UNIVERSITÀ DEGLI STUDI FIRENZE

PhD Program

Informatica, Sistemi and Telecomunicazioni
"*Telematic and Information Society*"

Instituted by the Italian University Consortium among
University of Florence

Cycle XXVI

Coordinator Prof. Dino Giuli

Models and tools for aggregating and relating audiovisual contents for education and entertainment

ING-INF/05

Ph.D Candidate	**Tutor**
Dr. Marco Serena	Prof. Paolo Nesi
____________________	____________________
(firma)	*(firma)*

Coordinator

Prof. Dino Giuli

____

(firma)

Years 2010/2013

Acknowledgements

I want to thank Prof. Paolo Nesi who support me in these three years of research, and all the professors, researchers, people and friends of DISIT Lab with whom I've worked.

I also thank all the ECLAP Partners with whom I've spent good times during the three years of the project.

Marco Serena

Contents

1.Introduction

In the last few years, the society of information has been permeated by new ways of fruition of multimedia contents on the web. The attention of users is more focused on content oriented web sites, several forms of content aggregation are offered in the context of social and best practice networks. In most cases, the content aggregation forms allow to collect, share, organize, and make accessible digital contents in order to make easier for users the fruition, especially in cultural heritage and educational environments.

More recently, several web portals and social services are growing as *indexing portals/engines*, for example, collecting metadata of content (articles, video, etc.), indexing citations, indexing cultural heritage content. Those portals and services are facilitators for content access, while the real content items, the digital essences, are only referred and thus they are only accessible in the original portal of the content provider. This may happen in the digital libraries of IEEE and ACM, which in some cases index metadata and refer other collections via URLs.

Among *indexing portals*, Europeana (i.e., the European Digital Library, http://www.europeana.org) collects cultural heritage metadata coming from several institutions, universities, foundations, museums, schools of art, that represents a cultural heritage of the huge European history. Europeana portal collects only metadata, while content files are referred via some URL. These URLs refer to the original content owner and/or to the Content Aggregator, facilitating the collection. Thus every time the users perform a query on Europeana, the obtained results contain only classification metadata, and to access the real content file, an external link is provided. From that link the content file may be directly or indirectly available. Directly, when the URL refers to a specific digital resource file, and indirectly when the URL brings to an intermediate page or service gate to access at the content files; may be after a registration and/or a payment. Despite to difficulties in reaching resource files, the *indexing portals*, as well as Europeana, have a strong validity for educational purposes.

Another evidence of the general trend is the growing of the diffusion of Linked Open Data, LOD, by which semantic descriptions can be distributed and accessible, and may be a means to reach the URL of the effective resources [Bizer et al., 2009]. Moreover, the linked media [Open Media] could be an extension of this aspect in order to build more interactive mashups, also allowing updates to the data in Linked Data servers. For example, a resource might represent a video on the internet, and depending on how I access the video I want to get either the video itself or the structured metadata about the video (e.g. a list of RDF links to DBPedia for all persons depicted in the video).

Thus the above mentioned technologies may be enforced into learning tools, with social support to provide solutions for: commenting and rating content, adding annotations [Kahan et al., 2001], organizing content in play lists, establishing citations, synchronizing audiovisual, [Huang and Hu, 2000], associating annotation to time line [Schroeter et al., 2003], etc.

The approach of linking content is valid for the leveraging the access and diffusion, while presents some problems in managing aggregations. They should be put in the hands of students/users to collaboratively prepare their works and elaborations. These aspects may leverage the educational environments to become richer in term of resources, more flexible and efficient in their purposes. Those solutions may give to teachers and students a new ways of learning, self aggregating and organizing contents and assigning a semantic depending on the medium chosen.

Also the way media usage for entertainment and edutainment is rapidly undergoing a range of transformations. New formats and content fruition modalities are appearing. Also on television, new ways of interactions with users have been proposed, through multi stream TV programs especially with the introduction of web and IPTV, synchronized multiples views/streams of the same event, and multiscreen/device experiences. In Internet, social networks and web sites are distributing videos (such as YouTube, Vimeo, etc.), providing support for nonlinear play of audio visual: augmenting the content under play with links and connections with other content, e.g., providing suggestions. The Internet capability of providing links of navigating and thus the virtually unlimited number of combinations/paths are very attractive to the users for both entertainment and learning, since it allows creating your own stories and experiences. Thus, many web social media and based applications are proposed, where the user can synchronously play video and slides, can see a video and related chatting channel, can see a video and jump to another audiovisuals, can share content (audio visual, comments, votes, tags, etc.). Moreover, social networks are widely used by many educational institutions as content delivering networks facilities, even if they do not cope with pedagogical and didactical points of views due to their lack of annotation structure, definition of relationships, formal model for classification, aggregation, composition, etc. Indeed, with the popularity and the explosive growing of audio visual data on internet and the omnipresence of large-scale multimedia database, any efficient access to them is becoming essential as a first step to provide educational tools.

So that, there are several education and training cases in which multi-camera view is a traditional way to work: performing arts and news, medical surgical action, sport actions, play instruments, speech training, etc. In most cases, users need to interact to those multi camera and multi audiovisual to create among their segments relations and annotations with the purpose of: comparing actions, gesture and posture; explaining actions; providing alternatives, etc. Most of the present solutions are based on custom players and/or on specific applications that constrain to create custom streams from server side. Thus leading to restrictions on the user activity on dynamically establishing other relations and accessing to the lessons. Web based solutions would be more appreciated and are complex to be realized for the problems related to the video desynchronizations.

This thesis has been developed inside a European project ECLAP [ECLAP].

ECLAP is the European Collected Library of Performing arts (http://www.eclap.eu), a collaborative environment to produce enriched content and metadata for content collections that are posted on Europeana in terms of EDM and made accessible as LOD (Linked Open Data). ECLAP is a network of 35 prestigious performing arts institutions, and it is used by them for education and research goals. ECLAP partners provide content in 13 different languages and mainly come from central Europe, plus Chile, South Africa, Russia. ECLAP present about 170.000 content elements, ranging from video, audio, documents, 3D, images, braille, etc., including performance, premier, libretti, scores, pictures, posters, manual designs, sketches, etc.

In this thesis, aggregation forms in ECLAP focusing the attention on MyStoryPlayer/ECLAP solutions are presented. The major contributions to the state of the art are related to:

- the semantic model to formalize the relationships and play among audiovisual determining synchronizations,
- the model and modality to save and share user experiences in navigating among lessons including several related and connected audiovisual,
- the design and development of algorithm for shortening the production of relationships among media,
- the design and development of the whole system including its user interaction model, and
- the solution and algorithm to keep limited the desynchronizations among media in the presence of low network bandwidth.

The solution proposed has been developed for and it is in use of ECLAP (European Collected Library of Performing Arts) for accessing and commenting performing arts training content. The thesis also reports validation results about performance assessment and tuning, and about the usage of tools on ECLAP services. In ECLAP, the users may navigate in the audiovisual relationships, thus creating and sharing experience paths. The resulting solution includes a uniform semantic model, a corresponding semantic database for the knowledge, a distribution server for semantic knowledge and media, and the MyStoryPlayer Client for web applications.

This thesis is organized as follows:

Chapter 2 reports an overview of related work to put in evidence the state of the art of the major problems at the ground of the proposed systems and solutions. The analysis put in evidence the similarities, differences, and problems to be solved to cope with the above mentioned scenario.

Chapter 3 presents the context of ECLAP and Europeana, putting attention to the aggregation forms in ECLAP, describing collections, playlists, and media relationships, giving some example related to the ECLAP model and how it is mapped in Europeana Data Model.

Chapter 4 reports the contribution of my thesis presenting the differences respect to the state of the art and putting in evidence the innovative solutions projected and developed in the work done in these years.

Chapter 5, the MyStoryPlayer model of relationships between media is presented by showing the formalization of relationships with respect to the mentioned requirements and scenarios. The semantic model of MyStoryPlayer in terms of triples and an example of the model managed by the client tool during the play of multiple media are presented, showing also how the media relations are produced automatically and some details about manual editing of relationships on ECLAP portal. The MyStoryPlayer is shown during its usage by providing a comprehensive example of navigation among the relationships and for the recording user experience.

Chapter 6, the MyStoryPlayer architecture is described together with sequence diagrams and showing how to access the ontology, through the SPARQL language, giving some examples. Moreover some details about how the synchronization problems have been solved have been provided.

Chapter 7 goes inside the Controller explaining the class functionalities and presents results regarding the assessment of results obtained.

Chapter 8 presents some usage data about the adoption of MyStoryPlayer by the ECLAP users.

Conclusions are drawn in **Chapter 9.**

2. State of the art

This chapter reviews the state-of-the-art techniques on audiovisual modeling and fruition for entertainment and edutainment. In particular, it focuses on three aspects:

- content modeling for synchronized and interactive audiovisual rendering via web,
- media annotation modeling and tools,
- authoring audiovisual annotations for streaming and web applications.

2.1 Content Modeling for Synchronized Audiovisual Rendering via Web

There is a large range of content formats and languages that can be used to model cross media content with synchronizations and relationships, such as standard: MPEG-21 [Burnet et al., 2005], MXF [MXF] AXMEDIS/MPEG-21 [Bellini et al., 2006], SCORM/IMS [SCORM], MPEG-4 [Pereira and Ebrahimi, 2002], and proprietary formats as Adobe Flash, MS Silverlight, etc. In most cases, the presentation layer of these formats are formalized by using specific description languages, such as SMIL [Bulterman, Rutledge, 2009], HTML/HTML5, MPEG-4 BIFS [Pereira and Ebrahimi, 2002], or the less diffuse models and languages as ZYX [Boll and Klas, 2001], NCL [NCL]. Among these formats, we should focus on those that can be managed to set up a *web based collaborative environment* in which several users can access to synchronized and annotated audiovisual to manipulate and interactively play the models. Among them, SMIL has been designed to produce interactive presentations, and it may have links to other SMIL presentations and graphical elements to allow user interaction. SMIL provides support to model transitions, animations, synchronization, etc. In SMIL 3.0, variables can be added to the multimedia presentation thus enabling adaptation to user interactions. Despite to its expressivity, SMIL is still a page description language that needs to reload the page to change context and it is presently not well supported by web based players and browsers. This means that to exploit all SMIL capabilities and constructs one has to install a plug-in into the browser, and modify the plug-in code to cope with the progressive access at the information related to the audiovisual relationships and interactivity details of the client user page. SMIL is mainly typically supported by non-web players such as Ambulant tool. On the other hand, in [Gaggi and Danese, 2011], a SMIL interpreter to rendering a reduced set of SMIL constructs on HTML pages has been created by using JavaScript. This approach is still unsuitable for rendering and controlling synchronized videos. SMIL and HTML has been compared and used in AXMEDIS/MPEG-21 for the presentation layer modeling of AXMEDIS content, exploiting the SMIL Ambulant player library [Bellini, Bruno, Nesi, 2011]. Alternative solutions could be grounded on the adoption of HTML5 with the usage of JavaScript, or the usage Adobe Flash or MS Silverlight. All these solutions may exploit internal features to model and describe the graphic layout of the web page, and interactive aspects, while some script is needed to cope with the access at the progressive information related to the change of context, jump to a different media context, add an annotation, etc.

A relevant problem refers to the difficulties of creating a web based playing of multiple synchronized videos. In this sense, HTML5 as well as Flash or Silverlight do not provide direct solutions. In more detail, a time skew and delay in the synchronizations can be detected among videos when multiple video streams are played / started on the same web page. Typically, the time skew is due to latency and variance in seeking the http based video access and stream, i.e., de-synchronization [Layaida et al., 2002]. These problems may be also depending on the network connection reliability. Moreover, when the user passes from the execution of a multiple video streams on a web page to another set of videos that should start at a selected time instants (i.e., the user changes context), specific techniques to seek the streams from a different starting point of the video are needed.

A different approach could be based on the production of an integrated stream combining the multiple single streams and maintaining the synchronizations, for example by using a unified muxed transport streams. On the other hand, also this latter approach does not provide the needed flexibility and the interactivity of web based solutions that may allow swapping from many connected relationships among a network of annotations. Moreover, seeking, preload and discharging techniques can be applied on specific file formats and keeping under control the client player of the context. Optimization and correction mechanisms for video streaming have been proposed in [Tsai et al., 2010], [Chilamkurti et al., 2010], [Lee and Park, 2010]. The latter adopted a scalable and adaptive video streaming model. In [Meixner and Hoffmann, 2012], a solution for intelligence download and cache management of interactive non-linear video has been presented. The solution proposed aimed at minimizing the interruption when the nonlinear video changes context due to user interaction. The adopted approach was founded on exploiting the knowledge of the structure of the nonlinear video and the adoption of a pre-fetch approach.

In [Gao et al., 2011], an analysis has been performed to solve the mentioned problems by using a combination of pre-fetching (preloading and cache) and frame discarding mechanisms, thus solving the problem by increasing accuracy and reducing delay. The experiments have been performed by using a graphic page built in SMIL. Therefore, the optimization has constrained to modifying the Ambulant player for SMIL 3.0, and not a common web browser. Additional problems have been detected related to the: (i) density of the seeking points in different video formats, (ii) highly compressed video file formats for which the seeking points are not regularly placed, (iii) saturation of the available connection bandwidth with respect to the frame rate and compression of the videos. The solution provided in [Gao et al., 2011] adopted raw videos to keep the delay limited. This solution is unsuitable for web page tools and present high limitation to the impossibility of using compressed videos.

As a concluding remark, the adoption of HTML5 or flash as well as the usage of SMIL in customized Ambulant-based player as in AXMEDIS and in [Gao et al., 2011]) cannot be regarded as the solution of the above identified problems. These formats are powerful coding tools for modeling the graphic page and the user interaction (including the accommodation of multiple videos on the same page). While, they are unsuitable to solve major problems, such as: how to model the audiovisual relationships/synchronizations and pass them to the client side without reconstructing the whole web page; how to progressively access at the needed information only at each change of context; how to cope with saving and reloading user experiences; and how to make the change of context (jump, swap and back) smooth and clean (keeping synchronizations among videos): with the aim of providing a higher quality of experience for the final user.

2.2 Media Annotation Modeling

The literature presents a large number of multimedia (audio visual) annotations models and tools that include both the modalities to annotate audiovisual and play them (i.e., play the media and see, contribute with other annotations). Most of them are grounded on semantic descriptors modeled as media or multimedia annotations formalized in MPEG-7 [MPEG-7] and/or recently in RDF as in Open Annotation of W3C Community Draft [OpenAnnotation]. Moreover, a number of other solutions should be mentioned. Vannotea solution has been proposed for collaborative annotation of videos [Kosovic et al., 2004], [Schroeter et al, 2003]. The main ideas of Vannotea are to allow the collaborative discussion on video content in real-time, thus producing annotations as comments formalized as MPEG-7 and Dublin Core [DC]. The DC is also used to index, search and retrieve video and single segments. The annotations are not typically used to establishing relationships among videos (audiovisual) but only to add on audiovisual segments some text, descriptors, etc. In that case, the annotation database has been developed by exploiting the Annotea model and solution. Annotea is a solution proposed by the Semantic Web Advanced Development group of W3C [Kahan et al., 2001], [Koivunen et al., 2003]. The idea of Annotea is to create annotations which can refer to the annotated media. The annotations are modeled as RDF. Xpointer is adopted to link them to annotated media content. The RDF-based annotations can be searched by using semantic queries, for example in SPARQL. The single annotations are associated with html/xml elements on the web. For their execution and connection to the media source (i.e., play and rendering) an *extended version* of Mozilla Firefox has been provided, thus constraining the users to adopt a specific browser. In [Bellini, Bruno, Nesi, 2011], AXMEDIS/MPEG-21 [Bellini, Nesi, Rogai, 2007] has been exploited to create annotations in cross media content. These cross media annotations are overlapped to images, audio and videos, and may refer to single elements/essences included into cross media content by using AXMEDIS internal links and protocol. AXMEDIS annotations may be produced, collected, and shared. While MPEG-21 has been initially formalized as a set of descriptors for content modeling and distribution [MPEG-21 DIS], managing also synchronization via MPEG-4 standard and related transport stream.

There is an ongoing work in the W3C Open Annotation Community Group for the development of a common data model and ontology for the representation of annotations [OpenAnnotation], a working draft was produced in Feb. 2013 http://www.openannotation.org/spec/core/. This group was jointly founded by the Annotation Ontology and the Open Annotation Collaboration. The OpenAnnotation model is based on one or more ‘Target’ elements referring to the digital resources (or its part) being annotated and some ‘Body’ elements that represent the body of the annotation (i.e., a textual comment), the annotation body can be a text but also any other digital resource (or its part). The OA model is quite general and does not prescribe how the annotations should be presented to the user, in particular it is not possible to represent ‘explosive’ annotations stating that a video at a certain time instant should be replaced by another one that for example provides details about a topic.

In the above mentioned cases, the annotation models are focused on augmenting scenes with additional descriptors and marginally on establishing relationships among audiovisual as in MyStoryPlayer. In most cases, the annotation models do not describe the underling semantic model to be executed during the play of the annotations on the browser/client player. For example, what happen when an audio-video segment is annotated by another video segment and the former has a higher time duration; how can annotate a video in a given time instant to play 10 minutes of another video and by stopping the annotated former video during the play of the latter; what happen when the context is changed (the user prefer to follow the story of the annotation and not the story into the annotated video). The answer to these questions can be provided only combining the annotation formal model with an executable semantic.

Moreover, these activities should be accessible online, via a sort of streaming annotations, as in the plan of the OpenAnnotation [OpenAnnotation].

2.3 Authoring Audiovisual Annotations for streaming and web applications

In the literature, several tools for media authoring have been proposed, see for a detailed review [Bulterman and Hardman, 2005]. In general, the authoring tools for media annotation range from professional tools for broadcasting annotations with videos for Video on Demand (VOD) distribution, web TV such as 4oD (Channel 4's download service), BBC i-player, Sky+. Users of these annotated videos are mainly passive and are interested to marginally interact and change the stories they are observing. An example is eSports, which is a collaborative video execution and annotation environment [Zhai et al., 2005]. The annotations can be integrated into the video stream to be distributed to final users via RTSP streaming. The annotations can be simple text, audio and geometrical forms and are coded by using MPEG-7. In this case, the audio visual can be used as annotations on other video and cannot be used for navigation on a multi-stream view, for example to jump on a different context by selecting one of the videos presented. A video annotation tool for MPEG-7 has been proposed also in [Neuschmied, 2007] mainly for describing the content inside the scene. Other video annotation models and tools, such as the IBM VideoAnnEx [Smith and Lugeon, 2000] allow to comment video with static scene descriptors. SMAT [Steves et al., 2000] allows to collaboratively annotate video clips with text on whiteboard. Hyper-Hitchcock [Shipman et al, 2008] is an interactive environment for authoring and viewing details on demand video. It includes an interactive editor to add details on videos through a process of composition and link creation, a hyper video player, and algorithms for automatically generating hyper video summaries of one or more videos allowing viewers to navigate among video chunks. What's Next [Shen et al, 2009] is a video editing system helping authors to compose a sequence of scenes that tells a story, by selecting them from a corpus of annotated clips. Authors can type a story in free English, and the system finds possibilities for clips that best match high-level elements of the story. It operates in two phases, annotation and composition, working with recommendation functionalities. NM2 (New Media for a New Millennium) [Hausenblas, 2008] aims at developing tools for the media industry enabling the efficient production of interactive media. NM2 productions consist of a pool of media assets to be recombined at runtime based on a logical description of the story and the end user's interaction.

On the other hand, in web applications the annotations and video relationships can be dynamically user generated. A simpler annotation web based model and player can be observed in solutions such as the users' annotations on the images of Flickr or YouTube Video Annotation [YouTube] which provides links to navigate among different related videos. Similar solutions can be obtained in HTML/HTML5 with java script (see for example, popcorn.js). The jump to the next video interrupts the experience of the users. In the latter case, an annotation may allow to select the next scene, to pass to different videos, to model the paradigm of Hypervideo (see MIT Hypersoap [Dakss et al., 1998], OverlayTV [OverlayTV], etc.). A range of authoring tools allows creating annotations for the analysis of the narrative of media such as [Lombardo and Damiano, 2012], [Ursu et al., 2008]. Simple paths may be created by starting a video from an overlapped link placed on another video. In some cases, the semantic description may also establish time synchronization of the annotation with the annotated audiovisual media. This occurs in *Lignes de Temps* [LigneDeTemps], eSports [Zhai et al., 2005], which allows to synchronize simple annotations to a video. This means that during the video play/execution annotations may pop up, be browsed and discussed. In this line, Virtual Entrepreneurship Lab, in short VEL of [Klamma et al., 2006], has proposed video annotations in MPEG-7 about educational activities, which, substantially, are links to other videos in a graphic page, and the user may select them. In VEL, a video may offer related videos which the user may jump on (restarting it from zero on the central player), and they may be considered as simple annotations. In those cases, the time lines of the video annotations are neither ordered nor related one another, and any video annotation is a stand-alone video. In [Meixner and Kosh, 2012], a XML format for producing non-linear interactive video (presentations in which video segments can be activated by users by using links, buttons, and by conditions) with the aim of providing a tool for navigating in structure has been presented. Also in this case, the model has been proposed for creating pre-produced elaborations that cannot be changed by the users enriching the non-linear video model.

Most of the above mentioned annotation tools are mainly oriented to provide a tool for creating annotations for preparing the video streaming (see for example the above cited: NM2, AXMEDIS, eSport, What's Next) and do not provide support for creating annotations from the client side, accessible and selectable from the web browser by the final users. The mesh of possible annotations and relationships among audiovisual can be prepared and offered to the users that can navigate in the related media exploiting aside menu or overlapped choices. Despite to this large work on annotation tool, most of them are focused on annotation formats that do not support the definition of the player behavior in executing the annotation (i.e., annotation semantics), and neither the progressive delivering of annotations on demand from a client and a server providing them. Moreover, they do not allow the annotation of an audiovisual with another audiovisual segment with the aim of creating a browseable set of relationships among audiovisual relationships and annotations, and presenting the temporally overlapped audiovisual as synchronous media. In most cases, they are limited in the number of streams and media that can be played at the same time and also the usability in terms of context change is limited.

For many applications, the preventive production of annotated media (non-linearly related video segments) by compounding media and providing all possible paths is not viable, since this approach could limit the interactivity and manipulation capabilities of the users (such as the creation and addition of other relationships). Moreover, the activity of media compounding would be regarded as the exploitation of media adaptation rights – e.g., [MPEG-21]. This right also implies the possibility of modifying the original media that is a formal right quite difficult to be obtained by the content owners. Therefore, a web annotation tool has to be capable to work on audiovisual and preserve their integrity [Bellini et al., 2011]. This means to create annotations and the composition and navigation of media relationships and synchronizations without changing neither combining in a stream the original media. Therefore, the solution to create new experiences based on the composition, aggregation and reuse of accessible audio visual can be restricted by establishing relationships that can be executed at the play time without changing the original media.

Another very important aspect is the mechanism to help identifying the media to be annotated on the basis of the associated metadata. A large amount of audio visual may directly arrive with some pre-established relationships among one another: sequences, multicamera shooting, collections, playlists, etc. For example, it is common to have audiovisual belonging to the same sequence may have title with a common or similar root. For example, “Enrico iV, act.1”, “Enrico iv, 2/2”.

3. Best Practice Networks: ECLAP inside Europeana

The richness and value of the European performing arts heritage is unquestionable. Most of valuable collections are now being digitized and published online, while they remained scattered, and coordination is lacking between digital libraries and the performing arts field; however, there is a high demand for access to this content. ECLAP (European Collected Library of Artistic Performance) partially filled this gap by creating a considerable online archive (with more than 1 million of items: document pages, images, audio, video segments) for all the performing arts in Europe, and providing solutions and tools to help performing arts institutions to enter the digital Europe by building a network of important European performing arts institutions and archives and publishing content collections on Europeana. Europeana (*http://www.europeana.eu*) collects cultural heritage metadata coming from several institutions, universities, foundations, museums, schools of art that represents a cultural heritage of the huge European history.

ECLAP has created a best practice network, making use of advanced semantic tools for the production and dissemination of the rich multilingual European heritage. This resulted in cultural enrichment and promotion of European culture, and in improvements in learning and research in the field of performing arts. ECLAP has been co-funded by the European Union ICT Policy Support Program.

The main objectives of ECLAP project are:

- **Bring together Europe's most relevant performing arts institutions** to provide their content on Europeana. The consortium has been enlarged to a higher number of additional affiliated partners that have contributed to ECLAP in terms of content and networking.
- **Create a best practice network** in order to connect European performing art institutions; ECLAP is now followed by a large number of institutions: former

partners, affiliated partners, and networking partners. http://www.eclap.eu/3578 . Most of them have contributed to the creation of best practice documents in major areas in which ICT can contribute to the performing arts digital valorization and management.

- **Make accessible via Europeana a large amount of performing art** material: theatre performances, theatre lessons, master classes, theatre teaching material, etc. in the forms of videos, audio, documents, images, and cross media, etc.; ECLAP provided more than 1 million of content items coming from performing arts institutions to Europeana.
- **Provide solutions and services to performing arts institutions** to bridge the gap between them and Europeana via published guidelines and ready to be used technical solutions; ECLAP has created a large number of tools to cope with performing arts content and metadata; and documents to explain how the performing art metadata can be used, processed, formalized and mapped into EDM Europeana and LOD. Early contributions to standards have been produced. Moreover, a number of tool for content annotation, enrichment, processing and semantic navigation have been produced as well, among them MyStoryPlayer, Social Graph, Metadata Editor, AXCP processing, recommendations, etc.
- **Provide solutions and services to final users** (teachers, students, actors, researchers, and performing arts lovers for edutainment, infotainment and entertainment) to access content never before accessible via the Internet and to satisfy their needs. AS described above, ECLAP has created a number of tools which are suitable for education and research on performing arts content, such as: MyStoryPlayer, Social Graph.
- **Create a virtuous self-sustainable mechanism** to provide continuously access to the content and to increase the number of online materials, providing revenues for those institutions that need them, in order to maintain and improve the service, increasing both functionalities and the quantity of materials available. Sustainability has been designed and planned while the real sustainability will be concretized after the project conclusion. DISIT is going to sustain the portal alive for a number of years after the completion of the project.

