossLanguage fields as usual free text strings. Be careful to use this feature, only when it is really necessary.

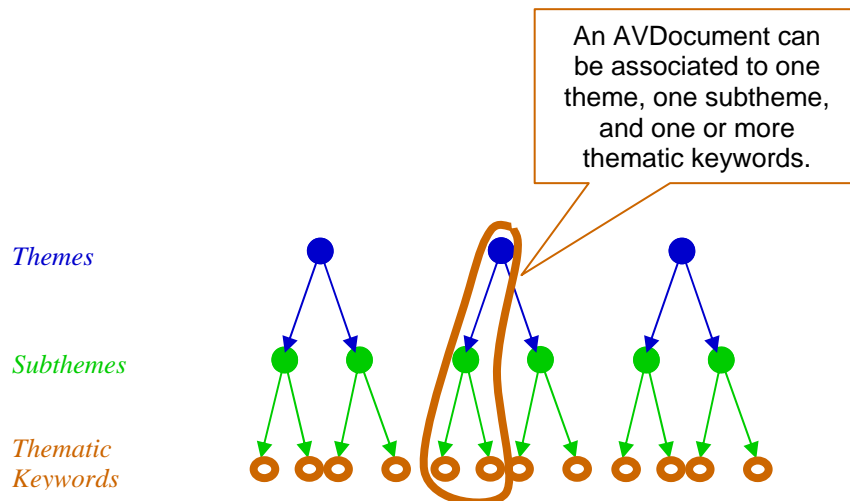


Figure 11 - The CrossLanguage hierarchical structure.

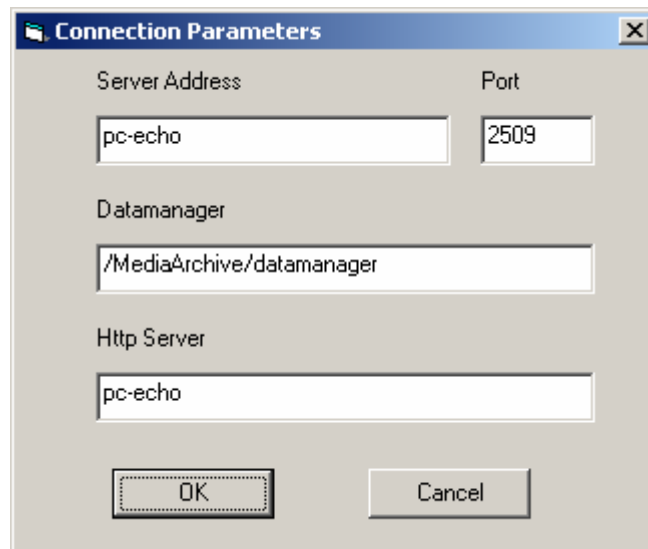
## 11 Installing and configuring Regia

### 11.1 Installation

Regia must be installed on a PC with MS Window 2000/XP by a user with Administrators privileges. Unzip the file *regia.zip* in a temporary directory, launch setup.exe, and follow the installation wizard.

### 11.2 Configuration

The only configuration issue is to network parameters for the connection to the Database (i.e., Echo Datamanger). Click on the menu item Options→Connection, the dialog of Figure 12 will appear.



**Figure 12 - Connection dialog.**

The meanings of the fields are the following:

- Server Address: it is the IP address of the host where the Echo Datamanager is running.
- Port: it is the number of the IP port used for the connection, typically is the port number 2509.
- Datamanager: path of the Echo Datamanager, typically is “/MediaArchive/datamanager”
- Http Server: IP address of the host where the Echo HTTP Server is running, typically it is the same as the Server Address.

## References

- [1] Echo: European CHronicles On-line. <http://pc-erato2.iei.pi.cnr.it/echo/>.
- [2] K.G. Saur Mnchen. Functional requirements for bibliographic records - final report, 1998. <http://www.ifla.org/VII/s13/frbr/frbr.htm>.
- [3] G. Amato, D. Castelli, and S. Pisani. A metadata model for historical documentary films. In José Luis Borbinha and Thomas Baker, editors, Proc. of the 4th European Conference ECDL, pages 328-331. Springer, 2000.