- **Contribute to the clustering activities** organized by the EC, comparing solutions and integrating them with the Europeana general view. This activity has been performed, as a major result ECLAP contributed to the definition of Europeana pro (former Europeana Share), and also contribution on working groups related to LOD, semantics, aggregations, metadata, etc., of Europeana.

3.1 ECLAP infrastructure and tools

Recently, professional users are unsatisfied by those general purpose social media solutions since they do not provide satisfactory facilities to perform advanced semantic aggregations and associations, learning management, that are, in effect, needed for educational and professional purposes. This new trend has determined the creation of a number of more specific and tuned services that in the case of digital library for arts can be identified as: Artyčok: http://www.artycok.tv , Digital Theatre: http://www.digitaltheatre.com , Digital Dance Archives: http://www.dance-archives.ac.uk , SP-ARK: http://www.sp-ark.org , etc. The needs of more sophisticated content services is becoming more concrete, pretending from the content and user services new social and semantic features with collaborative tools, aggregations tools, linked data, connection with social networks and mobile devices, augmented reality, navigation tools, etc.

ECLAP offers a wide support to cultural institutions on performing arts in moving towards the direct exploitation of new technologies for digital content management for different purposes that range from direct: dissemination, promotion towards Europeana, open data, social media, education and training, to the better understanding of new technologies and solutions. To this end, ECLAP is both a Best Practice Network and provider of Content and User Services.

ECLAP has set up a Best Practice Network and service portal, making use of advanced social network semantic solutions and delivery tools for the aggregation and distribution of rich multilingual performing art content. This will result in cultural enrichment and promotion of European culture, and in improvements in learning and research in the field of performing arts. The activity performed in ECLAP is focused on setting up an infrastructure (http://www.eclap.eu) for integrating multi-lingual metadata and content coming from several European Institutions in Europe, and it is open to other partners via the Affiliation and Networking agreement. On ECLAP, users can provide and aggregate content files and metadata. Several metadata formats and modalities are supported. A large set of different metadata formats can be ingested. To this end, a metadata mapping tool has been developed, and the mapping goes towards the ECLAP ingestion semantic model. Once the metadata area ingested, an intelligent content processing back office (based on AXCP media grid) is capable of collecting and automatically repurposing content for distribution via pc and mobiles, coping with more than 600 digital file formats.

The ECLAP **Content and User Service** exploits the use of advanced social media and semantic computing technologies and solutions for the content enrichment, aggregation and distribution of rich multilingual performing art content towards multichannel: PC and mobiles. Presently, ECLAP distributes more than 120000 distinct objects, up to 13 major metadata languages, towards a community of about 2200 registered users, world-wide distributed, while the largest communities of users and connections come from Italy, UK, USA, France, The Netherlands, Portugal and Slovenia.

This means that ECLAP users can search from a unique service both technical documents and performing arts content. This approach may be an advantage since in many cases the distinction from technical documents and performing arts content is not strict. For example, these aspects are mixed and blurred in backstage technical documents, interviews, educational content, comments, and web pages. This also means that, for this purpose, the indexed content on ECLAP is heterogeneous and cross media, ranging from video, audio, documents, images, to blogs, web pages, collections, play lists, annotations and tools. In fact, at each query the ECLAP query service provides an answer in terms of faceted results including different content types and formats. The indexing and search facilities of ECLAP provide support for fuzzy correction of typos and for advanced queries, with and/or operators, substrings, perfect matching, etc. ECLAP offers a wide range of innovative solutions and tools to support cultural institutions in managing, providing access to and disseminating their online digital content collections to users in 21 languages. The ECLAP major content services are reviewed in the following.

Figure 3.1: ECLAP Front Page

The ECLAP portal is capable of handling a large range of different media kind: video, audio, images, document, slides, synchronized slides and video, 3D, cross media (e.g., smil, mpeg-21, html, eBooks as e-pub), animations, pdf, blog, groups, comments, news, playlists, collections, annotations, forum, archive, tools, excel, etc. In the ECLAP, portal it is also possible to create content collections, playlists, aggregation for e-learning courses and audiovisual annotations via the MyStoryPlayer tool (see Chapter 5) and relationships navigation via SocialGraph (http://www.eclap.eu/116088). Both tools have been selected by Europeana pro and are not promoted on Europeana pro web pages.

ECLAP offers a wide range of innovative solutions and tools to support performing arts institutions in managing, providing access to and disseminating their online collections to a large number of users. This initiative is bringing together hundreds of thousands of Europe's most relevant performing arts content (often previously inaccessible on Internet), including collections on theatre, dance, music, cinema and film. These consist of performances, lessons, master classes, educational material, festival information, costumes, sketches, lyrics and posters. File formats include video and audio files, documents, images, animations, playlists, annotations, 3D, interactive content, e-books and cross media content.

Through ECLAP, users are able to: search, retrieve and play extensive high quality multilingual content via a semantic model with fuzzy support; enrich, validate and contextualize metadata for a large set of content types; aggregate content in play lists, collections and e-learning courses; upload and share multilingual content; receive suggestions and recommendations for similar content, on ECLAP and on Europeana; comment, annotate, tag, rate and vote on content; register and network with others colleagues; create discussion groups and distribution channels; upload digital resources for professional and user generated content; use tools to solve IPR issues with an IPR Wizard, to regulate content access; access all content via different devices such as PCs, tablets and Smart phones.

The aim of the ECLAP workflow is to ingest content and metadata in a range of formats and to bring them to reach a high quality level of metadata completion to provide them to Europeana. To this end, collaborative ECLAP tools (for ingestion, metadata enrichment, editing and validation, IPR definition) have been studied and developed. Once the metadata are enriched, ECLAP is providing semantically enriched metadata by mapping them from ECLAP semantic model to Europeana in EDM semantic model via an OAI-PMH ECLAP server.

Institutions and stakeholders may exploit the aforementioned features for their users plus a set of features to allow them to manage their content and the user community. To become an active institution in ECLAP they should become an Affiliated Partner. The main features for partners are related to:

- **Users:**

  - create a personalized group and collaborative environment for users and content, to present your content according to your preferred modalities;
  - engage users, create discussion groups and distribution channels to promote and valorize content and activities;
  - configuring user roles and accesses at different levels;
  - manage and stimulate user engagement during live events;
- **Content:**
  - publishing and promoting content to Europeana via ECLAP;
  - upload and provide any archival formats metadata;
  - upload professional digital resources, any kind of formats to be promoted / published on ECLAP, exploit tools for massive and automated content ingestion;
  - managing content for expositions and museums via QR code and GPS tools.;
  - exploiting advanced indexing and search capabilities of ECLAP;
- **General:**
  - collaborate to the definition of the next steps of ECLAP;
  - assessing and monitoring group activity on content and users;
  - use tools to control IPR issues, to regulate content access;
  - controlling the embed of content into third party portals.
  - create and manage e-learning courses;

3.2 Aggregation in Europeana

The digital library Europeana collects all the metadata from each projects related to it. For Aggregation Europeana intends the set of related resources in the Europeana system about one particular provided object from one provider and grouped together as an Aggregation. These are either created by the provider or generated from the metadata by the Europeana system.

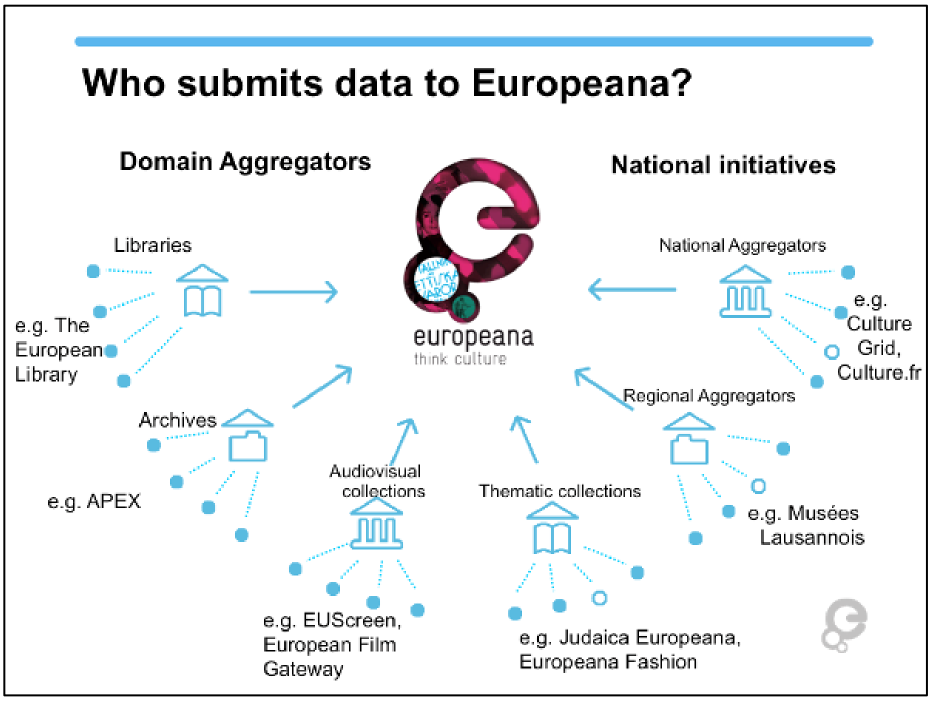


Figure 3.2: Aggregation in Europeana

An aggregator in the context of Europeana is an organization that collects metadata from a group of data providers and transmits it to Europeana. Aggregators also support the data providers with administration, operations and training.

Europeana aims to create strong partnerships and support the development of aggregators on a national level in Europe and of pan-European aggregators representing a specific segment or sector.

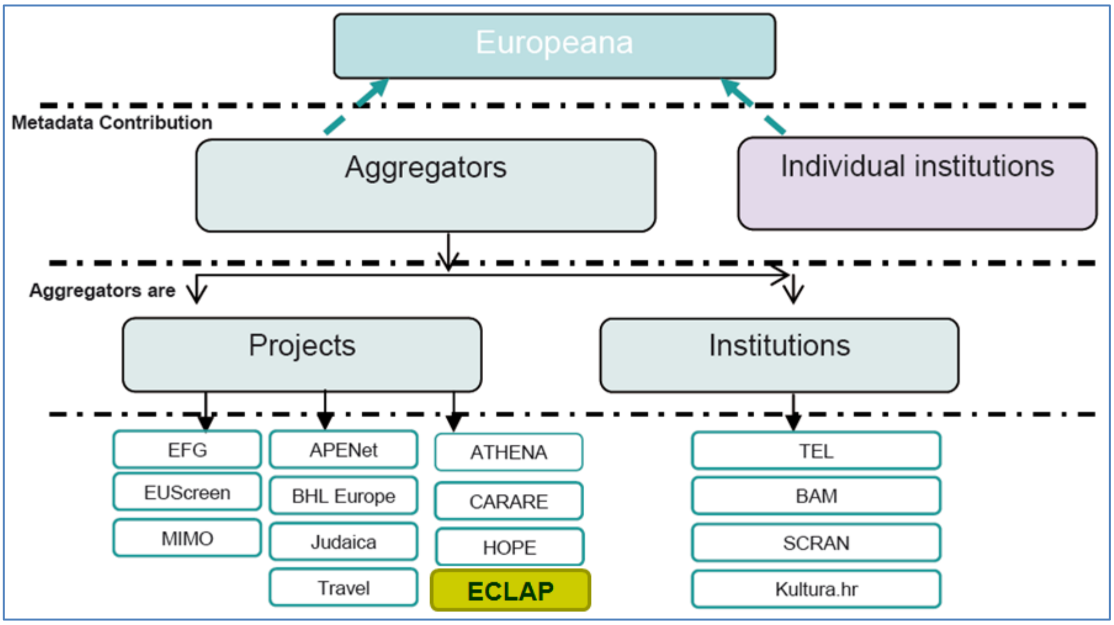


Figure 3.3: Aggregators in Europeana

3.2.1 EDM

The Europeana Data Model (EDM) is a proposal for structuring the data that Europeana will be ingesting, managing and publishing. The Europeana Data Model is a major improvement on the Europeana Semantic Elements (ESE), the basic data model that Europeana began life with.

EDM is not built on any particular community standard but rather adopts an open, cross-domain Semantic Web-based framework that can accommodate the range and richness of particular community standards such as LIDO [LIDO] for museums, EAD [EAD] for archives or METS [METS] for digital libraries.

EDM not only supports the full richness of the content providers' metadata but also enables data enrichment from a range of third party sources. For example, a digital object from Provider A may be contextually enriched by metadata from Provider B. It may also be enriched by the addition of data from authority files held by Provider C, and a web-based thesaurus offered by Publisher D.

EDM supports this richness of linkage, while clearly showing the provenance of all the data that links to the digital object.

The design of EDM has been influenced principally by requirements:

- distinction between “provided objects” (painting, book, movie, archaeology site, archival file, etc.) and their digital representations.
- distinction between objects and metadata records describing an object.
- multiple records for the same object should be allowed, containing potentially contradictory statements about this object.
- support for objects that are composed of other objects.
- compatibility with different abstraction levels of description.
- EDM provides a standard metadata format that can be specialized.
- support for contextual resources, including concepts from controlled vocabularies.

Also, a basic motivation for EDM is to support the integration of the various models used in Cultural Heritage data, so that all original descriptions could be collected and connected through higher-level concepts. This motivation, derived for the general goal of Europeana to exploit the richness of all available data in order to support the richest possible functionality, justifies three fundamental design principles:

- EDM allows data integration in an open environment: it is impossible to anticipate all data contributed.
- EDM allows for rich functionality, possibly via extensions.
- EDM should re-use existing (standard) models as much as possible.

These design principles are the basis for the choice of Semantic Web representation languages—RDF(S), OWL—for EDM. These allow flexible re-use and articulation of existing models, as demonstrated by the conception of the EDM model itself, and by the mapping approach to data integration which underlies the way EDM should be used in practice. Further, the Linked Data approach emphasizes the re-use and linkage of richly described resources over the web. This really fits the EDM ambition of making use of existing resources as well as supporting their enrichment, notably via the establishment of new relations between them. Whether these resources belong to one Europeana provider’s information space, to different providers’ spaces, or to external spaces used as knowledge references.

3.3 Form of Aggregation in ECLAP

This section is focused on presenting an overview of the ECLAP semantic model, with a specific accent on the aggregations: playlists, collections, annotations, synchronizations. The ECLAP semantic model is reported in Figure 3.4. Moreover, several other kinds of concepts and data are modeled ranging from content to users and their relationships. The different kinds of contents are classified with a thematic SKOS [SKOS]: genre, performing art type, historical period, subjects, etc. Also the user profile includes such a classification to allow users to express their preferences about content theme, and to exploit that information for suggestions and/or recommendations. Content in ECLAP can be: image, document, video, audio, e-pub, MPEG-21, HTML, archive, animations, etc., but also the aggregation as playlists, collections, web pages, annotations, comments, etc., and considered content as well.

In Figure 3.4, Object is specialized in a Collection that is associated with a set of collectable objects as document, playlist, archives, tool and audiovisual objects. An Object is associated with a set of Metadata elements that can be DublinCoreMetadata, TechnicalMetadata or PerformingArtsMetadata, and taxonomical classification. A Playlist is associated with a set of AVObjects. The AVObjects are specialized in Image, Video and Audio. A set of Annotations can be associated with an AVObject whereas an Annotation can be associated with one or two AVObjects. In this way, an Annotation may annotate an AVObject with another AVObject. Annotation and playlist are associated with time, in order to put in relation audiovisual objects each other. Regarding the connection with users, the RegisteredUsers provides a set of Content and can perform some Action that may modify a Content in some way. The Actions can be specialized in ToRate, ToComment, ToTag, ToSocialShare, ToRecommend, ToValidate, ToPublish, ToView. Moreover, a user can participate or prepare a course from ECLAP, depending on his role.

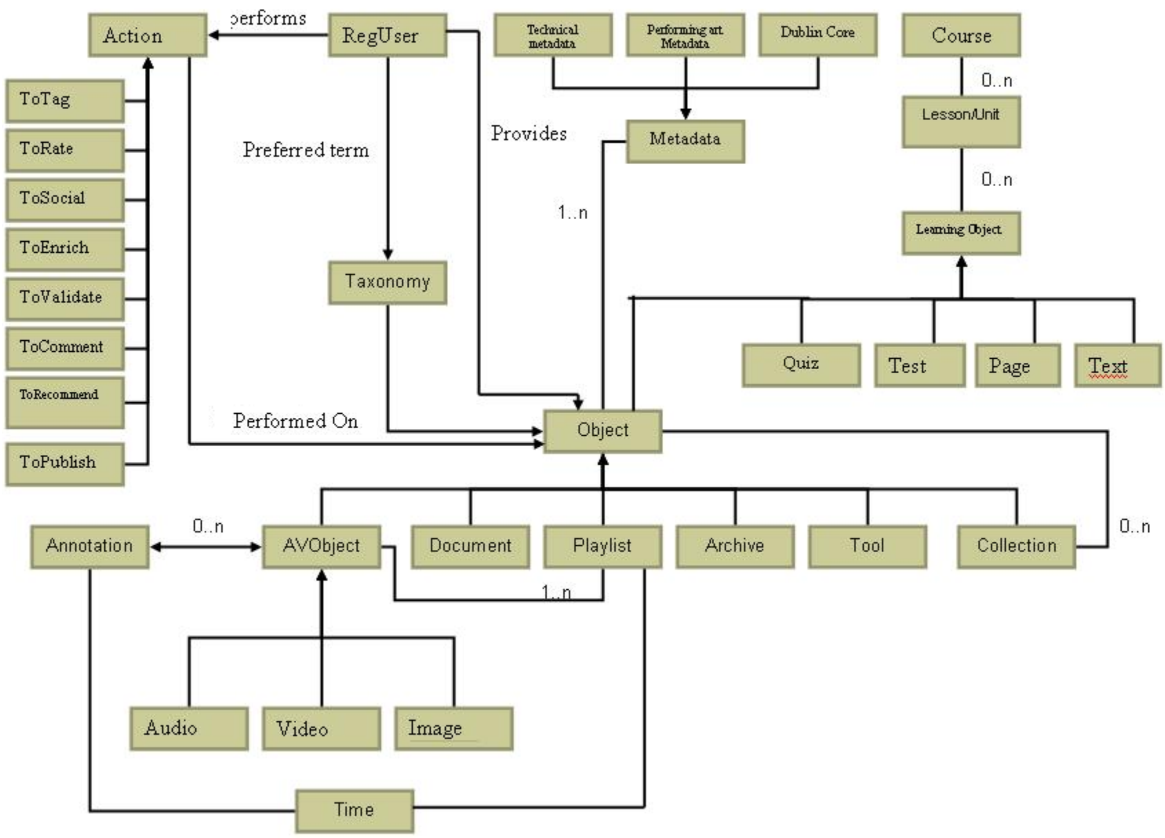


Figure 3.4: ECLAP Semantic model (a part)

As shown in the general model, Playlists and Collections are a way to aggregate ECLAP content Objects. Playlists can aggregate only Image, Audio and Video content while Collections aggregate any kind of Objects. Playlists and Collections are Objects and for this have their own metadata set and in particular have a workflow type, this allows to provide to Europeana also aggregated content. Playlist allows users to organize and reproduce in a specific order a series of media contents, or some part of them. Collections are a set of media or document that can be used for a course, or can be played according to a specific semantic. The e-learning courses can be created as a sequence of lesson units, each of which is produced by exploiting a set of selected objects, collection, playlists and related annotations.

ECLAP audiovisual annotations are performed and managed by MyStoryPlayer [Bellini et al., 2011] model and tools, that can be useful to record some experiences in an annotative structure where the teacher explores only a part of the entire possible worlds following a leit-motif specific for his lesson, in order to share his experience with the students. With the MyStoryPlayer annotation system it is possible to synchronize contents and view them in the same player. This could be useful when there are different synchronized views of the same scene or e-learning context (e.g., slides and teacher videos, teacher face and hands and camera in a surgery).

3.3.1 Collections

According to the ECLAP model of Figure 3.4, a Collection is a set of contents: audio, video, images, documents, pdf, playlists, zip files, etc. The collections can be created on the ECLAP Portal and may be published or kept private by the users: *Unpublished,* the collection will be visible only to creator in a draft form; *Published,* the collection has been uploaded as new a content item (also indexed, etc.) and can be shared with other users and published toward Europeana, for example.

In order to keep separate temporary collected objects and collections, the system creates different xml descriptions on server for each user. Once a content is on the temporary list, it may be added to a personal collections, to be published later. The descriptive xml structure is the reported as in the following example (where the AXOID is an UUID):

```
<collection xmlns="http://…”>
   <Record>
      <DrupalNode>3452</DrupalNode>
      <Media>
         <Title>mask</Title>
         <MediaType>Image</MediaType>
         <Axoid>urn:axm:…7fb6ce0</Axoid>
         <Ext>.gif</Ext>

      </Media>
   </Record>
   <Record>
      <DrupalNode>3451</DrupalNode>
      <Media>
         <Title>Theatre curtain</Title>
         <MediaType>Video</MediaType>
         <Axoid>urn:axm:00000:obj:e2e24d4b-22ab-
```

```
                48ec-a461-98250f283b55</Axoid>
            <Ext>.flv</Ext>

        </Media>
    </Record>
    <Record>
        <DrupalNode>3571</DrupalNode>
        <Media>
            <Title>i-theatre annotate</Title>
            <MediaType>Document</MediaType>
            <Axoid>urn:axm:...0da8b5a</Axoid>
            <Ext>.pdf</Ext>

        </Media>
    </Record>
</collection>
```


The descriptors of the elements collected into the Collection can be recovered by using the Drupal Node id and/or the UUID coded as AXOID into the system.

The collection may include classification and technical information to be shared according to LOD model. When a user publishes a collection, classification and technical metadata have to be assed. In ECLAP, also the association to taxonomy and groups is requested. The definition of metadata information keeps unchanged the multimedia objects composing the collection, but completes and produces an additional semantic descriptor to the collection created. This information is indexed, shared and accessible, so that it can be used to make queries for searching collection objects on the ECLAP Portal.

In ECLAP, a set of operative functionalities are available to manage collection such as: list of resources, add/delete resource, creation and editing, saving and publication of a collection, assigning metadata, updating of a collection already published. The collection can be uses as the primary source for producing courses. The user may create its own thematic collections for creating a repository of content for the composition of its own lessons/courses. In many cases, Collections may be published and exported on Europeana as described in the following. For example when the collection represent an added value: the pictures in a room, the set of picture of the same opera, the set of content used to prepare the opera, the different camera views of the same theatrical representation, etc.

In Figure 3.5, four resources have been selected and are present in a temporary list (e.g., ‘Theater curtain’, ‘i-theatre annotate’, etc.), while three of them have been grouped in the ‘Theater’ collection, by selecting them and adding them to the collection. Once a user has created an unpublished collection (i.e., private) he/she can publish on the Portal. Already published collections can be also updated without changing the metadata.

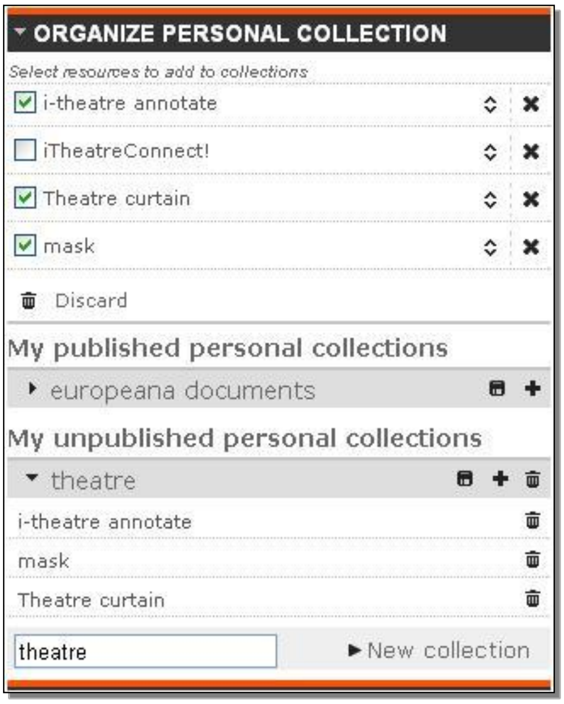


Figure 3.5: block for managing personal collections. In the upper side the temporary collection, in the second part published and unpublished collections are visible.

In general, the concept of Collection does not have an executable semantics, in the sense that the playback of a collection is not bringing to the frontal reproduction of the single content of the collection. As limit case, the collection would have an ordering of content elements, but only in some specific cases. In the next section, a more sophisticated model of collection is presented. It is typically proposed and called as PlayList.

3.3.2 Playlists

In most cases, the concept of playlist is used by the users to create specific collections of ordered audiovisual content, e.g., a set of audio tracks, a set of videos. The playlist execution consists in the sequential play of the single content objects. The executable semantic of play along the time can be easily applied to audiovisual content such as image and audio tracks.

In ECLAP, the play list model also support the inclusion of images and audiovisual segments of the media. According to the model of Figure 3.4, a playlist can be a set of audiovisual contents. The same media can be included in several playlists or even in the same with different or identical temporal segments. The order or execution and their timeline for each playlist and object can be user defined. Playlists are a way of playing an ordered series of multimedia contents.

Therefore, the playlist model of ECLAP is not the classical play list of many social networks. In ECLAP the user may put in the play list even a small segment of video and audio, without the modification of their corresponding files. This permits to create e-learning units without the need to cutting the audiovisual pieces. The semantic of playlist allows to play the audiovisual content according to the identified segments and sequence only once or in a continue loop (images, audio, and video).

From the operative point of view, a user can take from the portal any audiovisual segment to compose its own playlist. The operation is performed on a specific audiovisual flash player in which one can select a start and (|-) an end (-|) of the audiovisual segment, as depicted in Figure 3.6.

Figure 3.6: player loaded to select a segment of a video in a playlist

In the case of images included in the ECLAP playlist, a duration time of their permanence on screen has to be defined by the user. A duration of default is set to 5 seconds. In the play list, the same single media can be included multiple times. This allows to perform jumps In the temporal list of the audiovisual, and/or create narrative experiences with a simple tool.

Once a play list is complete, the user may decide to publish and share it in ECLAP. In this case, specific metadata are requested, plus taxonomical information. The published playlist can be aggregated in Collection and into Courses as basic elements. Please note that the association of metadata keeps unchanged the multimedia objects composing the playlists, and gives a semantic on the playlist created, assigning the order to included media segments to be played.

The XML descriptor of the playlist may be accessible from ECLAP to the playlist player on the client side to keep available all the information for the system during the playback phase. A reduced number of information regarding the playlist may also be passed to Europeana. In this case, the executable semantics of ECLAP playlists is lost mapping the information to EDM.

3.3.3 Media Relationships

The possibility of creating annotations on audiovisual is provided on several portals by using several different kind of tools and solutions, see [Bellini et al., 2011]. In most cases, the annotations are textual or semantic descriptions of the scene, e.g.: Carl is talking with Peter, a table with a two chairs, Sausalito beach. In addition, to this requirement, there is the need of synchronizing audiovisual media. For example, to show the video and slides synchronized each other, to see at the same time different synched views of the same scene (may be taken from different points of views) in a theatre (see this example on ECLAP portal).

The integration of classical annotations and synchronization functionalities is an important feature to be accessible in the educational environments.

Therefore, in order to satisfy these requirements, the model and tool enforced in ECLAP has been derived by the MyStoryPlayer [Bellini et al., 2011]. With MyStoryPlayer a user may annotate an audiovisual content segment (image, video, audio) with another audiovisual content segment. This allows to define synchronizations and to associate to those segment also textual and/or descriptive annotations. These annotations are saved into an RDF model located to the server, and partially sent to the player on demand according to contextualized SPARQL queries.
So that, in ECLAP, a user can perform annotations on any audio, video or image located on the portal. This activity is performed by using a simple on-line tool, as described in Chapter 5. Then the annotations performed may be played/executed by the MyStoryPlayer tool.
Moreover, according to the MyStoryPlayer model (see Chapter 5), there is no difference between a media and a annotation. The annotation can be also a media. A media can be at the same time a resource for an annotation and an annotation. This leads to build an relationship structure in which a user can navigate among the several annotations. An annotation, according to the semantic model, can involve one or many media. It is based on a starting and ending time, and a different semantic of reproduction can be associated to it.

As for the other features like play lists and collections, no alteration is done on the original file when media are annotated and/or synchronized each other. The model is based on a set of RDF triples to code the relation, saving it in the semantic database, in order to allow searching for relations on users profile.

3.3.4 Aggregation To Europeana

Recently, the new Europeana Data Model (EDM) for metadata ingestion and management has been proposed. The new model is based on well defined semantic web standards as ORE, Dublin Core [DC] and SKOS [SKOS]. Noticeable requirements for the EDM model were:

- distinction between “provided object” (painting, book, movie, archaeology site, archival file, etc.) and digital representation
- distinction between object and metadata record describing an object
- multiple records for the same object should be allowed, containing potentially contradictory statements about an object
- support for objects that are composed of other objects
- compatibility with different abstraction levels of description
- EDM provides a standard metadata format that can be specialized
- EDM provides a standard vocabulary format that can be specialized

One of the main goals of EDM is to allow the integration of the different data models used in Cultural Heritage data, in order to collect and connect through higher-level concepts all original descriptions coming from several Content Aggregators. This brings to three fundamental design principles:

- allows data integration in an open environment;
- it is impossible to anticipate all data contributed;
- allows for rich functionality, possibly via extensions.

Analyzing the EDM model in the context of Aggregation, two basis classes of resources provided to Europeana are identified: the “provided object” itself and a (set of) digital accessible representation of it. This permits to distinct between “works”, which are expected to be the focus of users’ interest, and their digital representations, which are the elements manipulated in information systems like Europeana. According to the ORE approach through the ore: Aggregation class , the provided object and its digital representation, given by one provider, form an aggregation. Each instance of ore:Aggregation is related to one resource standing for the provided object, through ens:aggregatedCHO property, and one or more resources that are the digital representations of the provided object through the ens:hasView property.

The present version of EDM integrates the former model of Europeana called Europeana Semantic Elements, ESE, by re-contextualizing each element in the more structured context of EDM. This means that each ESE element is viewed in the EDM as an RDF property, composed by a domain and a range, and connected to the rest of the EDM via a set of sub-property links. The rationale for the integration is to enrich EDM with a set of properties that have proven to be most useful in modeling Cultural Heritage Objects. As every object must have at least one ESE record, and as an ESE record is a valid EDM instance, these properties constitute an interoperability core within EDM.

Additional mapping possibilities coming from other data models are offered by ESE properties, augmenting the interoperability potential of EDM.

The integration of ESE into EDM leads ESE to be expressed in RDF, so that bringing additional advantage of exploiting the web architecture for linking resources.

In particular, in the context of EDM deployment, the values of ESE properties that are currently given as simple strings could be given, in a typical RDF form, as pointers to fully-fledged (RDF) resources standing for concepts, agents or places (to name a few) that would be provided with complete description and linkage to other resources. This applies in particular to both Dublin Core properties (e.g., dc:creator) and to ESE-specific ones (e.g., ens:isShownAt).

As EDM supports the delivery of aggregated content, ECLAP can use Collections as a kind of aggregated content that may be provided to Europeana.

Moreover, ECLAP can use the extensibility of EDM to define specific specialization for some properties to provide more detailed information on a content. For example custom properties can be defined in the following way:

eclap:director rdfs:subPropertyOf dc:creator.

eclap:lightDesigner rdfs:subPropertyOf dc:contributor.

eclap:dateOfPerformance rdfs:subPropertyOf dc:date.

Where director property is defined as sub property of Dublin Core creator, light Designer as sub property of contributor and date of performance as sub property of date.

The example of an aggregated content is the following (using the turtle syntax):

eclap:Collection rdfs:subClassOf ore:Aggregation;

rdfs:subClassOf eclap:Content.

<urn:axmedis:0000:obj:abc…> a eclap:Content;

dc:title “you PARA | DISO”@en;

dc:creator “emio greco & pc”;

eclap:director “erik lint”;

eclap:lightDesigner “xyz”;

eclap:dateOfPerformance “2010/07/15”;

…

<urn:axmedis:0000:obj:0123…> a eclap:Content;

dc:title “Divina Commedia, Paradiso”@it;

…

eclap: aggregation_10231 a ore:Aggregation;

dc:creator “ECLAP”;

ens:aggregatedCHO <urn:axmedis:0000:obj:a1b2…>.

<urn:axmedis:0000:obj:a1b2…> a eclap:Collection;

dc:title “Paradise from past to future”@en;

ore:aggregates <urn:axmedis:0000:obj:abc…>;

ore:aggregates <urn:axmedis:0000:obj:0123…>.

Where *<urn:axmedis:0000:obj:a1b2…>* is a collection aggregating the two contents *<urn:axmedis:0000:obj:abc…>* and *<urn:axmedis:0000:obj:0123…>*, while aggregation_10231 is the aggregation among the provided object and its digital representations.

According to the present EDM model, some aggregated content present in ECLAP cannot be fully exported to Europeana. In fact, by following the aggregation schema allowed by EDM the information about (i) the temporal segments of media involved in playlists, (ii) semantic information related to annotations and synchronizations modeled in MyStoryPlayer, (iii) the full courses; cannot be directly mapped into EDM. On the other hand, they continue to be additional features of ECLAP Content Aggregator with respect to the Europeana model and service. ECLAP also presents the management of social network and thus the several relationships with users.

4.Contribution of my thesis work and innovative aspects

The data available on the web in Semantic Web formats has typically been both eclectic and relatively small, and closely linked the interests of particular researchers. In the past years, however, the quantity and scope of data published on the public semantic web has exploded, and the size of the semantic web is now measured in the billions of assertions. It is a significant and growing resource for applications which depend on web-based resources for some or all of their knowledge.

The client-side technologies used in Web 2.0 development include Ajax and JavaScript frameworks such as YUI Library, DojoToolkit, MooTools, jQuery, ExtJS and Prototype JavaScript Framework. Ajax programming uses JavaScript to upload and download new data from the web server without undergoing a full page reload.

To allow users to continue to interact with the page, communications such as data requests going to the server are separated from data coming back to the page (asynchronously). Otherwise, the user would have to routinely wait for the data to come back before they can do anything else on that page, just as a user has to wait for a page to complete the reload. This also increases overall performance of the site, as the sending of requests can complete quicker independent of blocking and queuing required to send data back to the client.

The data fetched by an Ajax request is typically formatted in XML or JSON (JavaScript Object Notation) format, two widely used structured data formats. Since both of these formats are natively understood by JavaScript, a programmer can easily use them to transmit structured data in their web application. When this data is received via Ajax, the JavaScript program then uses the Document Object Model (DOM) to dynamically update the web page based on the new data, allowing for a rapid and interactive user experience. In short, using these techniques, Web designers can make their pages function like desktop applications. For example, Google Docs uses this technique to create a Web based word processor.

As a widely available plug-in independent of W3C standards (the World Wide Web Consortium is the governing body of web standards and protocols), Flash is capable of doing many things that were not possible pre-HTML5, the language used to construct web pages. Of Flash's many capabilities, the most commonly used in Web 2.0 is its ability to play audio and video files. This has allowed for the creation of Web 2.0 sites where video media is seamlessly integrated with standard HTML.

Figure 4.1: Semantic Web

In addition to Flash and Ajax, JavaScript/Ajax frameworks have recently become a very popular means of creating Web 2.0 sites. At their core, these frameworks use the same technology as JavaScript, Ajax, and the DOM. However, frameworks smooth over inconsistencies between web browsers and extend the functionality available to developers. Many of them also come with customizable, prefabricated 'widgets' that accomplish such common tasks as picking a date from a calendar, displaying a data chart, or making a tabbed panel.

On the server side, Web 2.0 uses many of the same technologies as Web 1.0. Languages such as PHP, Ruby, Perl, Python, as well as JSP and ASP.NET are used by developers to output data dynamically using information from files and databases. What has begun to change in Web 2.0 is the way this data is formatted. In the early days of the Internet, there was little need for different websites to communicate with each other and share data. In the new "participatory web", however, sharing data between sites has become an essential capability. To share its data with other sites, a website must be able to generate output in machine-readable formats such as XML (Atom, RSS, etc.) and JSON. When a site's data is available in one of these formats, another website can use it to integrate a portion of that site's functionality into itself, linking the two together. When this design pattern is implemented, it ultimately leads to data that is both easier to find and more thoroughly categorized, a hallmark of the philosophy behind the Web 2.0 movement.

In the last years many social networks grew up, making more attractive the use of internet and changing completely the way of fruition of multimedia objects, the way exchanging messages between users, the way of organize and sharing contents, and so on…also bringing to the usage of these new technologies on mobile devices like tablet or smart phones.

As a consequence of this, in this thesis new technologies and solutions have been developed, conducting the research in Best Practice Network environment. The needs were to project and develop a web based solution (models and tools) for creating relationships/synchronizations, for navigating among audiovisual media, and for saving and sharing user navigations. Some of the relationships among the audiovisual content may be established since the shooting phase of some event, or later during a qualified authoring phase by experts. On the other hand, final users, as students and researchers, need to provide their own contributions performing their own analyses, providing comments, creating annotations, and thus producing and sharing their experiences and relationships.

According to above described situation, the following requirements have been collected regarding the executable aggregation facilities of ECLAP, based on: playlists, collections, courses, annotations, and audiovisual synchronizations. These aggregations are finally exported towards Europeana as aggregations and their corresponding single entities, via a mapping and a simplification.

In this thesis, the aggregation models and tools on ECLAP are presented, focusing the attention on the solution MyStoryPlayer/ECLAP. The major contributions of this work to the improvement of the state of the art are as follows:

- The aggregation models at the state of the art does not cover all the solutions that are presented in this thesis, especially regarding playlists and collections. These solutions allow relating metadata to the new aggregated objects. These aggregations are finally exported towards Europeana as aggregations and their corresponding single entities, via a mapping and a simplification.

On this regard, the ECLAP users (teachers and students) have to be mainly capable of:

- creating and sharing a play lists, collections and courses with their own classification metadata, while the single content item has to preserve its own classification and identification.
- creating playlists as a sequence of segments/sub-segments of audio and/or video content or timed images without modifying the content sources, and thus that can be located on different servers.
- executing playlists with a semantic presenting them in a time sequence comprised of a combination of: segments of video identified from the whole video file, segments of audio identified from the whole audio file, presentation of images for a defined number of seconds.
- synchronizing video/audio segments and images, for example defining time relationships among them and the possibility of passing from one resource to another, via a non linear navigation.
- recording the navigation experiences of the users among the synchronizations of video, audio and images defined in the previous point.
- rating, commenting, promoting single content items, playlists, collections and courses.
- managing the e-learning course as a sequence of activities including, a set of content items, text for presenting and gluing them. Content items can be: audio, video, slide, documents, collections, playlists, excel file, tools. questionnaires, polls, etc.

In addition, the teachers can organize their courses in lesson units. Each lesson unit may contain contextual text and a set of digital resources including collection, playlists, annotations, etc.

Focusing more on MyStoryPlayer solution proposed in this thesis, the main aims produced are:

- The annotation models at the state of the art are not satisfactory to model all the aspects needed by the MyStoryPlayer as highlighted in the rest of the thesis. Therefore, the definition of MyStoryPlayer/ECLAP annotation model has been performed. It is an RDF based annotation [RDF] model to formalize the relationships among audiovisual. The MyStoryPlayer model solution is grounded on temporal patterns which are conformant with the concepts adopted in the education and training scenarios. The proposed RDF based annotation constructs can be progressively obtained by a client player to change the audiovisual context in real time, including synchronized and related audiovisual, thus avoiding the reconstruction of the whole web page. A part of the proposed MyStoryPlayer/ECLAP annotation model is exported as Linked Open Data using Open Annotation data model developed by a W3C community. The MyStoryPlayer/ECLAP annotation model is a part of the ECLAP ontological model described into [Bellini and Nesi, 2013], to specifically manage the audiovisual annotations.
- A user's experience can be regarded as a possible trace on the combination of annotations and user's actions about the interactive navigation performed during the playing of synchronized and annotated audiovisual (clicks and context change at specific time instants). The state of the art in this case is fully absent and thus the definition of a formal trace to save and share experiences on playing and navigating among synchronized and annotated audiovisual has been performed. The produced experiences can be saved and re-proposed to other users, thus replicating the user experience without creating a specific set of annotations on the same content of other annotations and audiovisual.
- Design and development of algorithms and tools for shortening the production of audiovisual relationships/annotations, mining the audiovisual metadata received with the aim of suggesting the possible relationships among audiovisual to the expert users during the production and authoring of the relationships. The tools have been collaboratively used by the ECLAP community for creating a set of relationships among audiovisual, currently used for education and training.

- Design and development of the MyStoryPlayer solution. It includes (i) a server at service of the ECLAP Social Network for producing, collecting and distributing related and annotated audiovisual content; and a service for user experience recording and share, and for searching annotations and relationships; and (ii) a client tool, called MyStoryPlayer Client, which provides support for navigating among the related and annotated audiovisual by starting from a reference content. MyStoryPlayer allows the user to interact and analyzing the several relationships among audiovisual: clicking on the annotations, changing the context and producing its own personal experience among played segments of the proposed network of relationships. The personal constructed experiences can be saved and shared with other users. MyStoryPlayer is one of the annotation tools promoted by Europeana, European digital library of cultural heritage content [http://pro.europeana.eu/web/guest/thoughtlab/enriching-metadata].
- Provide a solution to improve the quality for synchronous rendering of audiovisual content in presence of low bandwidth conditions. In those cases, if several videos are played on the same web page progressively downloaded via HTTP protocol they are typically affected by large diverging delays. The problems of synchronization are even higher in presence of: direct jump backward and forward with respect to the play execution time; swap from one video to another (i.e., master and annotation); back trace along the stack of performed swaps. The aim is to keep the desynchronizing limited. This is more critical in presence of low bandwidth and long video duration where relevant delays need to be corrected.

The MyStoryPlayer/ECLAP has been developed for supporting ECLAP social learning on a best practice network environment, the performing art content aggregator of Europeana.

5. MyStoryPlayer embedded in ECLAP Scenario

There are several edutainment fields where multi-camera shooting is a traditional way to grab and **report events and activities**. This happens in: (i) performing arts and news (taking different points of view of the same scene, recording workshops and master classes from different points of views); (ii) medical imaging for training, where the surgical action is taken from inside endoscopic camera, from the outside to show the action of the chirurgical team, and plus other views on monitoring instruments; (iii) sport events (offering multiple views, and giving the possibility to the user to select them), and in (iv) some TV shows such as the big brother.

Moreover, there are many other applications in which there is the need of aligning/relating different audio visual segments to play them together for comparing the scenes even if these have not been taken from the same event. That kind of activity is performed for enriching details, adding **comparative examples** and comments, providing alternatives. Examples are in the: (a) performing arts and films analysis (comparing and/or putting in evidence different posture and gesture of actors, different performance of the same opera, director and scenic design citations, alternatives scenes, etc.), (b) music education and training (comparison with the teacher, with previous performances, providing correction), (c) sport education and training (comparison with competitors, against correct posture and gestures; comparing different performances), (d) medical and surgical training, (e) public speech training (showing different point of view of the same event and human behavior), etc.

Synchronizations and comparison among audiovisual media segments can be produced by the creation of media relation and then annotations, which for the educational point of view can be regarded as a constructivist interaction and experience. A generalization of these aspects also include the annotation of an audiovisual segment with another shorter/longer one (comparison, correction, etc.), the association of an image to a video segment or instant to be shown for a while, the possibility of jumping from a related audio visual to another (with the corresponding change of context, that should show the annotations of the latter video and not those of the former) and may be jumping to the former with a Back button as in a browser; the possibility of recording the sequences of these actions to share them to students and colleagues, etc. Thus, a large set of possible experiences and combinations can be produced by exploiting a limited number of common audiovisual, similarly to the user navigation on the set of html pages on the web.

The rendering of multiple video streams synchronized together presents complexities if one would like to avoid delay in the change of context. For example, passing from a multi camera view in which 4 video are synchronously in execution to a different set by clicking on a proposed annotation. Moreover, the sophistication of the user interaction for establishing relationships among audio-visual, the navigation among audiovisual (change of context), the experience recording and playback the experiences, and needs of precise synchronizations make complex the design and implementation of a complete solution. For the same reason, these latter requirements lead to exclude the possibility of realizing the solution by using a preassembled streaming or the adoption of simple web pages containing synchronized playable audiovisual produced HTML.

5.1 Recalls

During the past three years my work has been focused on studying, projecting and developing solutions about ontological models, aggregated content on best practice network and semantic players able in managing different kinds of relations among media, with a different way of reproduction depending on time, type of relations and user interaction with the player itself.

As a consequence of this large range of topic, many technologies have been studied, in order to better understand which direction had to be taken during the developing phase.

Regarding Ontologies, OWL, RDF have been studied, and in particular the query language SPARQL, useful to access the information.

5.1.1 OWL

The W3C Web Ontology Language (OWL) is a Semantic Web language designed to represent rich and complex knowledge about things, groups of things, and relations between things. OWL is a computational logic-based language such that knowledge expressed in OWL can be reasoned with by computer programs either to verify the consistency of that knowledge or to make implicit knowledge explicit. OWL documents, known as ontologies, can be published in the World Wide Web and may refer to or be referred from other OWL ontologies. OWL is part of the W3C's Semantic Web technology stack.

In detail, OWL specifications propose three variants of increasing expressiveness (i.e., each of them is an extension of the previous sub-language) of the OWL language:

OWL-Lite, OWL-DL and **OWL-Full**. OWL-Lite was conceived as a light version of OWL with a low expressiveness aimed at supporting the hierarchical classification and simple properties. OWL-DL is based on a strict correspondence with Description Logic. This allows to maximize the expressiveness (it includes all OWL constructs with an expressiveness of SHOIN(D) such that only specific combinations of restrictions are not allowed), while preserving the advantages of being formally grounded to Description Logic (e.g., the availability of practical algorithms with known complexity). Description Logics (DLs), in fact, are a family of logics that are decidable fragments of the First Order Logic (FOL) with desirable properties. Both OWL-Lite and OWL-DL are based on DL, while OWL-Full was mainly conceived for compliance with RDF, that it semantically extends. It enriches FOL with new constructs but it is undecidable. OWL 2 DL has an expressiveness of SROIQ(D).

OWL allows to describe classes (corresponding to DL concepts), e.g., Person. Classes can be organized hierarchically by exploiting the *is a* relationship, so that sub-classes inherit super-class properties. For example a class Child is a sub-class of the class Person. The root class of this hierarchy is the owl:Thing class, while the common leaf is the owl:Nothing. Moreover, OWL allows the representation of instances and of properties. Instances correspond to DL individuals, while properties, corresponding to DL roles, are oriented relationships between two objects and, as such, allow to specify a domain and a range.

For example Mark is an individual of the class Person, while *has_person_pet* is a property having as domain the class Person and as range the class Pet. In detail, properties can be of two types: object properties between instances of two classes (e.g., *has_person_pet*) and data type properties between class instances and literal data (e.g., *is_person_age_old*, where age is of type integer). Moreover, each property can be defined as symmetrical (in case the relationship is bidirectional), transitive (i.e., if p(x,y) and p(y, z), then p(x,z), where p is the property and x, y and z individuals) or functional (i.e., for each x in the domain, only one y in the range is allowed) and a property can be defined as the inverse of another property (i.e., if p(x, y), the inverse of p, p^{-1} is such that p^{-1}(y, x)). For example, the object property *is_person_married_with* (having as both domain and range the class Person) is symmetric; an example of transitive property is the object property *is_elder_than*; a functional property could be the datatype property *is_person_age_old;* while the object property *is_wife_of* is the inverse of the property *is_husband_of*. Finally, several operators like union, intersection, complement, enumeration, disjointness, quantifiers, cardinality can be used for defining special classes as composition of others, as well as to specify special properties among classes (e.g., disjointness).

5.1.2 RDF

RDF is a standard model for data interchange on the Web. RDF has features that facilitate data merging even if the underlying schemas differ, and it specifically supports the evolution of schemas over time without requiring all the data consumers to be changed.
RDF extends the linking structure of the Web to use URIs to name the relationship between things as well as the two ends of the link (this is usually referred to as a "triple"). Using this simple model, it allows structured and semi-structured data to be mixed, exposed, and shared across different applications.
This linking structure forms a directed, labeled graph, where the edges represent the named link between two resources, represented by the graph nodes. This graph view is the easiest possible mental model for RDF and is often used in easy-to-understand visual explanations.

SMIL, HTML5, Silverlight and ActionScript have been studied in order to find the better solution to project and develop a proper player which satisfied our requirements, posing the attention on the interactivity with users and flexibility.

5.2 MyStoryPlayer media relationship model

The MyStoryPlayer/ECLAP scenarios need to be grounded by a suitable and expressive annotation model semantically conformant with the concepts adopted in the education and training scenarios, where different audiovisual resourced are formally related, frequently presenting synchronizations. This means to build a formal annotation model which has to provide a precise semantics and behavior for the constructs the player is following during the media playing and the related execution of annotations, and also in the cases of user's interaction and actions (that in most cases can create a change of context, destroying the video stream caching). For these reasons, the MyStoryPlayer solution and formal model have to provide support to: (i) create relationships among audiovisual media with multiple views for play and rendering and specific semantic associated with their play (execution of relationships at play time), (ii) jump to a different context (selecting the next central audiovisual at a time instant) and obtaining from the server a new set of streamed annotations from the new context, (iii) jump back in the stack of actions, thus keeping trace of the performed context changes, (iv) save and share the user experiences built on the basis of the performed plays and context changes among the audiovisual. The performed navigations (i.e., play, jump, back, swap to another video) among media relationships, executions and specific user actions to change the context (passing from one video to another) allows the users to produce their own experience in the navigation and play among audiovisual annotations, that can be saved and shared with other users. Moreover, the MyStoryPlayer facilitates the production of media relationships in the context of ECLAP social network, thus granting the production of a large mesh of browse-able media relationships.

According to the related work and the above description, the peculiar aspects of MyStoryPlayer and differences with respect to the state of the art have been remarked. Each relationship may have a descriptor associated to provide details about the contextual information: the reasons for the relationships, the description of the scene, etc. The MyStoryPlayer approach is quite different with respect to the annotation tools mentioned in Chapter 2, where the annotations to an audiovisual mainly refer to the scene content. In MyStoryPlayer, the descriptors are associated with the relationships between media and thus to the logic semantic related to the established relationships. For example, this audiovisual segment reports "*left view of the central view*", this segment reports the "*Dario Fo actor doing the same gesture but in a different context*", this segment is reporting a "*different actor interpreting the same scene of the previous one"*, this "*the dress used by Amleto has been taken out of the draft drawings, please remark last minute changes"*.

In the literature a wide effort has been devoted to model temporal relationships [Allen, 1981] and temporal logics [Bellini et al., 2009]. These aspects are out of the scope of this thesis, while a temporal notation is used to formalize the temporal relationships among audiovisual segment annotations.

In the following subsections, the main kinds of media relationships and their related meaning at the execution time are presented. In MyStoryPlayer, an audiovisual content has an associated executable timeline with a given duration. This is true for video and audio, while images can be shown for a given time duration. According to the analysis performed for the above presented scenarios, the typical relationships that can be needed to formalize the presentation of audiovisual in a lesson can be described as:

- **Explosion relationship** which consists in associating at a given time instant of a master audiovisual segment the execution of a second audiovisual segment, interrupting the execution of the former. This kind of relationship can be used when the teacher wants to explain a concept during a story thus opening a sort of parenthesis.
- **Sequential relationship** which formalizes that one audiovisual has to be executed after another. At the end of a given media execution, the sequential one will start automatically, changing the context on the player. This kind of relationship can be used when there are many videos parts of the same video; for example, Act 1 and Act 2, lesson 1 and lesson 2.

- **One2One relationship** which consists of relating an audiovisual segment to annotate another audio visual segment, with the aim of showing the former segments synchronously when the latter is executed, and not the opposite.

These relationships between audiovisual can be composed to create more complex structures and relationships among audio-visual that can be navigated and played.

Let us now formalizing the MyStoryPlayer model:

$MSP = < Media, O2O, Exp >$ where:

- $Media = \{M_1, \dots, M_N\}$ is the set of audiovisual content (video, audio and images) that are subject to **r**elationships;
- $d(M)$ where $M \in Media$ represents the duration of media M, for images the duration is considered unlimited; On the other hand, the images can be rendered according to a specified duration into the defined relationship;
- $M^{[s,e]}$ where $M \in Media, s \geq 0, e \leq d(M)$ represents the section of media M starting at time *s* and ending at time *e*;
- $M^{[t]}$ where $M \in Media, t \geq 0, t \leq d(M)$ represents the media frame that can be seen at time *t*;
- $O2O = \{\left(M_A^{[s_A,e_A]}, M_B^{[s_B,e_B]}\right) \dots\}$ is the set of One2One relationships, where media section $M_A^{[s_A,e_A]}$ is related with media section $M_B^{[s_B,e_B]}$
- $Exp = \{\left(M_A^{[a]}, M_B^{[s_B,e_B]}\right) \dots\}$ is the set of Explosive relationships, where media M_A is exploded at time instant *a* with media section $M_B^{[s_B,e_B]}$

Moreover two useful projection functions are:

- $O2O[M_a] = \left\{\left(M_a^{[x_1,y_1]}, M_{a_1}^{[s_1,e_1]}\right), \left(M_a^{[x_2,y_2]}, M_{a_2}^{[s_2,e_2]}\right) \dots \left(M_a^{[x_n,y_n]}, M_{a_n}^{[s_n,e_n]}\right)\right\} \subseteq O2O$ is the subset of $O2O$ with the One2One relationships related to media M_a;
- $Exp[M_a, s, e] = \left\{\left(M_a^{[x_1]}, M_{a_1}^{[s_1,e_1]}\right), \left(M_a^{[x_2]}, M_{a_2}^{[s_2,e_2]}\right) \dots \left(M_a^{[x_n]}, M_{a_n}^{[s_n,e_n]}\right)\right\} \subseteq Exp$, $s \leq x_1 < x_2 < \dots < x_n \leq e$ is the subset of Exp with Explosive relationships related to media M_a that are within time instants *s* and *e*;

Other useful definitions are:

Definition Media concatenation operator:

$M_A \oplus M_B$ represents the media obtained by concatenating media $M_A$with media M_B.

Definition Media translation operator:

$M_A|_x$ represents the media obtained by translating media M_A x seconds in the future; for example $V_A^{[5,60]}\Big|_{30}$ presents the section from second 5 to 60 of video V_A after 30 seconds.

In the following subsections, the media relationships are analyzed from the point of view of their semantic meaning. Without losing generality, in the following examples, we are talking about video, similar issues can be stated for audio tracks, and images with a given duration.

5.2.1 Explosion Relationship

Explosion relationships aim at expanding the execution time line of a video (e.g., V_1) with the insertion of an identified segment of a second video, V_2; just returning to the execution of V_1once the V_2segment execution is completed (see Figure 5.1). This model is equivalent to the action of opening a parenthesis where some aspects can be recalled, and then closing it to restart from the point where the parenthesis had been opened. The **Explosion** relationship can be used to explain a single time instant with an expanded scenario; to show possible cut scenes, to stress possible variants, to insert comments by the director, to explode a single time instant with a more complex scenario, etc.

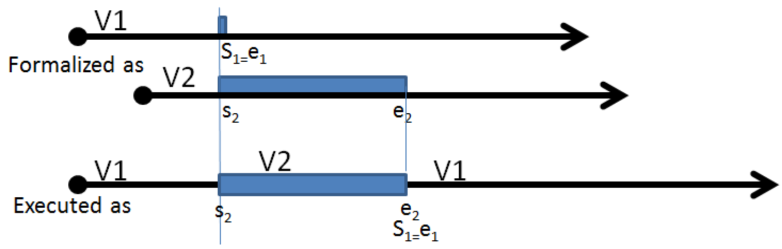


Figure 5.1: Explosion relationship: Formalized as above and Executed as below: at V1.s_1 the player starts reproducing V2 from s_2 until point e_2; then the reproduction switches back to V1, just a time instant after s_1.

The Explosive relationship consists in the association of a video time instant of media/video to a second media segment. At the play of V_1, the relationship is executed by starting the play the second video segment. According to the above model, the screen rendering of an audiovisual media is a function $\mathbb{M}[M, Exp]$ that given a media $M \in Media$ and a set of Explosive relationships Exp provides the media to be played on the main screen considering the explosion relationships that are present in Exp regarding media M. Thus, it can be defined using the recursive function $\mathcal{M}$, $\mathbb{M}[M, Exp] = \mathcal{M}[M, Exp, 0, d(M)]$ defined as:

$$\mathcal{M}[M, Exp, s, e] = M^{[s,a_1]} \oplus \mathcal{M}[M_1, Exp, s_1, e_1] \oplus M^{(a_1,a_2]} \oplus \mathcal{M}[M_2, Exp, s_2, e_2] \oplus \ldots \oplus \mathcal{M}[M_n, Exp, s_n, e_n] \oplus M^{(a_n,e]}$$

Where:

- $Exp[M, s, e] = \left\{\left(M^{[a_1]}, M_1^{[s_1,e_1]}\right), \left(M^{[a_2]}, M_2^{[s_2,e_2]}\right) \ldots \left(M^{[a_n]}, M_n^{[s_n,e_n]}\right)\right\}$

  with $s \le a_1 < a_2 < \cdots < a_n \le e$
- considering that: $\mathcal{M}[M, Exp, s, e] = M^{[s,e]}$ if $Exp[M, s, e] = \emptyset$

On the basis of the model, we can see the following example in which a set of explosive relationships are associated with set of media:

If $Exp = \left\{\left(V_1^{[10]}, V_2^{[15,60]}\right), \left(V_1^{[60]}, V_2^{[30,80]}\right), \left(V_2^{[40]}, V_3^{[0,30]}\right)\right\}$, and $d(V_1) = 100, d(V_2) = 80, d(V_3) = 30$, then:

$$\mathbb{M}[V_2, Exp] = V_2^{[0,40]} \oplus V_3^{[0,30]} \oplus V_2^{(40,80]} \quad \text{and}$$

$$\mathbb{M}[V_1, Exp] = V_1^{[0,40]} \oplus (V_2^{[15,40]} \oplus V_3^{[0,30]} \oplus V_2^{(40,60]}) \oplus V_1^{(10,60]} \oplus (V_2^{[30,40]} \oplus V_3^{[0,30]} \oplus V_2^{(40,80]}) \oplus V_1^{(60,80]}$$

5.2.2 Sequential Relationship

Sequential relationships are used to model the sequences of media, for example when different videos are taken from the same event as sequential parts. Thus, there is the need of creating a sequence of audiovisual reproductions, as in the playlists. The associated behavior of this relationship puts the new video in place of the previous one and loading and showing the set of relationships associated with the new context. The formal definition of this relationship consists in the association with the last time instant of a video to the start of the successive video. The Sequential relationship has not been modeled as a primary operator and set of the MyStoryPlayer model since it can be derived from the other operators. In this example, in order to sequence video $V_1$with video V_2, an explosive relationship is put at the last instant of video $V_1$with

$$Exp = \left\{\left(V_1^{[d(V_1)]}, V_2^{[0,d(V_2)]}\right)\right\},$$

in this case

$$\mathbb{M}[V_1, Exp] = V_1^{[0,d(V_1)]} \oplus V_2^{[0,d(V_2)]} \oplus V_1^{(d(V_1),d(V_1)]} = V_1^{[0,d(V_1)]} \oplus V_2^{[0,d(V_2)]} = V_1 \oplus V_2$$

5.2.3 One2One Relationship

According to the MyStoryPlayer model, it is possible to associate with a specific time instant of an audiovisual content a time segment of other audiovisual content, establishing a One2One relationship. For example, it is possible to relate a video segment (starting from 1':30'' to 10':00'' with respect to its beginning) to another video segment (from 2':15'' to 7':50''). The possible cases are reported in Figure 5.2, where media segments are marked with their start *s* and end *e*, and are aligned to their starting points (from which their synchronous execution has to start), and they are taken from two different media of different length/duration. This model allows the execution of synchronized audiovisual media by starting from a given time instant. For example, to

- compare different performances of the same opera with different or the same actors, different years,..;
- compare different interpretations of the same actions or of different actions;
- show/see different point of views or what happens at the same time in another place; nice for: presentations, big brother events, sport events, theatrical show, related historical scenarios, etc.;
- show a video with the scene without visual effects; nice for: backstage presentations in training and for infotainment;
- remind past related events, or link to possible future events; useful for aside advertising;
- provide the comment of the director;
- etc.

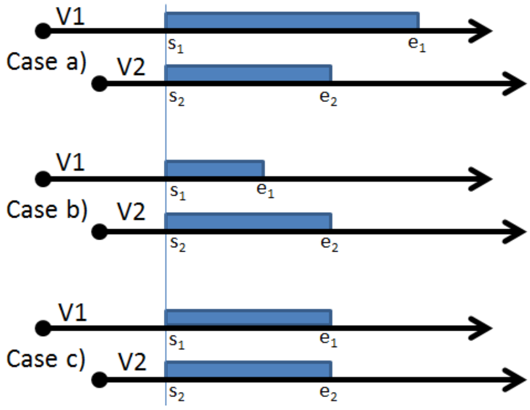


Figure 5.2: One2One Relationships between two audiovisual, the main cases.

The formal definition of the relationships consists of the association of a video segment of $V_1$to a video segment V_2. At the play of video V_1, the relationship is executed by starting the play of $V_2$from s_2. The V_2segment starts synchronously playing from s_2 time instant only when its master $V_1$is put in execution and reach s_1 time instant. The determined relationship between the two videos is asymmetrical since *relating* V_1*with* $V_2$is semantically different with respect to relate V_2*with* V_1. During the execution, the temporal relationships (time segments) as well as the video screens are proposed to the user to allow them changing the context.

A specific behavior of the player occurs in the different cases reported in Figure 5.2:

- **Case a)** during playing the screen of $V_2$is frozen at time instant e_2, then the screen of $V_2$is removed to leave space at other relationships.
- **Case b)** during playing at e_1 time instant the synchronous execution of V_2stops. If the user wants to watch the entire V_2segment, has to switch to it by clicking on the segment or related audiovisual; thus loading a new context based on $V_2$and allowing the reproduction of segment s_2,e_2. Then the screen of $V_2$is removed to leaving space at other relationships.
- **Case c)** this case corresponds to synchronized play of the two video segments.

The definition of the screen rendering for this One2One relationship is defined as function $\mathbb{S}[M, O2O]$ that given a media $M \in Media$ and a set of One2One relationships $O2O$, provides the set of media to be played on the side screens when media M is played on the main screen:

$$\mathbb{S}[M, O2O] = \{M_1^{[s_1, \min(e_1, s_1+y_1-x_1)]}\Big|_{x_1} \dots M_n^{[s_n, \min(e_n, s_n+y_n-x_n)]}\Big|_{x_n}\}$$

Where: $O2O[M] = \left\{\left(M^{[x_1,y_1]}, M_1^{[s_1,e_1]}\right), \left(M^{[x_2,y_2]}, M_2^{[s_2,e_2]}\right) \dots \left(M^{[x_n,y_n]}, M_n^{[s_n,e_n]}\right)\right\}$

To be noted that the video is translated to the starting point of the annotation x_i and that the end of the annotation media is updated to $\min(e_i, s_i + y_i - x_i)$ in order to view the target media M_i for the duration of the annotation $(y_i - x_i)$ or until the end of the target video section if less.

According to the above cases, an example including both Case (a) and (b) with:

$$O2O = \{\left(V_1^{[10,15]}, V_2^{[20,35]}\right), \left(V_1^{[40,70]}, V_3^{[80,90]}\right)\}$$

We have that the side screen video for $V_1$are

$$\mathbb{S}[V_1, O2O] = \{V_2^{[20,25]}\Big|_{10}, V_3^{[80,90]}\Big|_{40}\}$$

Then two videos, $V_1$and V_2, are synchronously executed when the V1 is played (Case (c)):

$$O2O = \{\left(V_1^{[0,d(V_1)]}, V_2^{[0,d(V_2)]}\right)\}, \text{ with } d(V_1) = d(V_2)$$

We have

$$\mathbb{S}[V_1, O2O] = \left\{V_2^{[0,\min(d(V_1),d(V_2))]}\Big|_0\right\} = \left\{V_2^{[0,d(V_2)]}\Big|_0\right\} = \{V_2\}$$

However, when for a media both explosive and One2One relationships are present, then the media for the side screens depends on the video section currently played on the main screen which can be a section of a media, e.g. $M^{[s,e]}$, for this reason the Side Screen presentation function needs to be extended as follows.

$\mathbb{S}'\left[M^{[s,e]}, O2O\right]$ is a function that given a media section [s,e] of $M \in Media$ and a set of One2One relationships $O2O$ provide the set of media to be played on the side screens when playing the media section.

Considering the one2one annotations that are present on media M

$$O2O[M] = \left\{\left(M^{[x_1,y_1]}, M_1^{[s_1,e_1]}\right), \left(M^{[x_2,y_2]}, M_2^{[s_2,e_2]}\right) \dots \left(M^{[x_n,y_n]}, M_n^{[s_n,e_n]}\right)\right\},$$

have to be considered only the ones that are active in time interval [s,e] and also should be restricted the section being annotated to cover the [s,e] interval, considering that:

$$[x_i, y_i] \cap [s, e] = [\max(x_i, s), \min(y_i, e)] = [\bar{x}_i, \bar{y}_i]$$

we have

$$O2O\left[M^{[s,e]}\right] = \left\{\ldots \left(M^{[\bar{x}_i,\bar{y}_i]}, M_i^{[s_i+\bar{x}_i-x_i,\, e_i]}\right), \ldots\right\}$$ considering only terms having $\bar{x}_i \leq \bar{y}_i$

Where the starting point of the target media is updated to $s_i + \bar{x}_i - x_i$ to consider the case when s is in the middle of interval $[x_i, y_i]$ and thus the part of target media for the duration from x_i to s have to be skipped.

Finally we have that:

$$\mathbb{S}'\left[M^{[s,n]}, O2O\right] = \{\ldots M_i^{[s_i+\bar{x}_i-x_i, \min(e_i, s_i+\bar{y}_i-x_i))]}\Big|_{\bar{x}_i-s} \ldots\}$$ considering only terms having $\bar{x}_i \leq \bar{y}_i$

To be noted that the media translation points are provided relative to time instant s.

For example if

$$O2O = \{\left(V_1^{[10,20]}, V_2^{[20,60]}\right), \left(V_1^{[40,70]}, V_3^{[10,90]}\right), \left(V_1^{[60,80]}, V_4^{[10,50]}\right), \left(V_1^{[90,110]}, V_5^{[0,40]}\right)\}$$

$$\mathbb{S}'\left[V_1^{[50,100]}, O2O\right] = \{V_3^{[20,40]}\Big|_0, V_4^{[10,30]}\Big|_{10}, V_5^{[0,10]}\Big|_{40}\}$$

5.2.4 Reciprocal Synchronization Relationships

The above presented models can be combined for creating more complex relationships of synchronization in the presence of multicamera. In MyStoryPlayer, the Reciprocal Synchronization relationship implies a couple of relationships – e.g., *from* V_1 *to* V_2 *and vice versa*. This relationship can be useful in the event of a video related to same issue and with the arising need to synchronize them (different aspects, different point of views, etc.). There are many contexts where this can be applicable like for example in a didactical framework for theatre lessons. It is useful to have different points of view of the scene and have them played in a synchronized way; for performances and sport related events.

The described relationship of synchronization can be defined among several audiovisual (e.g., 8 camera views of the same events). In the case of *N* videos a total of N^2-N relationships are defined. This allows seeing synchronized camera regardless of which is the starting point. This allows to access at the full set of videos relationships by playing any of the involved videos. This kind of formalization can be useful to synchronize multiple views, independently of which one has been selected to play. Similarly to the One2One type, the lengths of synchronized media segments can be chosen independently. What is different with respect to One2One is the nature of the relationships which are mutual (in both sense) and can involve *N* media.

5.3 Semantic Model of MyStoryPlayer

As described in the previous chapters, some scenarios and solutions are focused on associating formal descriptors to audiovisual content, so as to describe the included scene (the main aim is to allow indexing and retrieval of the scenes). In MyStoryPlayer, the focus is on the relationships among media content. The semantics of relationships among media describes some temporal associations among them for the purpose of their playing/execution. This approach can be very useful for didactical purposes, to compare, to explain, to annotate with another audiovisual, etc. The navigation among the relationships may describe a story: the aim of director, the linear time, the activities of a given actor, the movements of an object, etc. Therefore, the MyStoryPlayer may be used to create a huge amount of new personal experiences with the same content pool, accessing them from a given point and from that point navigating to an indefinite network of relationships. Thus, in order to allow replicating the experience, the MyStoryPlayer permits the recording and the sharing of such experiences.

The main idea of MyStoryPlayer is to put into the hands of the final users a tool for creating and navigating in a mesh of relationships, providing the user with full access to the media, which may refer to a set of relationships with their related segments. These relationships may be: (i) labeled to allow grouping and selecting/deselecting them, (ii) commented to index and search them. Moreover, the single media (audio, video and image) used in MyStoryPlayer has its own classification in the ECLAP archive based on ECLAP model which includes multilingual: Dublin Core, performing arts metadata, aggregation information, tags, taxonomical classification, and technical metadata [Bellini, Cenni, Nesi, 2012]. To this end, Figure 5.3 reports the semantic model of MyStoryPlayer tool and RDF database. RDF navigation and queries allow extracting the relevant segment of the knowledge to be shown and executed by the MyStoryPlayer tool. The relationship has a description that can be textual or may include additional ontology to describe the facts by using more accurate, complete, and specific information for domain and contexts. In ECLAP, that part is implemented as free text that can be searched.

Moreover, the flexibility of the RDF model can lead to the creation of customizable part of description in the form of semantic model, depending on the environment and application uses. The MyStoryPlayer model has not yet implemented this kind of part in the semantic model, since to this end many other tools can be used. Thus, MyStoryPlayer is open to accept additional contextual descriptions of scene.

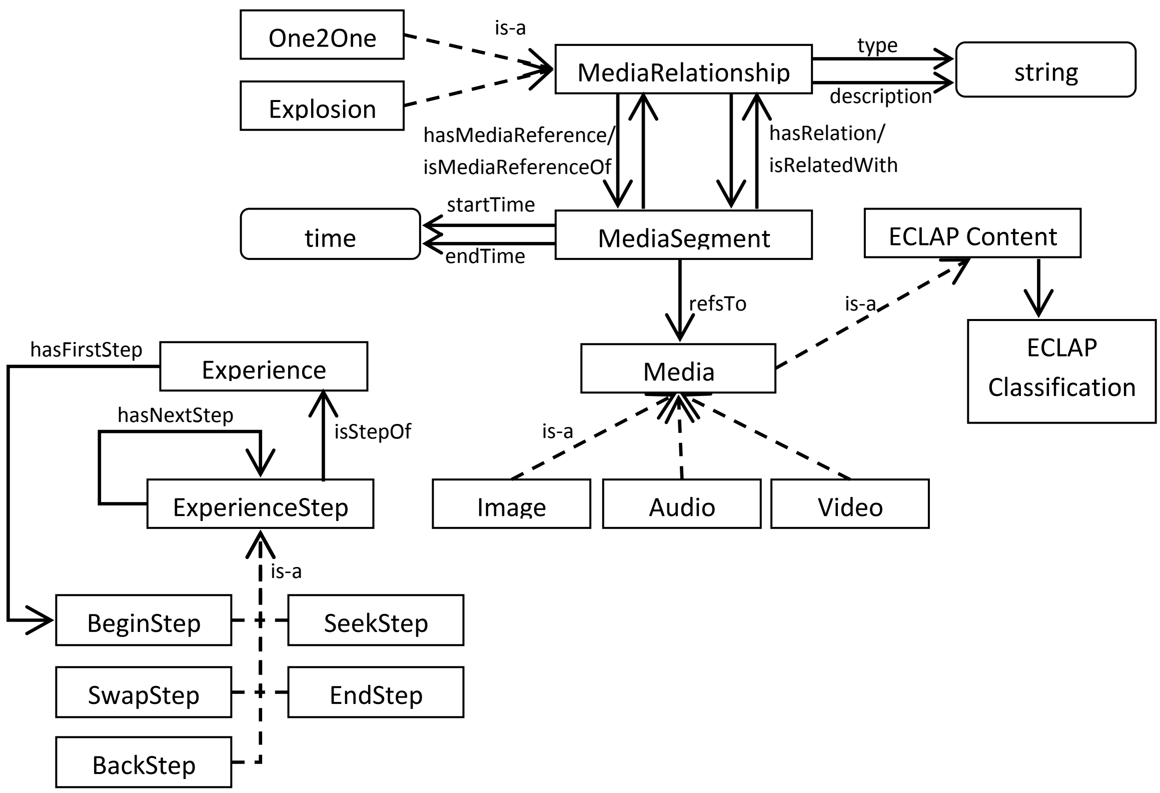


Figure 5.3: Semantic model of MyStoryPlayer

In the following is reported the description of the classes using the Manchester OWL syntax:

MediaRelationship = (*<u>isRelatedWith</u>* exactly 1 *MediaSegment*) and
(*<u>hasMediaReference</u>* only *MediaSegment*) and
(<u>label</u> max 1 string) and
(<u>description</u> max 1 string) and
(*createdBy* exactly 1 User) and
(<u>createdAt</u> exactly 1 dateTime)

MediaRelationship disjointUnionOf *One2One*, *Explosion*

MediaSegment = (*<u>refsTo</u>* exactly 1 *Media*) and
(<u>startTime</u> max 1 time) and
(<u>endTime</u> max 1 time)

Audio, Video, Image SubClassOf *Media*

Media SubClassOf (*ECLAPContent* and (<u>duration</u> exactly 1 *time*))

Experience = (*<u>hasFirstStep</u>* exactly 1 *ExperienceStep*) and
(<u>title</u> exactly 1 *string*) and

(description exactly 1 *string*) and

(date exactly 1 *date*)

ExperienceStep =(*hasNextStep* max 1 *ExperienceStep*) and (*isStepOf* exactly 1 Experience)

ExperienceStep disjointUnionOf *BeginStep,SwapStep, BackStep, SeekStep, EndStep*

BeginStep = *ExperienceStep* and

(*isFirstStepOf* exactly 1 *Experience*) and

(*mediaUri* exactly 1 Media) and

(clickTime exactly 1 time)

SwapStep = *ExperienceStep* and

(*swapRelation* exactly 1 *One2One*) and

(swapTime exactly 1 time)

BackStep = *ExperienceStep* and

(*backTo* exactly 1 *Media*) and

(backTime exactly 1 time) and

(backToTime exactly 1 time)

SeekStep = *ExperienceStep* and

(clickTime exactly 1 time) and

(seekToTime exactly 1 time)

EndStep = *ExperienceStep* and

(clickTime exactly 1 time)

Each audiovisual media may have multiple relationships and each of them with its own reference to other media segments, labels and description. Media relationships are divided into two specializations: *One2One, Explosion*. These relationships are semantically different and consequently they are interpreted differently by the MyStoryPlayer, giving the users more flexibility in both the creation and navigation phases. In the model is also present the part related with the representation of the user experience that can be saved and retrieved.

The relationships are made available also as linked data using the OpenAnnotation model as reported in [Bellini and Nesi, 2013], the media relationships are mapped to OpenAnnotation, in particular the 'isRelatedWith' property is mapped to the OA *hasTarget* property representing the media (or its part) on which the annotation/relationship is stated, while the 'hasMediaReference' property is mapped to the *hasBody* property as it describes the associated media that describe the main media being annotated. Moreover to indicate the kind of relationship (One2One or Explosive) and additional rdf:type indication is provided. The following is an example of an relationship/annotation available through Linked Open Data linking a media from time 29 to 227 with a One2One relationship with another media from time 67 to 119.

```
<rdf:RDF …>
 <oa:Annotation
rdf:about="http://www.eclap.eu/resource/annotation/SideAnnotation_136...">
   <rdf:type rdf:resource="http://www.eclap.eu/schema/eclap/One2One"/>
   <oa:hasTarget>
    <oa:SpecificResource>
     <oa:hasSource
rdf:resource="http://www.eclap.eu/resource/object/urn:axmedis:00...1"/>
     <oa:hasSelector>
      <oa:FragmentSelector>
       <rdf:value>t=npt:29,227</rdf:value>
       <dcterms:conformsTo rdf:resource="http://www.w3.org/TR/media-frags/"/>
      </oa:FragmentSelector>
     </oa:hasSelector>
    </oa:SpecificResource>
   </oa:hasTarget>
   <oa:hasBody>
    <oa:SpecificResource>
     <oa:hasSource
rdf:resource="http://www.eclap.eu/resource/object/urn:axmedis:00...2"/>
     <oa:hasSelector>
      <oa:FragmentSelector>
       <rdf:value>t=npt:67,119</rdf:value>
       <dcterms:conformsTo rdf:resource="http://www.w3.org/TR/media-frags/"/>
      </oa:FragmentSelector>
```

```
    </oa:hasSelector>
   </oa:SpecificResource>
  </oa:hasBody>
  <oa:hasBody>
   <cnt:ContentAsText>
    <cnt:chars>this is really interesting</cnt:chars>
    <dc:format>text/plain</dc:format>
   </cnt:ContentAsText>
  </oa:hasBody>
  <dc:type>acting style</dc:type>
  <oa:annotatedBy rdf:resource="http://www.eclap.eu/resource/user/229"/>
  <oa:annotatedAt>2013-02-13T09:32:09</oa:annotatedAt>
 </oa:Annotation>
</rdf:RDF>
```


The original description of the relationship using MyStoryPlayer model is:


```
<rdf:RDF …>
<msp:One2One rdf:about="http://...#Relation">
 <msp:type>acting style</msp:type>
 <msp:description>this is really interesting</msp:description>
 <msp:createdBy rdf:resource="http://www.eclap.eu/resource/user/229" />
 <ms:createdAt>2013-02-13T09:32:09</msp:createdAt>
 <msp:isRelatedWith>
  <msp:MediaSegment>
   <msp:refsTo rdf:resource="urn:axmedis:00…1" />
   <msp:startTime>00:00:29</msp:startTime>
   <msp:endTime>00:03:47</msp:endTime>
  </msp:MediaSegment>
 </msp:isRelatedWith>
 <msp:hasMediaReference>
  <msp:MediaSegment>
```

<msp:refsTo rdf:resource="urn:axmedis:00…2" />

<msp:startTime>00:01:07</msp:startTime>

<msp:endTime>00:01:59</msp:endTime>

</msp:MediaSegment>

</msp:hasMediaReference>

</msp:One2One>

</rdf:RDF>

5.4 MyStoryPlayer tool, user interface features

The exploitation of the above described relationships consists of the play of media and thus synchronous execution of different videos, audios and images. MyStoryPlayer supports the execution of complex relationships among audiovisual of different kinds: One2One, Explosion, Synchronization and Sequential.
On ECLAP portal there are many groups of videos from the same event, divided into sequential parts and taken in multicamera, related each other creating a playable structure on which the user can navigate view/play in an interactive and synchronized way through MyStoryPlayer facility. Figure 5.4 shows a simple example of synchronous and sequential relationships among videos. The example refers to a segment of the relationships established for modeling the theatrical laboratory by Toni Servillo, activity which took place at CTA Rome. This laboratory lasted several days and it has been recorded from three cameras (right side, frontal and left side), the video recording of each day is divided into many parts. In Figure 5.4, an intuitive representation is provided, the column of videos represents the videos taken at the same time (from left, centre and right camera, sometimes close-up shooting) while the raw of videos represent the successive recordings, which correspond to time (part, lessons, acts, days), etc. The established relationships create a playable structure ordered in time by sequence and synchronized by point of view. The result is a multi-views vision for each part of the sequence, as depicted in Figure 5.5.

In Figure 5.5, letters from (a) to (c) are indicating some entry points (examples of links that can be provided by the teacher to start playing the lesson). They are a way to access in the non-linear structure of relationships that may be executed by the MyStoryPlayer. Users may start playing the above structure of media by entering in it via anyone of the above media, as a click from a query result. For example, via the video indicated by (c); this is possible by accessing the MyStoryPlayer by using the small icon on that video 3 into the ECLAP portal, in any list of query results where that video is listed. Starting from the (c) media, the MyStoryPlayer is going to put as main video central video/part 1 (first day), playing the other synchronized videos (left and right) aside the main one, as depicted in Figure 5.4. According to the sequential relationships, at the end of the main video the execution passes to load the synchronized videos of the second part, and so on.

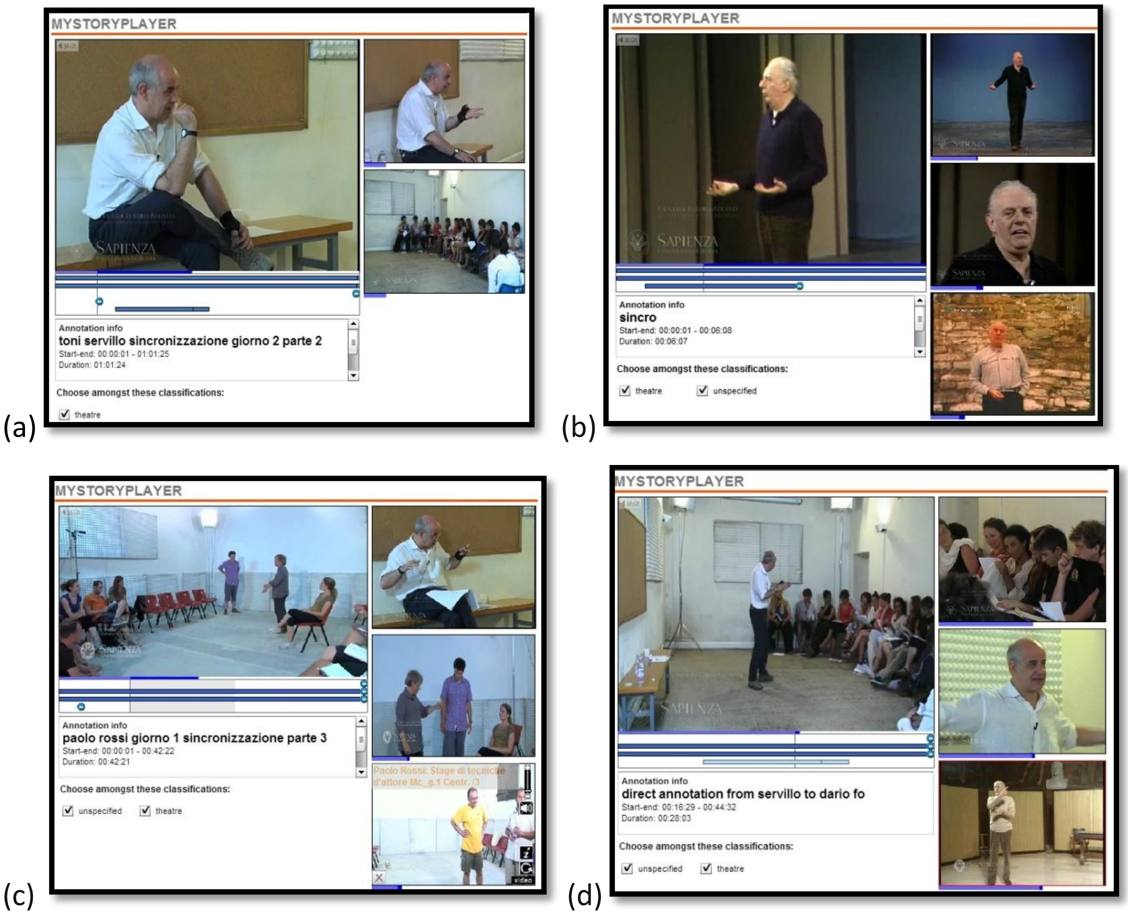


Figure 5.4: Example of MyStoryPlayer execution adopting as master point video the video marked with (a) in Figure 5.5. The screenshots depict the user point of view in different situations reached during the navigation among relationships and highlighted into Figure 5.5.

(a) two reciprocal synchronization relationships, one Exp and a One2One relationship;

(b) the explosive annotation jumped on the master position, becomes active and the scenario changed, going to Dario Fo synchronization of Miracolo di Gesu Bambino play; Once the explosive annotation is terminated the context returned in the context of (a);

(c) the user clicked on the One2One relationships in (a) going to the Paolo Rossi's Theatrical Lab. The grey zones overlapped on the time lines represents the length of relation which the media is played with;

(d) *the user came back in (a) situation from (c) and clicked on a video on left, thus changing the context by swap to new scenario (d) with a direct annotation to Dario Fo.*

According to Figure 5.4, the user can see on the left side the master media. The related audiovisual according to One2One relationships are dynamically listed on the right side: for example the left and right camera, the next explosive media to be played, etc. According to the above discussed relationship model and semantics, the media are streamed synchronously with the master media. The set of relationships among the involved videos are reported in Figure 5.5, the corresponding snapshot is reported in Figure 5.4. For example, in Figure 5.4 (a), under the master video the time bar reports the streaming progress and depicts the relationships with the other media: Explosive and One2One relationships (respectively depicted as circle in the timeline and as a segment inside a timeline) and as a Sequential relationship reported at the end of the time line as a small circle. Synchronizations are additional bars on the main timeline.

On the MyStoryPlayer user interface (that partially activates passing on the master with the mouse), the user may:

- click on the master time line to **jump** forward and backward on the play time;
- click on one of the right side media to **swap** it on master position, thus changing the media of the main context, that also implies the visualization and activation of the relationships associated to that media;
- click on the **back** button on the upper left corner to swap back;
- start, pause and stop the execution of the master media and thus of the whole rendering;
- start and stop of the experience recording;
- activate and/or regulate the audio coming from the different media sources (audio and video) which are synchronously executed;
- select labels (below the master position and timeline) and thus highlights the media which he/she is interested in, among those in execution on the right side;
- move the mouse over the several media to see their titles and other pieces of information;

- move the mouse over the time lines depicted for the several media to see the descriptions of the related media and identify them on the right side list.

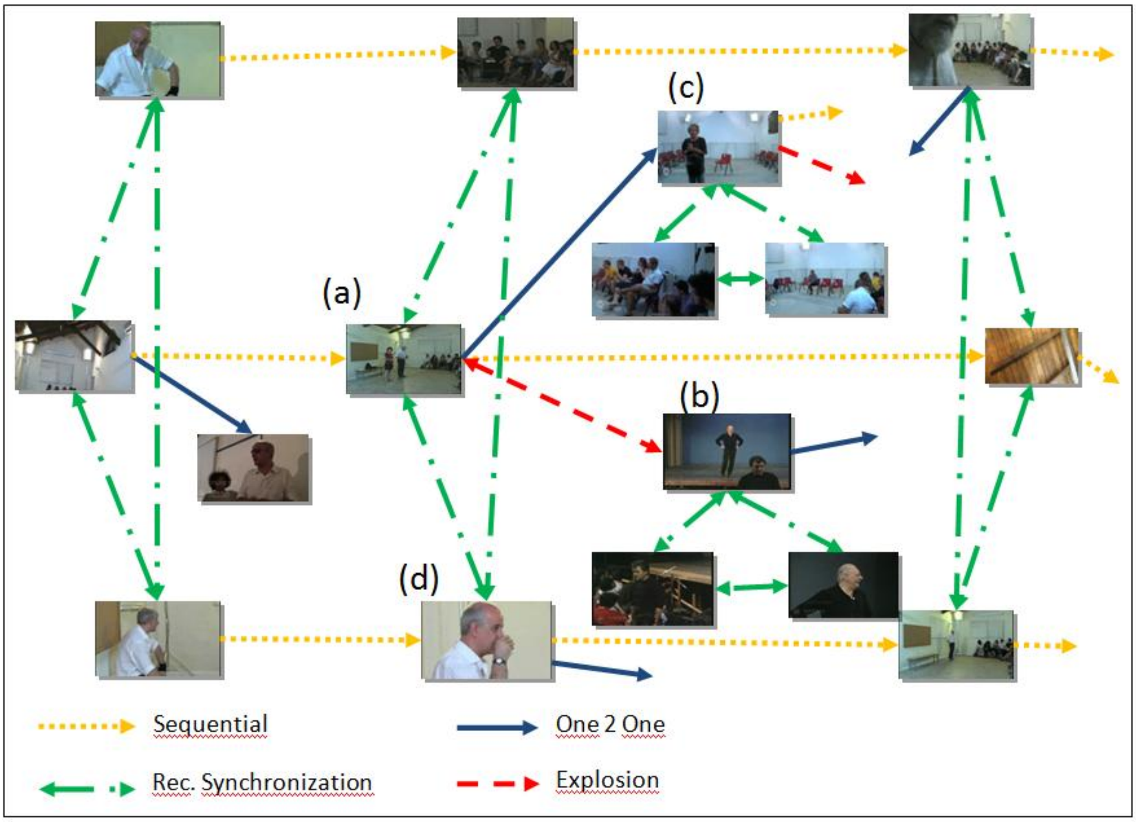


Figure 5.5: Example of relationships among several media executed by the MyStoryPlayer (a part).

The letters identify the scenarios corresponding to the snapshots commented in Figure 5.4.

The example of Figure 5.4 and 5.5 focuses only on a possible path that the user can follow in the presented relationship structure. As we can see from Figure 5.5, there are many possible paths that the user can follow in his experience on MyStoryPlayer. This approach is quite similar to browsing and navigation experience on web pages hyper-links. The more the relationships grow, the more the user is provided with paths to navigate on. *Moreover, the performed navigation can be saved and shared with other users as depicted in next subsection.*

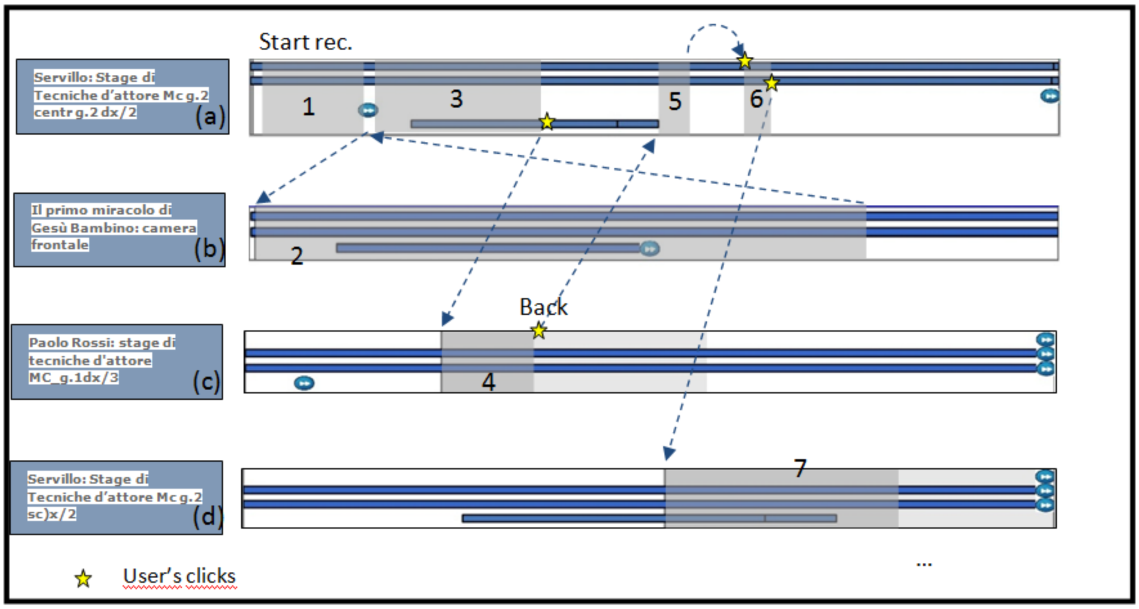


Figure 5.6: Example of user navigation among several media relationships.

5.4.1 Exploiting relationships to navigate on audiovisual media by using MyStoryPlayer

In this section, an example about the use of MyStoryPlayer to perform and save an experience in navigating among relationships shown in Figure 5.5 is provided. When the user starts executing the MyStoryPlayer from the media indicated by (a) in Figure 5.5 (e.g., (a) snapshot of Figure 5.4), the MyStoryPlayer loads the scenario with the main video synchronized with other two videos, with an explosive annotation, a direct annotation in a specific segment of the timeline, and a related video in sequence. Therefore, the user has the opportunity to abandon the execution of the master media by jumping on a different medium/context and following one of the proposed relationships, and thus the new media and related relationships are executed synchronously, as well. Thus, the user may switch from one video/media to another and return back in the stack of events, by using the BACK button. From that point, in Figure 5.6, a possible user navigation is reported as described in the following points, executing a set of segments:

1. from (a) context, the MyStoryPlayer reproduces synchronously the main media with the other two. The user started the recording at a time instant in the context of the first media of *Toni Servillo*. The user could choose to swap to another related media. In this case, the user keeps on viewing the media with original title "*Servillo: Stage di Tecniche d'attore Mc g.2 centr g.2 dx/2*";
2. at a specific time, according to the Explosive relationship the context passes at the related media (thus passing to (b)). In (b), three synchronized media (a master plus other two) (see (b) on Figure 5.4, *Il primo Miracolo di Gesù bambino*) are provided for the duration of the relationship. The duration is identified with the dark zone 2 in Figure 5.6, in which a One2One relationship is included. Once zone 2 is played, the MyStoryPlayer returns back automatically to context (a). The MyStoryPlayer and the Explosive relationship bring back the user to former scenario (a) to execute segment 3;
3. in segment 3 of Figure 5.6, the scenario has 2 Synchronizations and One2One relation. During the viewing of the media and context (a) has a master the original title "*Servillo: Stage di Tecniche d'attore Mc g.2 centr g.2 dx/2*". The user decides to view the video related with One2One relation by clicking on video (c) on the right side. This happened at the time instant marked with a yellow star on segment 3. The user changed the context by swap passing to context (c) and placing that video to master position;
4. the execution passes to play segment 4 in (c). The user could view the video for the entire duration, or can go back in the previous scenario. On the other hand, after a while, the user goes back, by click on the Back buttons. Thus the player brings him back to the end of the relationship in the former video, to execute segment 5;
5. segment 5 is executed for a while, then the user decides to perform a jump forward on the timeline of video (a), thus passing to segment 6;
6. During the execution of segment 6, the user performs a swap selecting video (c) on the right side of the user interface. The new context presents some synchronizations and annotations;
7. On segment 7, the user decides to play for a while. Then the experience recording is stopped by the user and the experience can be saved.

The experience saved can be formalized as follow using RDF-XML, recording only the actions performed by the user:

```
<msp:Experience rdf:about="http://www.eclap.eu/msp/experience/Exp1">
    <dc:title> Test for Paper</dc:title>
    <dc:description>Experience for Paper</dc:description>
    <dc:date>2013-10-28</dc:date>
    <msp:hasFirstStep rdf:resource=" http://www.eclap.eu/msp/experience/Step1"/>
</msp:Experience>
<msp:Begin rdf:about="http://.../Step1">
    <msp:isFirstStepOf rdf:resource="http://.../Exp1" />
    <msp:hasNextStep rdf:resource=" http://.../Step2"/>
    <msp:mediaUri rdf:resource="urn:axmedis:00000:obj:b44bad4b-817c-43ac-8553-8b83d5764a0b"/>
    <msp:clickTime>00:04:56</msp:clickTime>
</msp:Begin>
<msp:Swap rdf:about="http://.../Step2">
    <msp:isStepOf rdf:resource=" http://.../Exp1" />
    <msp:hasNextStep rdf:resource =" http://.../Step3"/>
    <msp:swapAnnotation rdf:resource="http://...#Relation_1333215764296_1493" />
    <msp:timeSwap>00:18:57</msp:timeSwap>
</msp:Swap>
<msp:Back rdf:about="http://.../Step3">
    <msp:isStepOf rdf:resource="http://.../Exp1" />
    <msp:hasNextStep rdf:resource ="http://.../Step4"/>
    <msp:backTime>00:20:12</msp:backTime>
    <msp:backTo rdf:resource="urn:axmedis:00000:obj:b44bad4b-817c-43ac-8553-8b83d5764a0b"/>
    <msp:backToTime>00:31:01</msp:backToTime>
</msp:Back>
<msp:Seek rdf:about="http://.../Step4">
    <msp:isStepOf rdf:resource="http://.../Exp1" />
    <msp:hasNextStep rdf:resource =" http://.../Step5"/>
```

```
    <msp:clickTime>00:34:14</msp:clickTime>
    <msp:seekToTime>00:36:27</msp:seekToTime>
</msp:Seek>
<msp:Swap rdf:about="http://.../Step5">
    <msp:isStepOf rdf:resource="http://.../Exp1" />
    <msp:hasNextStep rdf:resource =" http://.../Step6"/>
    <msp:swapAnnotation rdf:resource="http://...#Relation_453748659348_2367" />
    <msp:timeSwap>00:37:42</msp:timeSwap>
</msp:Swap >
<msp:End rdf:about="http://.../Step6">
    <msp:isStepOf rdf:resource="http://.../Exp1" />
    <msp:clickTime>00:49:06</msp:clickTime>
</msp:End>
```

5.4.2 An example of media relation model

Clicking on one of the videos (audio or images, as well) on the right provokes the change of context, thus bringing the clicked media in the first place (on left). From the point of view of the ontological model, the change of context means to contact MyStory Server with a query to get the relations of the selected media and to recreate the knowledge into MyStoryPlayer. The knowledge may grow around the present network of annotations and past annotations may be discharged after some jumps. It is self-evident that new annotations have been loaded, whereas the backlog of past annotations is still accessible. In order to guarantee a number of back jumps to rewind the experience, the stack of the performed activities has to be kept, storing also time instants where the context changes have been performed. To store the full experience on the player side could be quite expensive in terms of memory, while the server may store this activity for the user. On the other hand, to have all the story saved into the client computer can allow to reuse the information for navigation in local information and therefore for any replaying of the experience without connection costs.

The recent trends of linked open data are substantially in the direction of stimulating, not only the open formalization of descriptors, but also the interrelationship among different content descriptors in different portals. Descriptors may bring forth connections among audio visual content as well, thus creating the possibility of having linked open content with their relationships and aggregations. A strong emphasis has been given by new semantic models. For example, the EDM (Europeana Data Model) includes different kinds of relationship to define content collections, aggregations, events in time, etc. [EDM]. In some cases, these relationships among content are established among their metadata semantics, such as related content items: coming from the same archive, representing the same actor, representing the same historical period, etc.; thus formally creating collection, play lists, courses, etc. To this end, formally explicit semantic descriptors can be defined as in EDM and in ECLAP models [Bellini, Cenni, Nesi, 2011].

5.5 Production of Media Relationships

As explained in the previous section, MyStoryPlayer allows playing established relationships among media. These relationships can be produced manually, automatically and semi-automatically. The automated production is performed during the content ingestion process of ECLAP, and includes the possibility of defining aggregation relationships among media and content on the basis of tags provided in the metadata by the content providers [Bellini et al., 2011].

In this section, the manual and semiautomatic production processes to create relationships are presented. According to the relationship model as presented in Section 5.3 a tool has been developed and integrated into ECLAP for registered users. The first step to access the Add Relationship tool and define a new relationship is to select the "Add Relationship" from an audiovisual content in any ECLAP list of content, including content in results of queries, content featured, last posted, top rated, etc. The addition of relationships is part of the typical work teachers would like to perform with their presentation and content organization, thus fully exploiting the MyStoryPlayer model.

For users the first step to access the Add Relationship tool and define a new relationship is to select the "Add Relationship" from an audiovisual content in any ECLAP list of content, including content in results of queries, content featured, last posted, top rated, etc., see Figure 5.8.

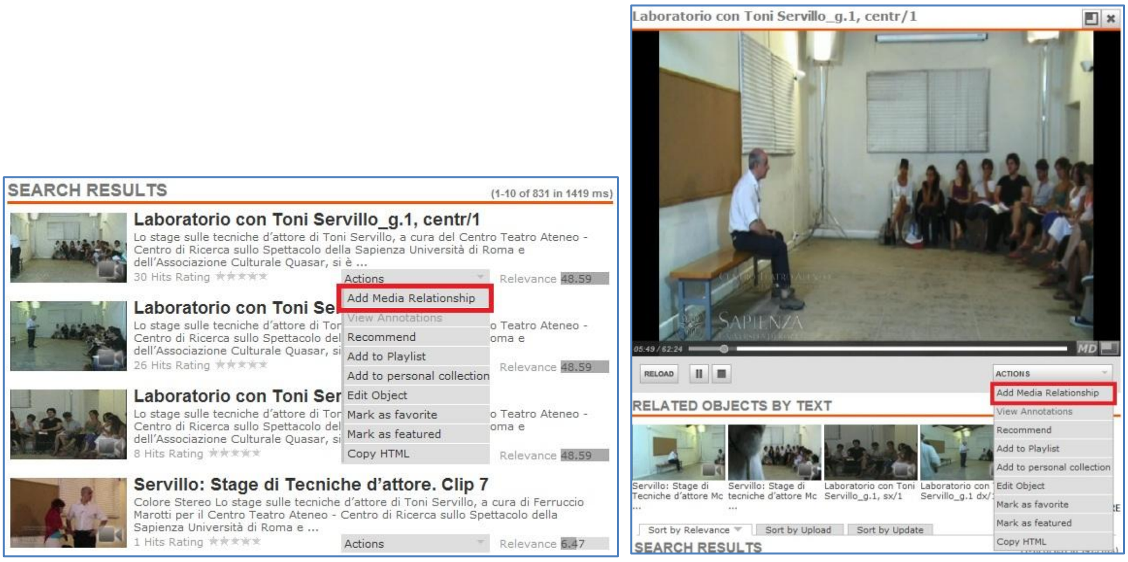
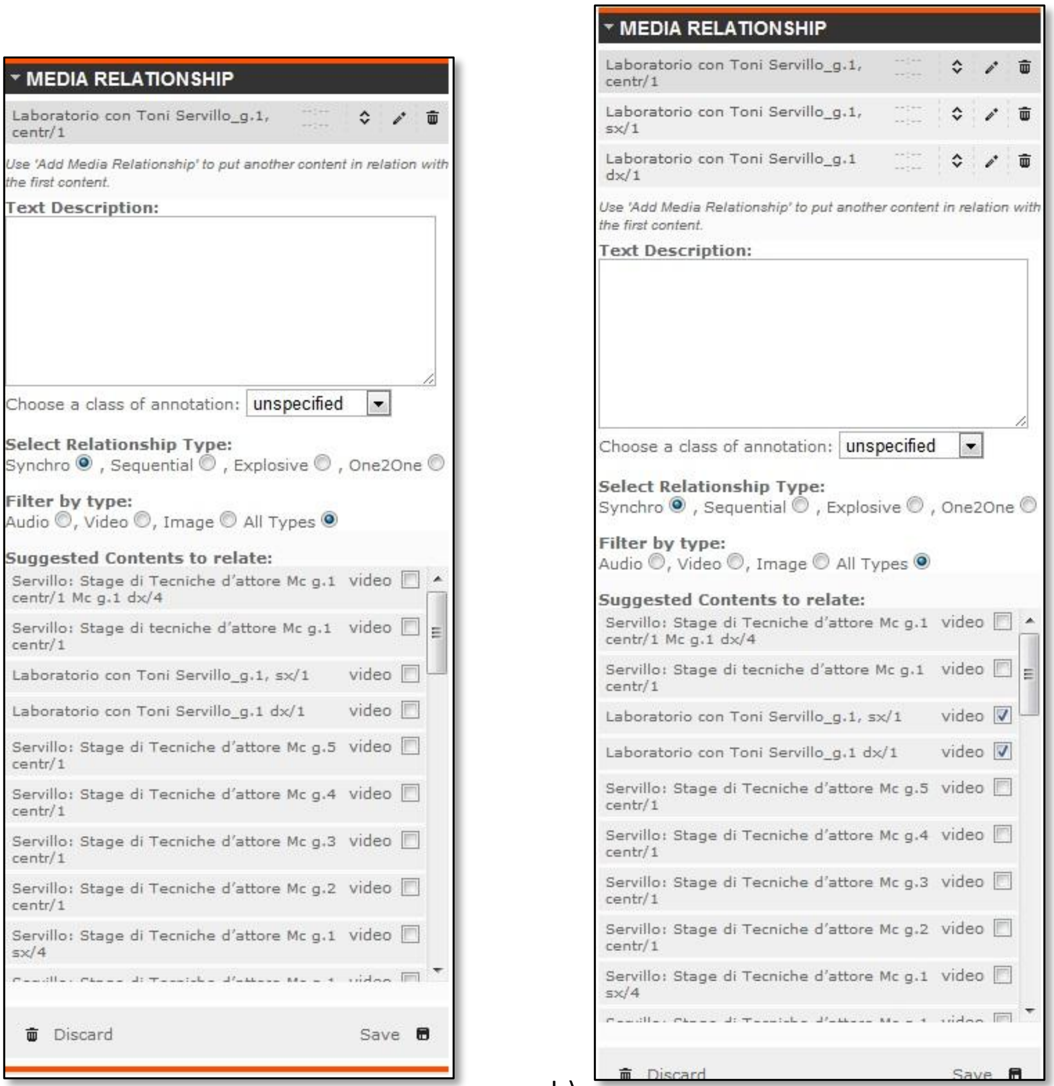


Figure 5.8: Add Media Relationships action on ECLAP

Therefore, once decided to add a relationship, the Media Relationship tool (depicted in Figure 5.9) is proposed, always referring to the content selected (in the same manner other content can be added, while the first is considered to be the master). The interface provides information to the user, and allows deciding which kind of relationship to be chosen: *One2One, Synchronous, Sequential, or Explosive*. The ECLAP portal has a large amount of content, users may use the search facilities of ECLAP to identify the content to be related and aggregated according to those models.



a) b)

Figure 5.9: Creation of assisted relationships among media. The user may select the kind of relationship to be exploited: One2One, Synchronous, Sequential, or Explosive. For example, when selecting 'synchronize', a list of suggested media to be related is proposed (by clicking on the check box related to items, the medium is added to the top list). Then saving the selection implies creating all reciprocal relationships. The list of suggested media is produced by a similarity algorithm based on metadata similarity. Different algorithms are used in different cases, and the user may filter the results for different media: only video, audio, images, etc.

The Add Relationship tool provides a different behavior according to the type of relationship as described in the following.

One2One: in this case, the involved media can be two (among video, audio and image), and the time segments should be chosen directly on a proper player, independently. The ECLAP audiovisual player allows defining the start and end positions on their time line for audio and video, while for images the duration is requested (see Figure 5.10).

Figure 5.10: Editing time segment along the temporal line of a video by positioning the cursor in the selected time instant and clicking on the corresponding marker points for start [- and for end -].

Explosion: In MyStoryPlayer domain, a different kind of executable relationship among audiovisual is defined as the Audio Visual Explosion. This case can be regarded as a special case of the previous scenarios where the master audiovisual segment had a zero duration, that means to have an audiovisual annotation attached to a single time instant.

In this case, the involved audiovisual can be only two, the user has to identify only a single point on the master audiovisual, and a segment in the second. The ECLAP audio visual player allows defining those points with simple clicks on the time line.

The execution of this audiovisual relationship among Video 1 (V1) and Video 2 (V2) allows expanding the execution time line of V1 with the identified segment of V2; just returning to the execution of V1 once the V2 segment execution is completed. This model is equivalent to the action of opening a parenthesis where some aspects can be recalled, and then closing it to restart from the point where the parenthesis had been put. This kind of relationship can be used to explain a single time instant with an expanded scenario; to show possible cut scenes, to stress possible variants, to insert comments by the director, to explode a single time instant with a more complex scenario , etc.

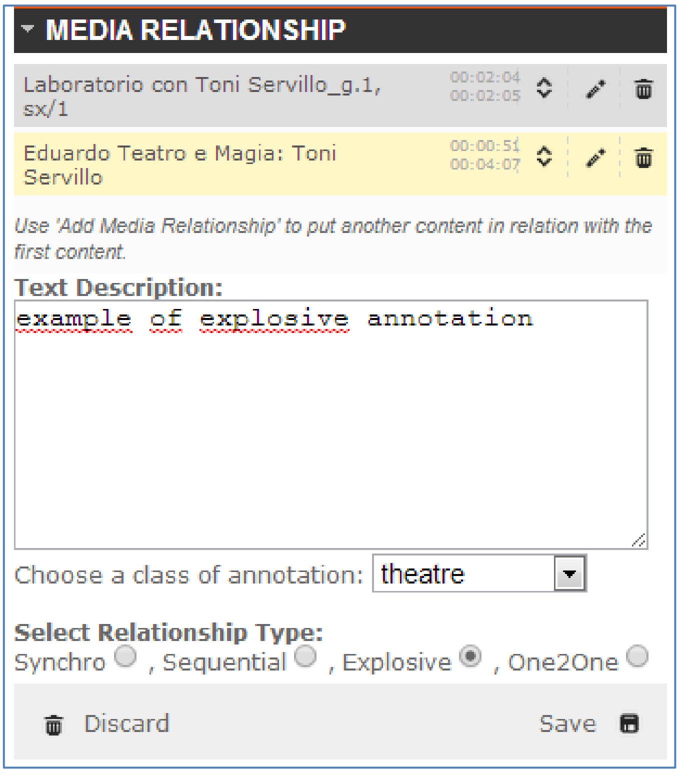


Figure 5.11: Example of authoring of explosive relationship

Synchronous: in this case, the number of media can be very high since for N media N^2-N relationships are produced. They can be selected from those suggested or from the other ECLAP content lists. The pool of selected content is automatically related one another and reciprocally, thus avoiding to replicate the same relationships with the same time segments and text. Also in this case it is possible to set media segments to relate them, instead of the entire media, starting from the beginning of each media source. As in the One2One relation, it is possible to set the start and the end position out of the player interface and create reciprocal annotations just for those selected parts. If no segment is chosen, the entire medium is taken by default. In Figure 5.4 (b), the three chosen media identify the first part of the first day, taken from three different points of view. When creating these relationships, the system will create 6 reciprocal relationships relating each item to the others, which means that all the media can be regarded as masters.

Sequential: in this case, the number of media can be very high. This kind of annotations is made to satisfy the need to put in sequence media concerning the same event, or to give a sequential order to media according to a timeline or a narrative path. They can be selected from those suggested or from the other ECLAP content lists. A system of suggestions has been developed, which helps users in finding similar objects with similar titles and having the sequence of the event already encoded (for example, title_1/3, title_2/3..). When the relationship is created, the pool of selected content is automatically related one

another according to the order such content is presented with in the Add Relationship tool. The user may move up and down the selected content by using drag and drop mechanisms. Also in this case, the system avoids replicating the same relationship among the same media with the same time segments and text. The default value sees the media aligned since the end of each item, when automatically the player switches to the following media.

By clicking on Save button, the relationship(s) are added to a content; thus, the number of relationships for that content is increased and displayed in the content lists on ECLAP via small icons such as 3, which means that 3 relationships are available for that content.

5.6 Delete a Relationship

It is possible to delete a personal annotation, simply by going on the menu *Content->MyAnnotations*.

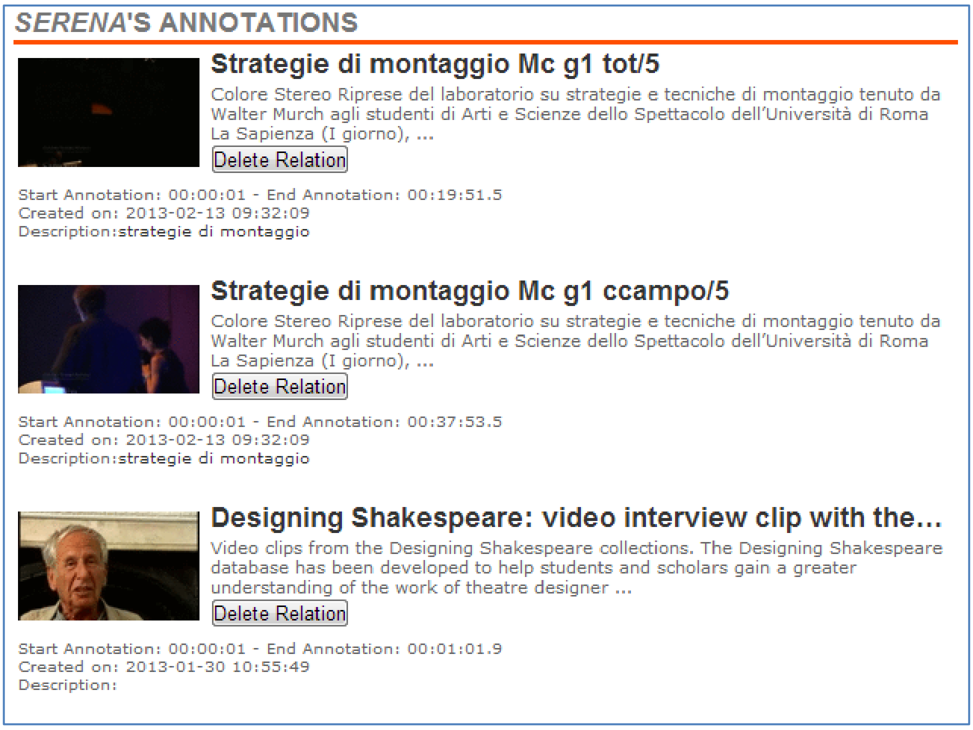


Figure 5.12: menu my Annotation. It is possible to delete a relation.

A list of personal annotations will be displayed, and clicking on the button *Delete Relation* it is possible to cancel the annotations one by one.

6. MyStoryPlayer Architecture

According to the above-presented scenarios, the user starts accessing the MyStoryPlayer on ECLAP portal for example: (i) by clicking on a small icon highlighting the presence of annotations/relationships associated to a given media, (ii) deciding to add a new relationship, (iii) performing a query on the annotations. Once a relationship is selected, the chosen master medium is put in execution with its related digital media essence, thus the MyStoryPlayer automatically displays a set of relationships and time line. Relationships are executed aside the main essence according to the time line and to the semantic related to them (One2One, Sequential, Explosive, Synchronization).

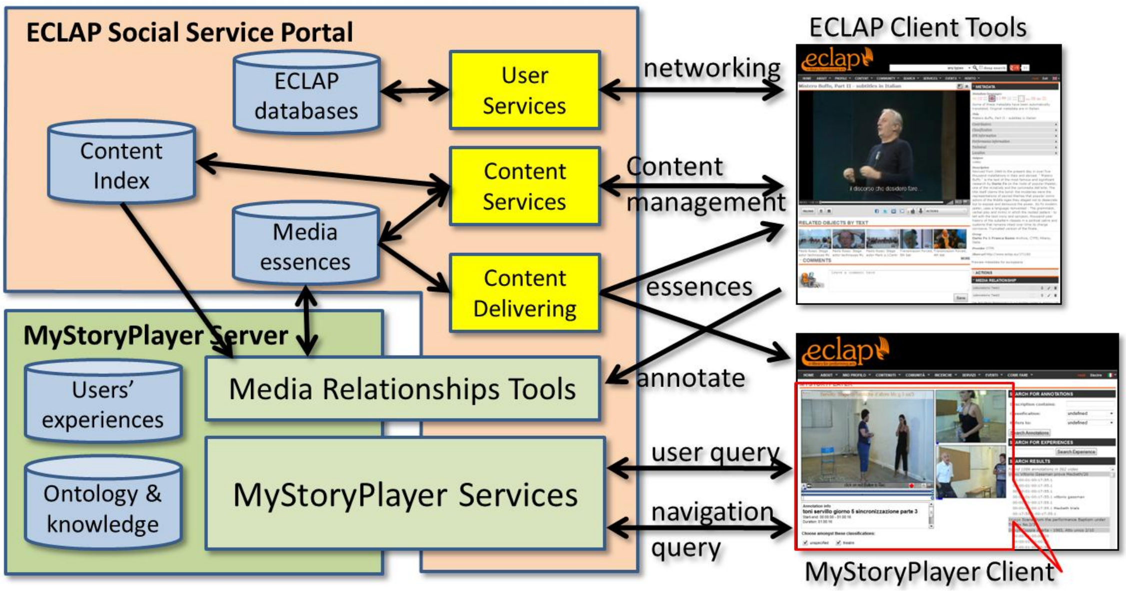


Figure 6.1 – General Architecture of MyStoryPlayer and its integration with ECLAP Service Portal.

In Figure 6.1, the general architecture of MyStoryPlayer is provided in relationship to the ECLAP service portal [ECLAP]. The MyStoryPlayer Server can be integrated with complex social networks such as ECLAP Social Service Portal (http://ww.eclap.eu). In this case, the ECLAP users adopt an integrated module (Media Relationship Tools, described in Chapter 5) to establish relationships and realize annotations among media, exploiting the content indexing for producing suggestions for creating relationships. The established relationships are stored into an RDF database according to the ontology presented in Chapter 5, thus creating the ontological knowledge database managed by SESAME [Sesame], [Broekstra et al., 2002]. The relationships refer to the media essences ingested and managed by the social network via content delivering service (based on *xmoov* streaming server to enable the seeking forward on server side).

The architecture of MyStoryPlayer is composed by MyStory Server, ECLAP Social Service Portal, the authoring tool and the player.

MyStory Server is composed by the ontology and knowledge base stored in a Sesame repository and by the users' experiences. MyStoryPlayer Services is accessible from the player during the navigation, when there is the necessity to load information about new contexts, and from the users when make queries on the portal in order to take new media as entry points in their experiences.

When the user opens content into the MyStoryPlayer page may perform a semantic query to the semantic database. As a result of the semantic query in SPARQL [SPARQL], a list of entry point media and their related scenario of relations are provided. An Entry Point scenario may be a video, a starting point and from that a set of relations and annotations connected to the video that has to be placed in execution at the same time or prepared.

Moreover, the MyStoryPlayer client tool enforced into the web page is activated on the basis of a media, and thus it creates a direct connection to the MyStoryPlayer Services to progressively download the set of RDF relationships associated to the master media every time the master media changes. For example, when the MyStoryPlayer client is forced to change context according to the user interactions. In this case, the change of context also implies to restart progressive download of media streams.

6.1 MyStoryPlayer Client

The MyStoryPlayer Client tools has been designed and developed to execute the model presented in Chapter 5. It has been developed in Adobe ActionScript, so that the player in Flash is automatically provided and loaded during the web page loading as a flash player tool. It could be realized in: SMIL providing an ActiveX including the Ambulant interpreter as in AXMEDIS [Bellini, Bruno, Nesi, 2011], extending the JavaScript solution as in [Gaggi and Danese, 2011], and probably in HTML5 and JavaScript but with some complexity in managing the synchronization as discussed in the sequel. Thus, in any cases the problems to be solved remain unchanged:

- The progressive download and interpretation of the media relationships in RDF. For progressive download we intend the limited download of the information and relationships needed to manage the activated context, and perform the successive download every time the context is changed. It is possible to load in advance the next possible RDF descriptors without waiting for the change of context. This can accelerate the change of context and perform a partial semantic reasoning on the client side;
- The precise management of media synchronizations as described in the assessment and validation section of this paper. The problems of media synchronization are provoked by the start of the MyStoryPlayer Client, and at each change of context: jump, swap, back, etc., and also when One2One relationships are activated in the time line. On the other hand, they are more relevant in the case of low bandwidth. The knowledge of the relationships structure may be exploited to implement preload strategies for the involved media;

- the management of user driven context changes such as jump, swap, back, etc.; the new context has to be loaded and interpreted to be put in execution on the tool. In this case, the change of context also implies to restart progressive download of media streams.
- the management of the user interactions and controls on the user interface: volume on multiple media, play/pause/stop o the master that have to control also the other media, independent on/off of audio for each media involved in the synchronous rendering (thus allowing the listening to the talk of a teacher when slides are shown, hearing and watching director interview and comments together with the scene, watching multiple scene views of the same event, etc.), and other minor controls;
- the recording of the users' actions, thus managing the modeling, save and load of user experiences with the MyStoryPlayer Services;
- the possibility of rendering master media from different resolutions according to the available bandwidth, also taking into account the number of simultaneous video to be played and their bitrates;
- and the rendering of textual comments and other information, the management of labels, etc. Textual descriptors to the single annotation and resource can be used to highlight specific aspects and to start discussions.

For implementing some of these major features with the above mentioned technical tools (SMIL in JavaScript, SMIL/Ambulant, HTML5 and JavaScript, and Flash/ActionScript) some limitations have been detected in terms of interactivity and fast response. For these reasons, the implementation of the MyStoryPlayer client has been realized by using Flash/ActionScript development tool kit where the needed flexibility and control has been verified.

The MyStoryPlayer Client tool has been designed by using object-oriented technology and satisfy the above reported requirements. Its design is reported in Figure 6.2, in which one can see:

6.1.1 The Loader

The Loader makes queries, downloads and interprets the RDF triples of the involved relationship with the master media. This class is created and activated every time there is a need to access the repository Sesame, where the ontology is saved. So that, at each load of video, swap, back actions corresponds a query to the repository in order to take all information (URL, type of relation, type of media related, time codes of relationships, title of media, text description) of the media is playing and the others related to it.

It explores the triples of the RDF/OWL ontology, taking the subject of the element to explore and through the function getPropertiesOf(), querying and collecting every single information about that subject.

6.1.2 Experience Manager

Experience Manager collects the user experiences activities, models them as RDF, send them to server, download and play them according to the user requests. Talking simply, an experience is a mixture of clicks done by users during the navigation on the player.

These kind of clicks can be: Swap, Back and Seek. The class is managed by the controller that creates it once the user decides to record his path on the player. Every time the user makes an action, this class saves the event, collecting all information needed in order to replay it and creates the RDF code suited to the model in order to save it, keeping trace of the user, the date, text description and title of the experience.

In Section 5.4.1 there is a example of an experience.

6.1.3 Relation

Relation class is created by Controller, in order to manage the relations of a media.
The class Relation creates rectangles on the player and communicates with the Loader obtaining information about each relation in order to display them in the description box under the rectangles.

The main features of this class are focused on creating instances of Media References associated to each relation. For each relation associated to a media, a Relation Class Object is created.

6.1.4 Media Reference

The Media Reference class is instantiated every time the Relation class receives information from the Loader. Video, Audio or Image class is created. Different behaviors are associated with each class. Both Video Reference and Audio Reference have a duration related to the media, that is provided by the metadata of the object. For images as well duration is associated in order to reproduce it as a video, associating relationships with a start and end. The duration of an image is 20 second by default, but users can change it in the production phase of relationships on ECLAP.

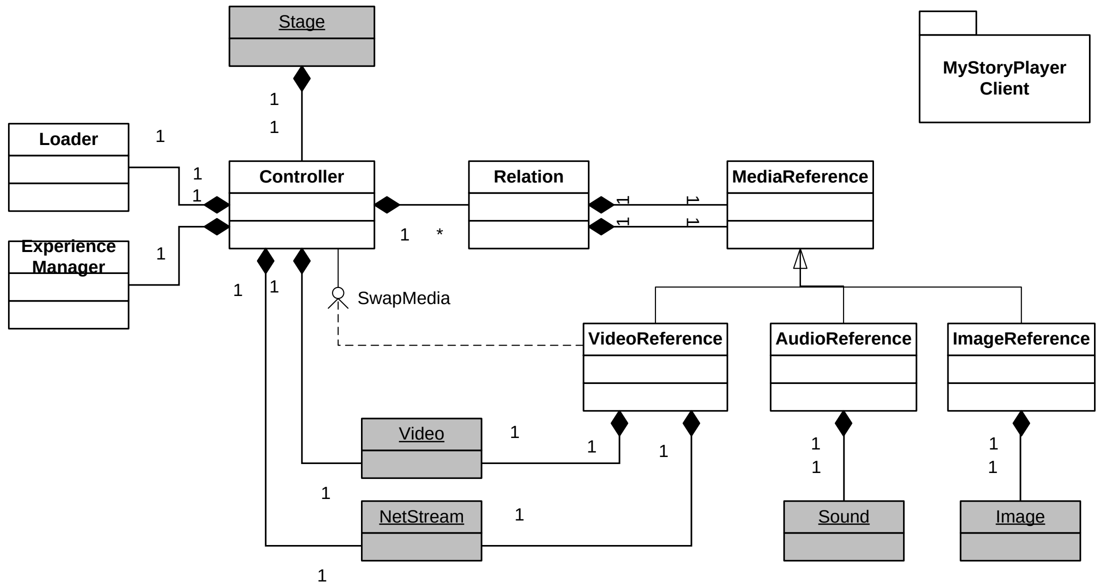


Figure 6.2 – The main classes of MyStoryPlayer Client tool and their relationships.

The classes reported in grey are those provided by the ActionScript library and framework.

6.2 Sequence diagram

To better understand the importance of synchronization for MyStoryPlayer usage, let us take a look on a diagram of sequence that shows how the classes are created and instantiated, taking in consideration the function *onLoop()* and the creation of Net Stream objects.

It is important to notice that when there are many video references, for each media a Net Stream object is created and managed independently from the others, in order to display on the player at the same time many synchronized videos when they are active. Each instance of Media References acts a connection with the Server, and if the type is a video, the Controller creates a Net Connection object and a Net Stream object in order to associate them to video Reference.

The Net Connection class creates a two-way connection between a client and a server. The client can be a Flash Player or AIR application. The server can be a web server, Flash Media Server, an application server running Flash Remoting, or the Adobe Stratus service.

NetConnection.connect() is used to establish the connection, the *NetStream* class is used to send streams of media and data over the connection. In order to manage all the fluxes, and due to the nature of ActionScript as event-driven language, it has been decided to start *onLoop()* only once all the instances have been initialized.

In order to manage all fluxes, some controls have been created, stating that until all the classes have not been initialized, the *onLoop()* function would not been activated. Because of the nature of Event-driven language as Actionscript, it has been necessary to put these statements, to prevent the lack of synchronization of fluxes or worst the malfunctioning of the application.

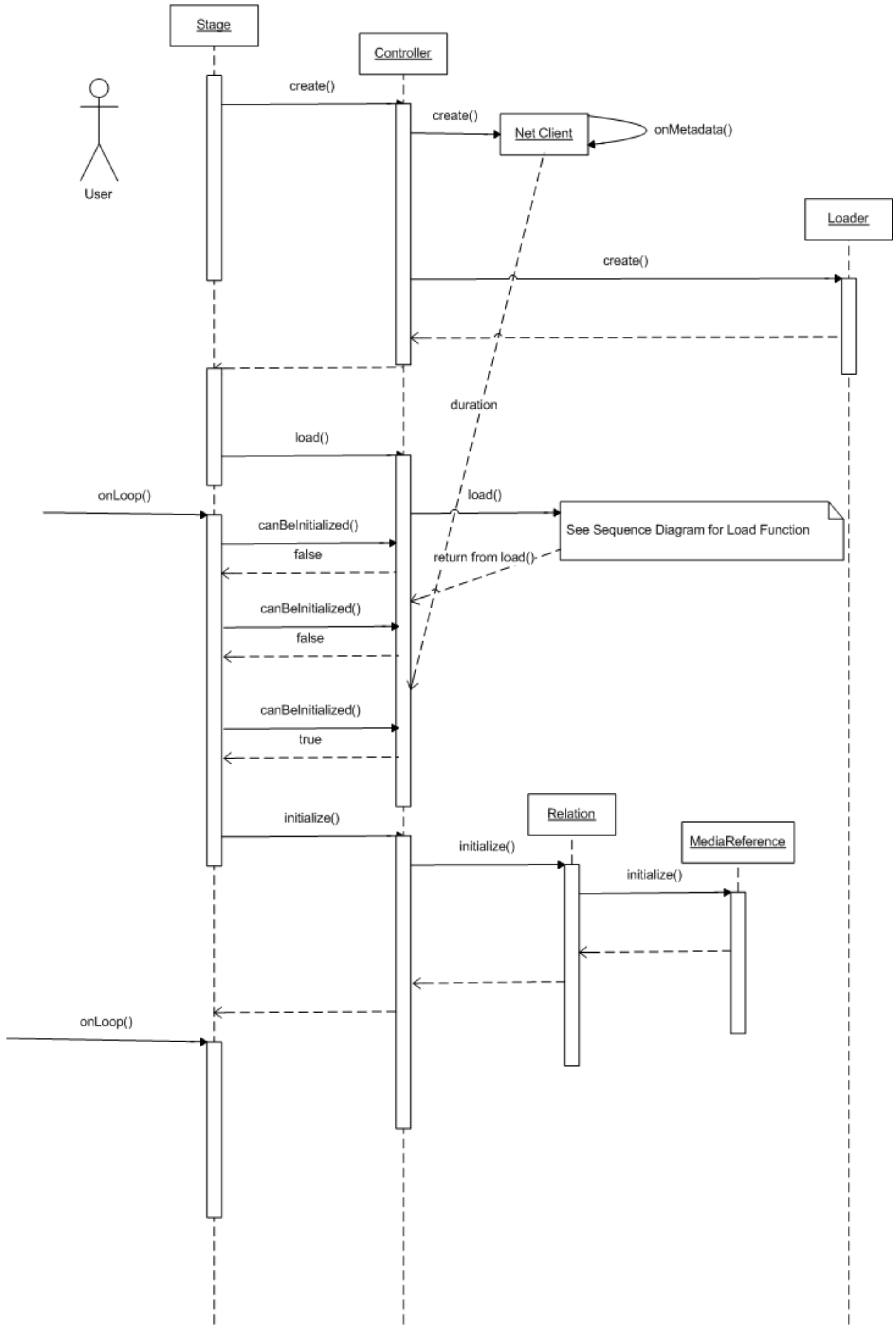


Figure 6.3: Sequence diagram of initialization player conditioned by data loading

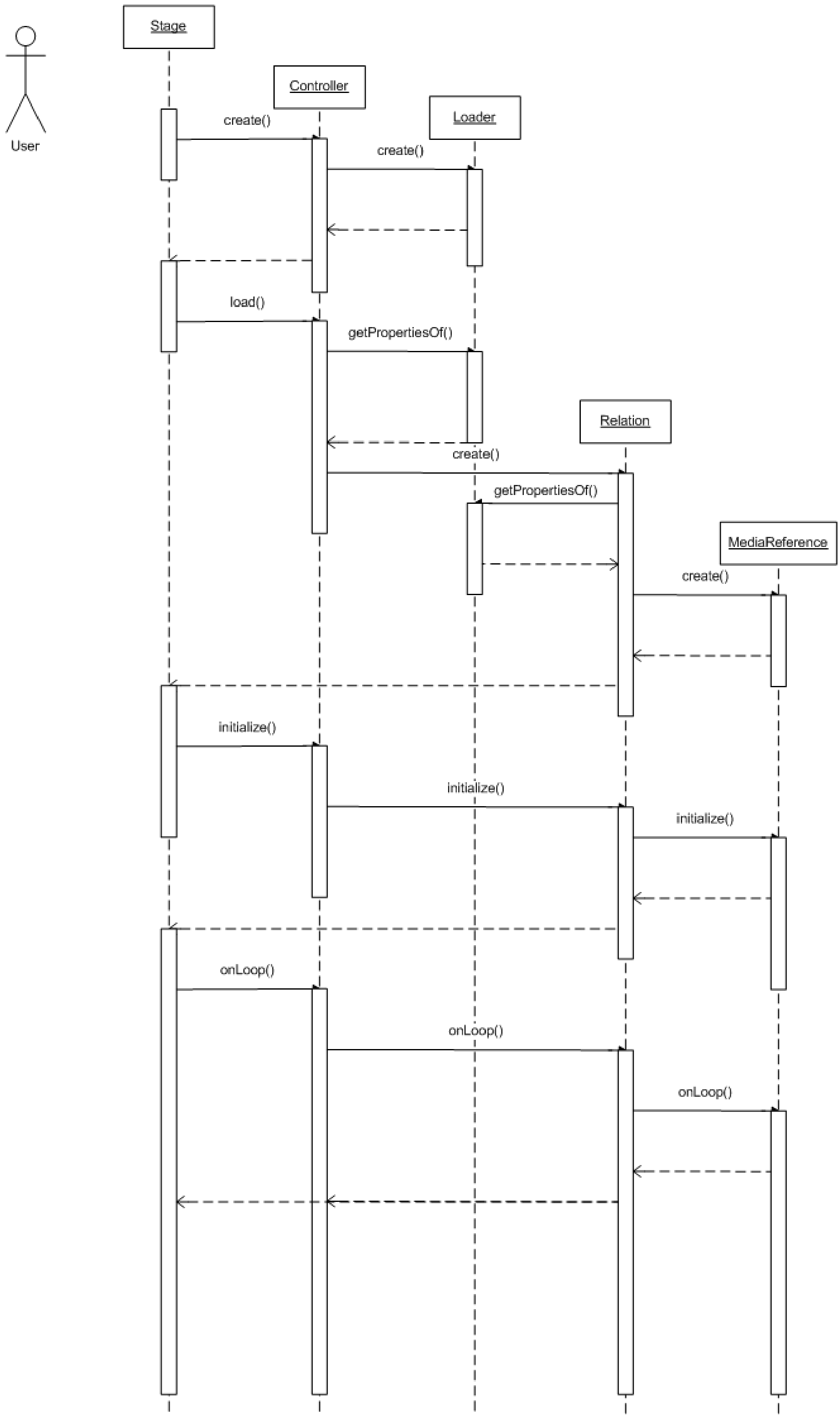


Figure 6.4: Sequence Diagram

In order to manage all the streams it has been created some controls inside of classes stating that until all the classes have not been initialized, the function *onLoop()* has not been activated.

In figure 6.4 it has been shown the sequence of player actions once activated.

First of all, the Stage creates an object of class Controller, passing as parameter the address of repository where to take RDF metadata, and a video object to which *NetStream* will be attached.

Controller creates a Loader class object in order to allow the execution of query to repository RDF taking information needed for the application.

Once created the two classes, *load()* function is called from an External Interface, loading the contents to be played. The control passes to the Controller, that calls Loader through the function *getPropertiesOf()* in order to take relationships related to the chosen media. During the response with the information of relationship, a vector of annotation is filled up, in order to store them for successive use and creating also Relation objects. For each relation one object is created. Each Relation object calls the Loader requesting information about the relationships, as the start, the end, the type, etc. Also the Media Reference is created depending on the type of media related (video, audio, image).

Once all these operation are completed, the control passes to the stages that can initialize all the instances, starting from the Controller.

The Controller initialization consists on initializing the rectangle of relationship, the slider of volume, but especially to call the initialization functions of Relation object.

Then, the control passes to Relation class, that initialize the rectangles of annotations and calls the function of initialization of MediaReference linked to the media relation.

MediaReference assigns a Net Stream object to the media, creates the icons of info, swap, volume slider and set the position of the media reference depending on how many are active during the reproduction.

Later, the control backs to the Stage, that call the function *onLoop()*. This function is called each 200 ms, 5 times per second, and pass from Controller trough Relation and MediaReference. It takes care the management of data flux and the successive calls to the class functions depending on the user actions or the nature of relationships

This is the algorithm that explain how the *onLoop()* function works a regime, once that all classes have been instantiated.

Algorithm

```
for all annotations related to MainVideo{
do
if(typeRelationship==explosive)
do {
        save current state,
        save the times of begin and end of media reference
        load new contents
return
}
else
        relation.onLoop(currentTime)
}
function relation.onLoop(currentTime)
        if (relation is active at currentTime){
                mediaRef.onLoop(currentTime-startRelationTime);
        active=true;
        }
        else{
                hideContent()
                active=false;
        }
```

6.2.1 Sequence Diagram for an explosive annotation

The starting point is function *onLoop()* that analyzes the annotation type activating different actions according to its semantics. When an Explosive Relation is identified, the context is saved through the function *saveCurrentState()* which saves the Net Stream information, the URL and the return time (end of relationship) of the video to be swapped, so as to be ready to return back after the end of the "Explosion". At this point the player has to load new information about the relation, such as the URL, the beginning and end time of the reference, and all the annotations related to the media referred by the Explosive Relation, via the mediaReferences. The function coping with this is *getPropertiesOf()* that interacts with the ParserRDF class to take all the information. Once this information is available, the player provides the user with a new context where in the main area there are the main annotation media, and aside the annotations related to them. The reproduction starts at the reference beginning and stops at the reference end; after that, the context of player is moved back to the saved execution point, through the back function.

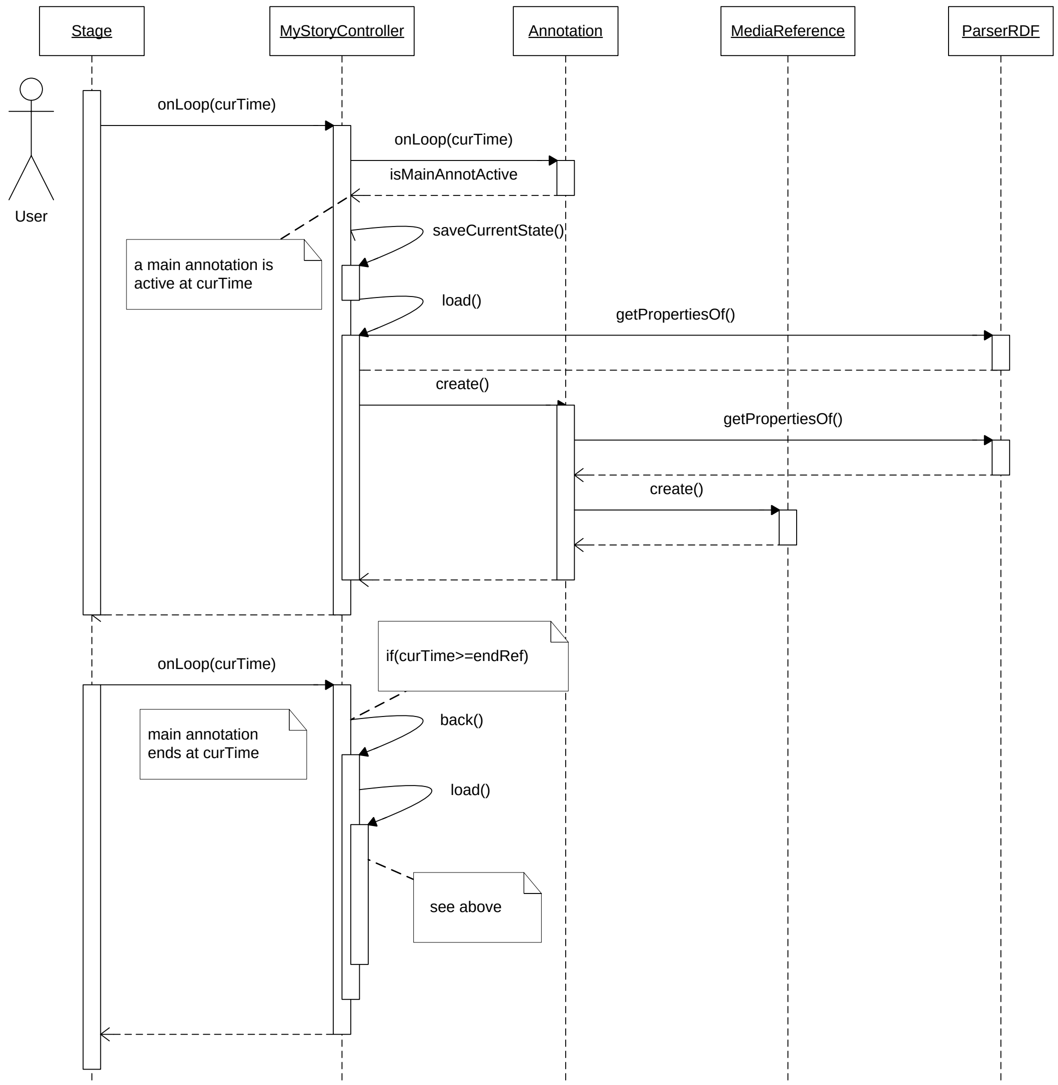


Figure 6.5 – The sequence diagram describing the internal actions in the MyStoryPlayer.

6.3 Access to information on Sesame Repository

RDF is a directed, labeled graph data format for representing information in the Web. SPARQL [SPARQL] can be used to express queries across diverse data sources, whether the data is stored natively as RDF or viewed as RDF via middleware. SPARQL contains capabilities for querying required and optional graph patterns along with their conjunctions and disjunctions. SPARQL also supports extensible value testing and constraining queries by source RDF graph. The results of SPARQL queries can be results sets or RDF graphs. For this reason SPARQL fits with the purpose of searching in the MyStoryPlayer semantic model the relationships amongst media essences. It has been created a suited page on the portal where the users can access in order to searching the scenes in the graph of relations on performing arts content in ECLAP.

According to the semantic model adopted, the user may perform semantic queries in SPARQL. For example it is possible to search some label of classification or text description previously saved in the ontology , and users may perform the following SPARQL queries to get all scenes in which the description contains the word Dario Fo and is classified as "Theatre" scene..

For example, a semantic query in SPARQL to get “All the scenes where in the description is present the string "Dario Fo" and it is classified as "Theatre" and refers to a video essence”, is reported below:

SELECT DISTINCT ?Video ?Annotation WHERE {

?Video a msp:Video. ?Video msp:hasAnnotation ?Annotation.

?Annotation msp:hasClassification ?classification. ?classification rdfs:label "Theatre".

?Annotation msp:hasTextDescription ?description. ?description xml:string "Dario Fo".

?Annotation msp:hasMediaReference ?mediaReference. ?mediaReference msp:refsTo axoid.

?axoid rdf:type "Video"

}

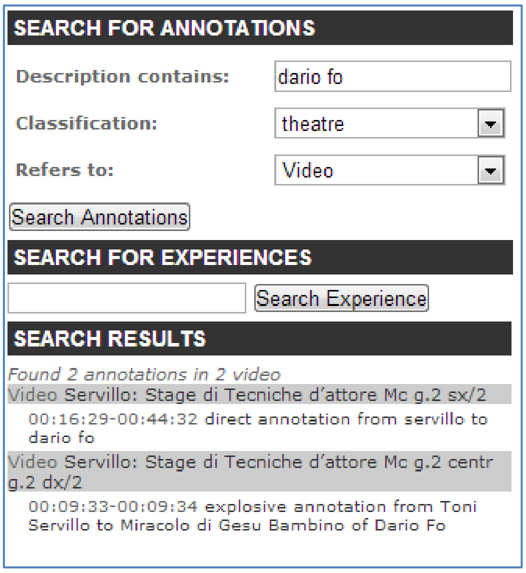


Figure 6.6: results of query on the front page of MyStoryPlayer

As we can see from the figure 6.6, the user compiles the search mask and clicking on Search annotations, a list of corresponding results will be shown.

The provided results are composed by the title of the media and the annotations related to it. clicking on the title, the media will be open from its begin, otherwise, clicking on the annotation the player will start from the begin of annotation.

7. Inside Controller and Synchronizations

In order to better explain the problems to be solved for the implementation of MyStoryPlayer Client, the activities of class Controller has to be better analyzed.
The class Controller manages:

- the initialization of the several instantiated media player and context by activating the Loader of RDF relationships,
- the creation/allocation of media Relations that creates specific areas on the user interface of the MyStoryPlayer and communicates with the Loader to obtain information about each annotations (associated to relations) in order to display them in the description box under the main media player area,
- the activation of Experience Manager.

Once the reproduction starts, the singleton of class Controller manages *onLoop()* function by which the event management is performed. Among the events to be managed the Controller has: the user interactions on video controls (play, pause, seek, stop, volume control, etc.), the context change such as clicks to jump, swap and back, and the events coming from the network streams management, as buffer empty, buffer full, for each media. Exploiting these capabilities it is possible to manage multiple streams, relations and synchronizations among them through activities of preloading, buffering and seeking of video according to suitable seeking algorithms that may depend on the events generated by the Net Stream.
The main aim of MyStoryPlayer Client is providing a good quality of experience to users in order to guarantee a suitable synchronization among media conformant with the expectations of the user who has established the relationships. To this end, it is important to minimize desynchronizations and reestablishing synchronizations among media when:

1. They are synchronized via One2One and Reciprocal Synchronization Relationships. This can lead to code the contemporaneous start of a set of videos or the start of additional video streams at a given time instant, while other are playing.
2. The master media execution is requested to perform a jump, for example when the user clicks on the timeline of the master video. The jump can be backward or forward, inside or outside the already buffered video stream. In presence of synchronized media, all of the synchronized media have to jump as well reestablishing the synchronizations in short time.
3. One of more video streams finish their cumulated buffer, and thus they do not have frames to be played. If this happen to the master, all the other video streams should wait for the new buffering of the master to restart. When this happens to other video streams, large cumulative desynchronizations may be generated.
4. The context changes for the execution of Explosive relationships the other media are stopped to live time at the exploded media segment to be played, and then return back to the master in the same time instant. This means that the other media can continue to the buffered while are stopped.
5. The context changes for a swap/back of media for the user intention (clicking on a video stream on the right side of the MyStoryPlayer). In presence of new context, a set of synchronized media may be present, and all the streams have to restart and rapidly reach the synchronization again. The new condition can be anticipated by pre-buffering all the media of all possible scenarios that can be reached by a swap, and saving the stack of buffered video. On the other hand, this solution is very expensive in terms of bandwidth, and thus cannot be applied.

The state of the solutions are typically based on creating a muxed stream compounding multiple video streams to prepare precooked synchronizations to be streamed to the clients. This approach is viable for broadcasting applications (for example in MPEG-2 TS), and the client player can be quite simple, while the coded synchronizations and relationships can be difficulty changed in real time. A surrogate of changing stream can be implemented by turning off a stream and play another one

from the several streams already available on the muxed multiple streams, similarly to what happen to camera selection of F1 Gran Premium on satellite broadcasting.
In order to have a more dynamic solution, the media streams have to reach the client player as independent streams. Multiple video streams could be provided by using RTSP server to some custom player to be kept synchronized according to the RTSP protocol which constraints to establish specific client-server connection for each stream. On the other hand, today the most diffuse solutions for web-based video players are typically based on simpler and less expensive HTTP progressive download (see for example Flash players, and HTTP 5). In [Gao et al., 2011], the multiple video streams are received by a custom player to be rendered. In this case, in order to keep video synchronized each other the custom player implemented strategies skipping frames and jumping / seeking on the local stream. This approach constrained the solution to receive the full stream of non compressed video, to be capable of selecting frames. This last solution constrained to install a custom video player and does not optimize the exploitation of network, since the player needs in any case to get the full rate video stream. Thus is not suitable for low bandwidth conditions.

MyStoryPlayer has been implemented in different versions during his development phase, and thanks to many tests and consequent improvements, has become more accurate and precise especially from the synchronization point of view.
The early implementation of the MyStoryPlayer Client (called in the following Solution 0) was based on starting the HTTP video streams when needed as [Bellini, Nesi, Serena, 2011]. This approach leads to cumulated errors of desynchronizations especially in presence of low bandwidth. In order to reduce these problems, the size of stream buffer to be cumulated before video start can be increased. On the other hand, in the cases of long videos to be kept synchronized, such as in lessons, the probability of cumulating relevant delays is very high, thus resulting in increasing delays and variance of the desynchronizations among the streams, especially in presence of low bandwidth (see Section 7.1).
In order to guarantee a sense of continuity between clips, it is important that there is minimal presentation delay when swapping between them or minimum delay among video that are synchronized and reproduced together, so that several new

MyStoryPlayer Client solutions have been created and tested including an adaptive algorithms that combine a set of techniques:

- Performing a client side seeking within the stream buffer already loaded by the client (thus acting a sort of frame skip at on the client side). Flash Media Server and Flash Player work together to support smart seeking in VOD streams and in live streams that have a buffer. Smart seeking uses back and forward buffers to seek without requesting data from the server. Smart seeking reduces server load and improves seeking performance. In MyStoryPlayer Solution 1, this seek is performed every time the difference between media reference time and main media time is higher than a threshold and the point where to seek is inside the buffer already loaded for the media reference. The chosen value of the buffer is greater than 1 in order to make concrete the jump ahead or back, keeping more continue the perception of streaming to the users. The trade-off from an high value that reduce the number of seeking but augmenting the waiting for the streams has been set to 5 seconds.

  This kind of policy leads to good performances in case of high bandwidth because the delay between media and media references generally remains low, leading to a limited number of seek.
- requesting a seeking of the video stream to the MyStoryPlayer Server (*SeekServerTime*) on server side, thus reducing the network workload for sending frames that are not necessary.

  This seeking is performed when the difference between media are higher than buffer time. The service *xmoov* is used with pseudo streaming randomly accessing to parts of a video that has not been downloaded yet.
- adaptively adjusting delay in actuating the seeking on the client side (*DelayToSeekClient*).
- adaptively adjusting delay in obtaining a frame seeking from the server side and see it actuated on the client (*DelayToSeekServer*), this approach avoids the workload of the network with frames that are not played.

  This parameter induces a jump ahead respect the main media timeline in order to allow, especially for low bandwidth, a rapid convergence to the main media time.

- adaptively adjusting delay in starting a video at a certain time instant, for example in the case of One2One relation (*DelayToStart*).

The mentioned *DelayToSeekClient*, *DelayToSeekServer*, and *DelayToStart* have to be continuously estimated to create an adaptive adjustment solution. This approach is needed since the available network bandwidth is not stable and may change from stream to stream.

This approach allows to align in a rapid way many synchronized media giving a new solution with the ahead jump on the timeline, taking in consideration the delay of SeekServer.

The typical estimation model is based on the running average of the last 3 estimated values, more sophisticated algorithms are difficult to be implemented since the *onloop()* function has to run all actions in 200ms for all videos. The *DelayToStart* can be taken into account to anticipate the video buffering when One2One relations are approaching. On the other hand, in presence of low bandwidth it can create destructive effects to the current rendering. The HTTP protocol and ActionScript have limited capabilities in controlling the video buffer on client side when compressed video code is used, as in this case. Moreover, the seeking has to be avoided if the desynchronizations is lower than the acceptable threshold, thus avoiding un-useful oscillating jumps in the video play. This limit is related to the resolution of the *onLoop()* function of the Control singleton in the player. In the case of Flash player, we have a cycle every 200 ms, that could be an acceptable desynchronizations error in the cases of most video lessons.

A simplified algorithm is reported in the following pseudo code:

Algorithm

```
Error=abs(MasterTime – Slave.CurrentTime);
If ( Error>MinError ) then
If (Slave.BufferIsIn(Master Time+SeekTime) ) then
     Slave.SeekClientTime(Master Time+DelayToSeekClient);
else Slave.SeekServerTime(Master Time+DelayToSeekServer);
Endif
```

In order to assess the system behavior and the improvement 3 different solutions have been taken into account. Solution 0, the early proposed, Solution 1: implementing buffering and taking into account the above described algorithm addressing (i) SeekClientTime; and Solution 2, which has been realized by improving Solution 1 with the management of (ii) SeekServerTime, (iii) DelayToSeekClient, (iv) DelayToSeekServer, and (v) DelayToStart.

7.1 Synchronization performance of MyStoryPlayer

In general, the buffering of videos could improve the quality of the synchronizations among multiple videos on the same player. On the other hand in presence of multiple video streams on limited network bandwidth, setting up a long buffer (for example 10 seconds) would lead to wait for the player to start for many seconds since not all videos starts since the beginning to fill the buffer. However, the limit of bandwidth and the number of media playing simultaneously are issues that could decrease the performances, augmenting the delays among media. For this reason, the buffer has been kept limited to 1 second in both cases, to keep this parameter stable. According to our experiments, the value imposed is a good compromise to get quality without constraining the users to wait for longer time span.

It is difficult to provide measures of performances which cover all the possible cases and conditions. For this reason, two main cases are taken as reference to show the obtained performance and improvements passing from Solution 0 to Solution 2, by using videos of 384x288 pixels, 25 fps, in H264 compression, with audio, for an overall bit rate of 407 Kbps:

Test Case 1: a master video synchronized with 5 videos (One2One relationships) since time 0. This resulted in 6 synchronized videos that should keep (or reach) the synchronization in different network bandwidth conditions and when the user performs jump forward and backward, swaps to one of the related video, etc. This Test Case results to a demand of network bandwidth of 2442 bps, and can be played on ECLAP with MyStoryPlayer by clicking on:

http://www.eclap.eu/portal/?q=msp&axoid=urn%3Aaxmedis%3A00000%3Aobj%3Af3783c88-62bf-4960-ba1c-18da63e783b6&axMd=1&axHd=0

Figure 7.1: test Case 1

Test Case 2: a master video with other 5 One2One relationships of synchronization with other videos starting at 20 seconds of time. This resulted in 6 synchronized videos that should wait for 20 seconds and then start keeping the synchronization in different network bandwidth conditions and when the user performs jump forward and backward, swaps to one of the related video, etc. This Test Case could result in the demand of network bandwidth of 407 kbps for the first 20 seconds, and then of 2442 bps, and can be played on ECLAP portal with MyStoryPlayer by clicking on: *http://www.eclap.eu/portal/?q=msp&axoid=urn%3Aaxmedis%3A00000%3Aobj%3Ad5b311d0-bda8-4f42-81ab-a8f00b059f53&axMd=1&axHd=0*

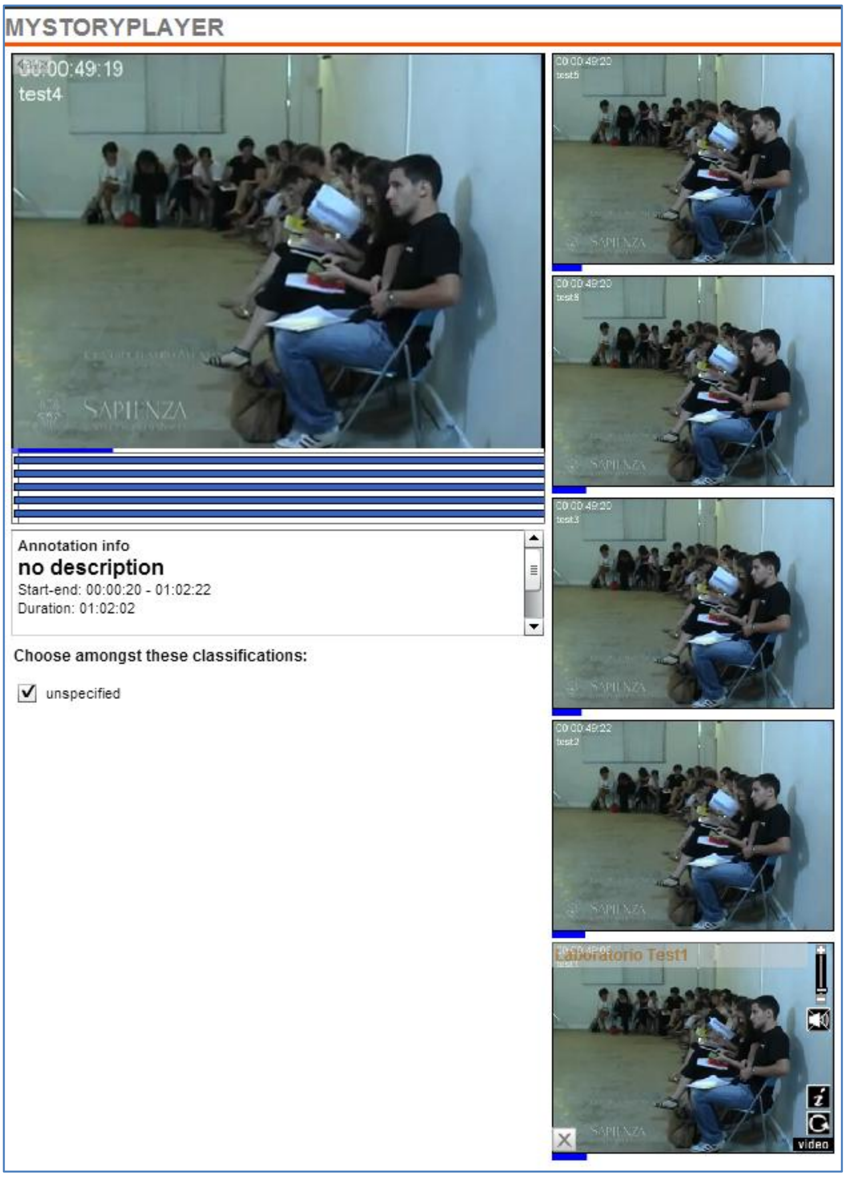
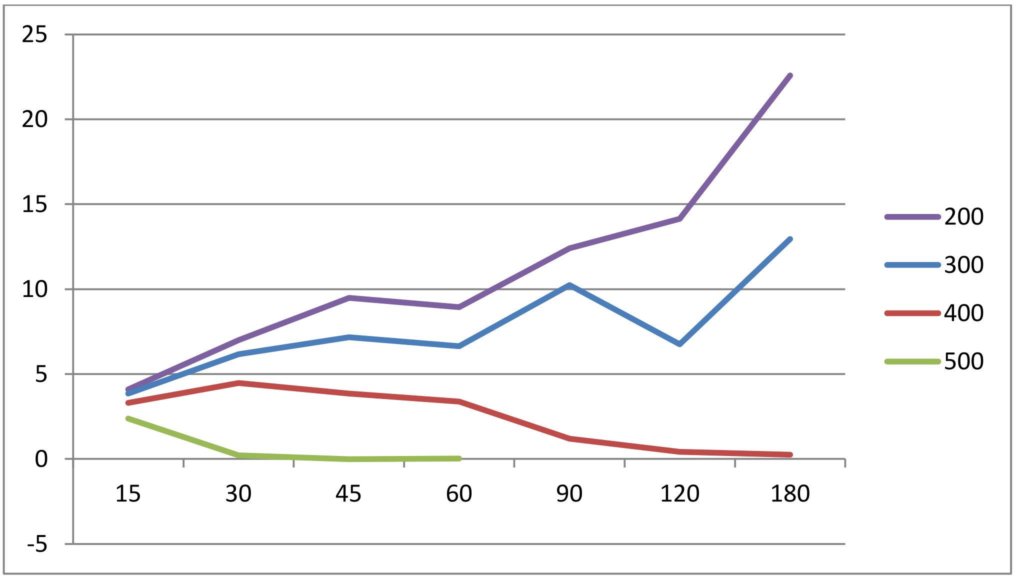


Figure 7.2: Test Case 2

The Solution 0 against TC1 provided an average delay of 11,2s after 60s of play (at 2400 Kbps as maximum bandwidth) with a variance of 32,5 (these values have been estimated by performing 10 tests in the same conditions). At 4000 Kbps the conditions were slightly better providing an average delay of 7,7s and a variance of 15,2. Similar results are obtained with respect to TC2 where it has been measured: at 2400Kbps after 60s, an average of 9,9s of delay with a variance of 30; and at 4000Kbps after 60s, an average of 7,6s with a variance of 12,8. These conditions are obviously not acceptable since lead to very large delay after 1 hour of lesson rendering.

7.1.1 Results of Test Case 1 for Solution 1

a) mean delay



b) mean variance

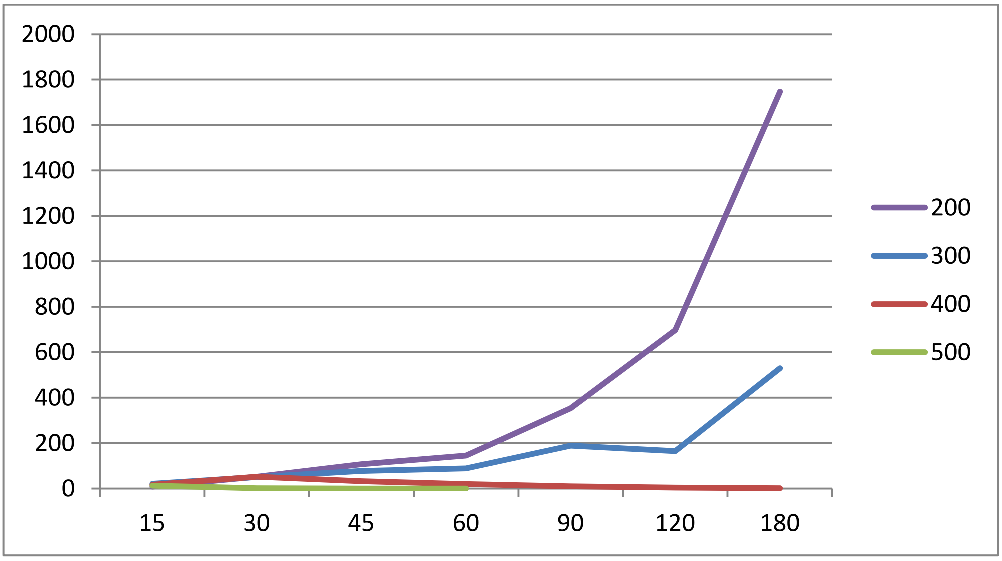


Figure 7.3: a) b) mean delay and variance of Solution 1 for different levels of bandwidth

In figure 7.3 the results of solution 1 are reported. The part *a)* shows the mean delay and the part *b)* the variance. The results are taken after 15, 30, 60, 90, 120, 180 seconds of reproduction of master with 5 synchronized media references for different disposition of bandwidth (1600, 2400, 3200, 4000 Kbps). It is important to notice that for low bandwidth there is instability regarding the variance. This is due to the fact that the bandwidth is not sufficient to play all the media in continuous way, accumulating delay on media references respect to the main. This delay changes very randomly leading to a non convergence for the variance and delay itself. The more a video is left behind, the more his delay tends to augment while the reproduction is going.

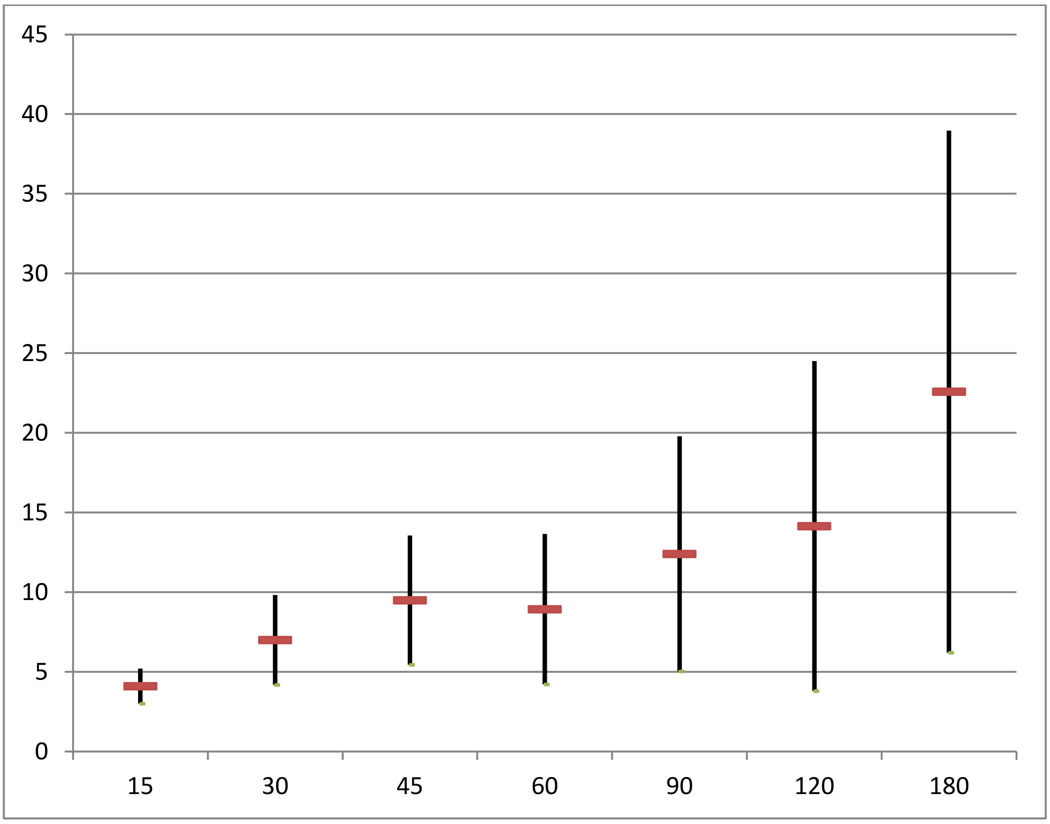


Figure 7.4: Results for bandwidth=1600Kbps

	15sec	30sec	45sec	60sec	90sec	120sec	180sec
Mean delay	4,1016	7,0012	9,4932	8,9316	12,4032	14,1452	22,588
Variance	7,875	51,765	106,869	144,9173	353,9176	697,3801	1747,343

Table 7.1: values of variance and delay for bandwidth=1600Kbps

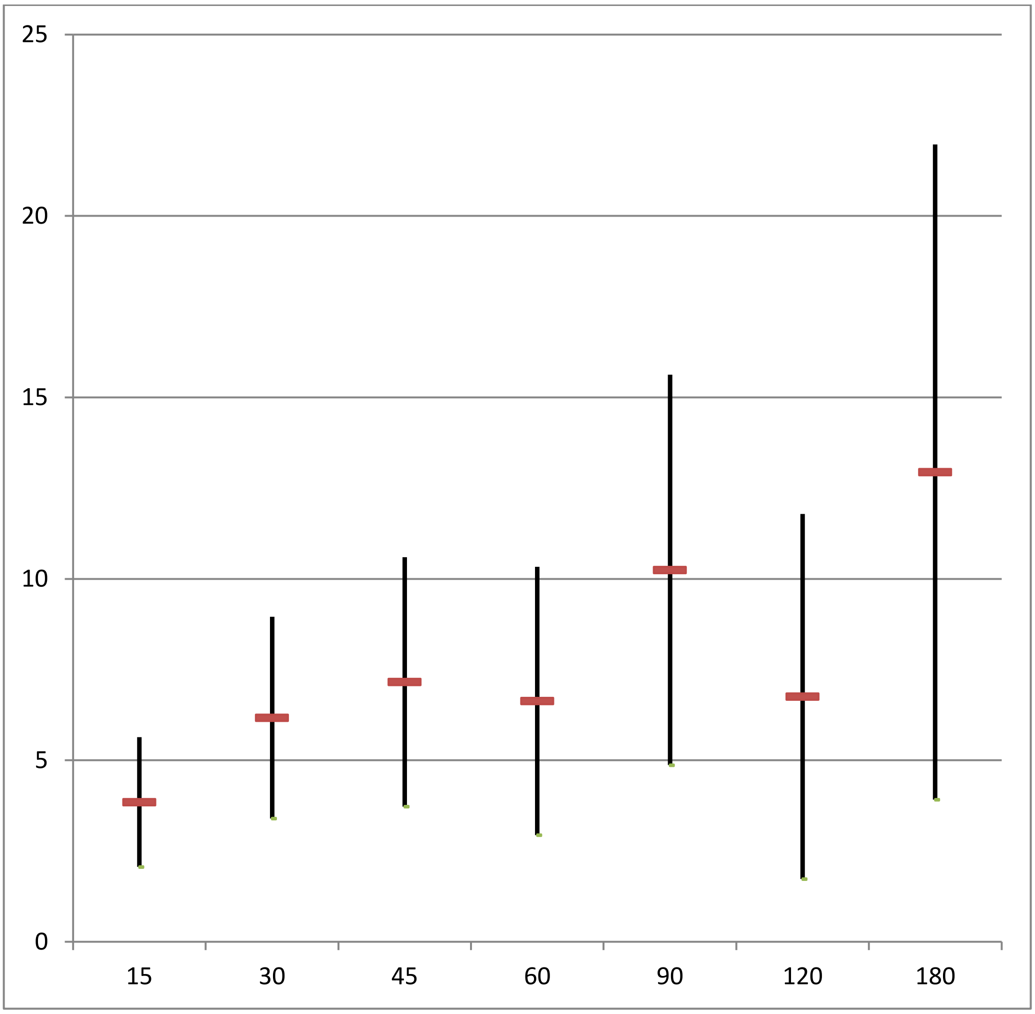


Figure 7.5: Results for bandwidth=2400Kbps

	15sec	30sec	45sec	60sec	90sec	120sec	180sec
Mean delay	3,8484	6,1736	7,1608	6,636	10,246	6,7584	12,9432
Variance	20,787780	50,201649	76,81374	88,94772	188,386	164,6417	530,3619

Table 7.2: values of variance and delay for bandwidth=2400Kbps

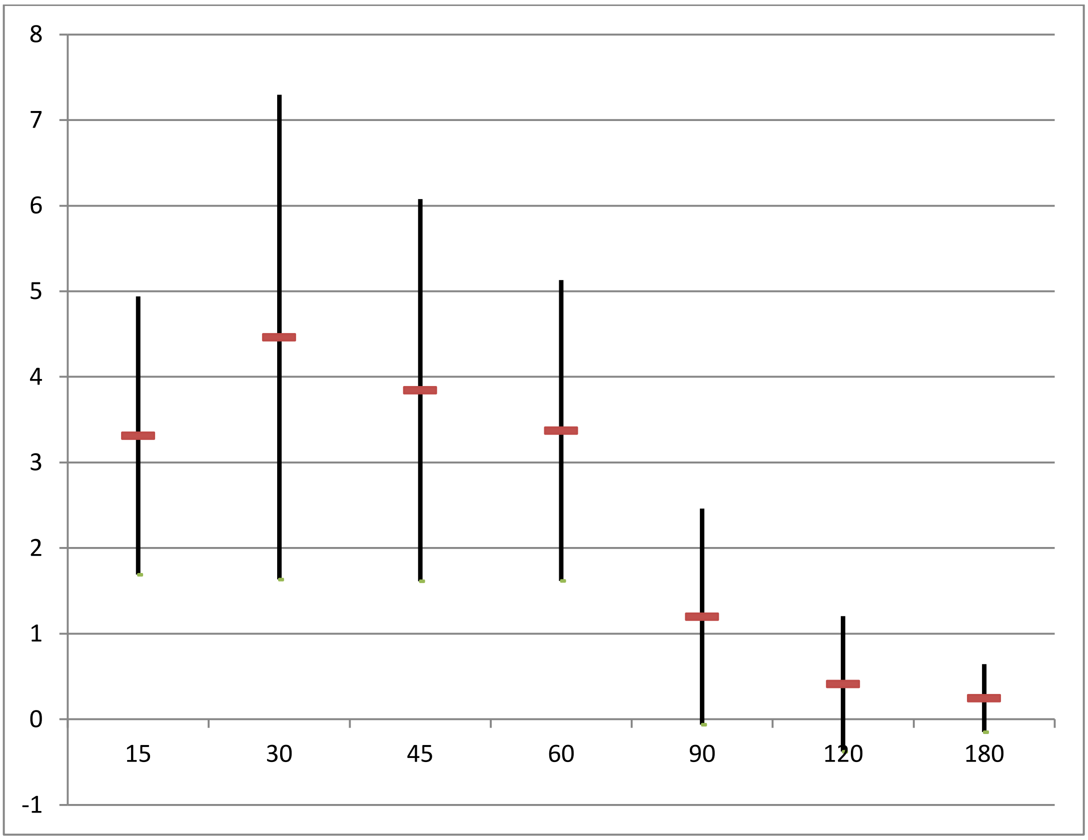


Figure 7.6: Results for bandwidth=3200Kbps

	15sec	30sec	45sec	60sec	90sec	120sec	180sec
Mean delay	3,3144	4,4644	3,8456	3,374	1,2004	0,4144	0,2488
Variance	17,16502	52,145659	32,3961	20,06003	10,36117	4,073759	1,028253

Table 7.3: values of variance and delay for bandwidth=3200Kbps

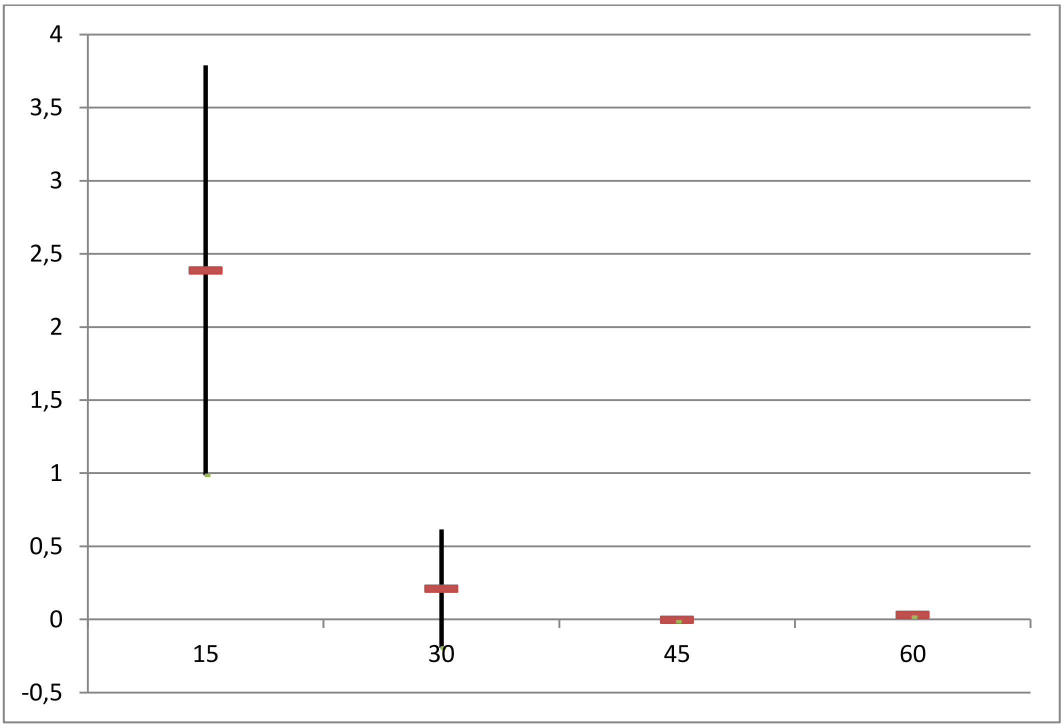


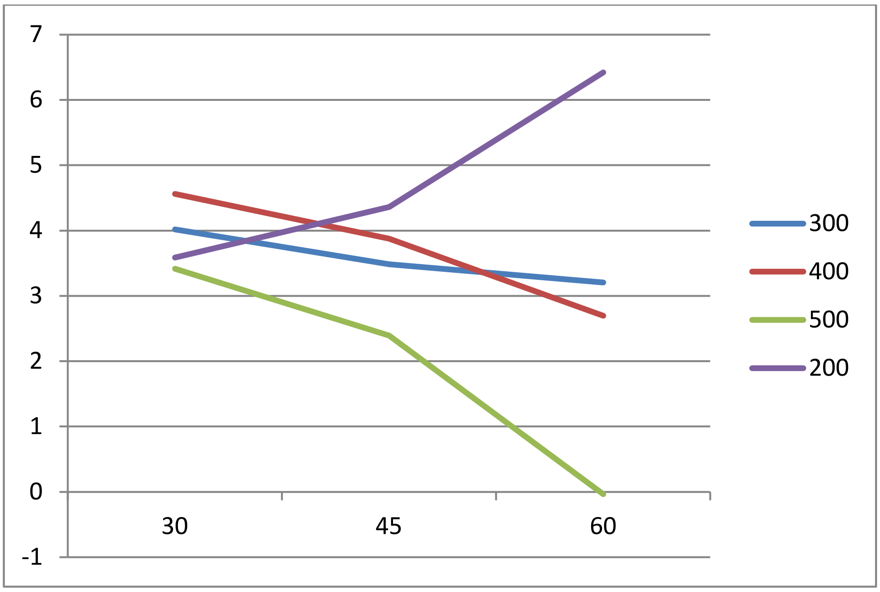
Figure 7.7: Results for bandwidth=4000Kbps

	15sec	30sec	45sec	60sec	90sec	120sec	180sec
Mean delay	2,3876	0,2116	-0,0012	0,0328	--	--	--
Variance	12,775619	1,069630667	0,001103	0,001621	--	--	--

Table 7.4: values of variance and delay for bandwidth=4000Kbps

7.1.1 Results of Test Case 2 for Solution 1

a) mean delay



b) mean variance

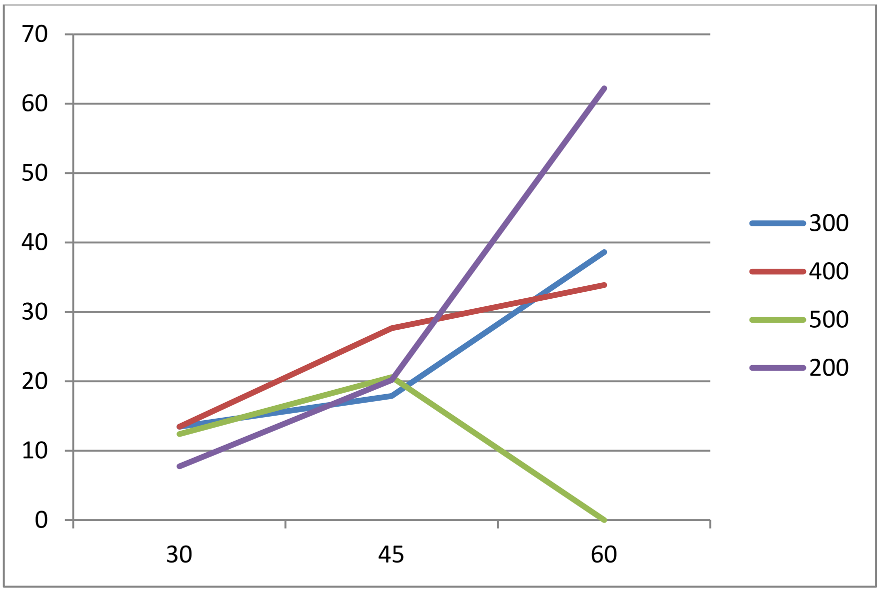


Figure 7.8: a) mean delay of test case 2 for solution 1.

In figure 7.8 the results of solution 1 are reported. The part a) shows the mean delay and the part b) the variance. The results are taken after 30, 45, 60 seconds of reproduction of master with 5 synchronized media references for different disposition of bandwidth (1600, 2400, 3200, 4000 Kbps).
As we can see from the figure 7.8, also in this case the delays tend to augment when we are in presence of low bandwidth (in particular for 1600Kbps). The values of variance augment exponentially for low bandwidth leading to instability and progressively to augmenting delays that keep the application not usable by the users for low bandwidth.

7.1.2 Results Solution 2

In the several experiments, the noticed range for the adaptive corrective values have been estimated to be: *DelayToSeekClient* [0.2s-0.4s], *DelayToSeekServer* [1.2s-6.0s], and *DelayToStart* [1.2s-15.0s]. As described before, their precise value is adaptively computed at run time on the basis of the previous actions. The *DelayToSeekClient* depends on the hardware hosting the MyStoryPlayer Client tool, while the others also depends on network bandwidth. In the case of jump forward and backward from the user, the seek can be inside the loaded buffer or not, provoking delays as *DelayToSeekClient* or *DelayToSeekServer*, respectively. These user provoked delays cannot be anticipated but they can be corrected as unpredicted stop for re-buffering.
For the swap and back actions different considerations have to be done. Each action brings to a change of scenario, so that the delay from a situation to another has to be calculated considering some factors.
First of all, the active Net Stream objects have to be swapped, the graphic interface has to be changed and new information has to be loaded. Note that every time a swap is performed, the player has to save the current state where to come back, cancel all icons, relation rectangles, video objects on the stage and load information about the new media chosen by the user. This goes beyond to a "trivial" operation of swapping two net streams, provoking an augmenting of delay depending on how many objects

have to be erased and how many information have to be loaded. The same concept is applicable to the back action.

The preload of info and videos is not acted, so that the player is slower in the reaction. A possible way to improve this action could be a "double-level" preload of information and video. Double-level preload of information means that for each relation of mater media all the information related to them are loaded, making a preloading also on video related to the first level relations.

Obviously, if there are many video to preload, the risk is that the bandwidth could be saturated leading to a bottleneck and to a decreasing of performances. For this reason we decided to not act a double-level preload, in order to preserve the performances in synchronization especially with low bandwidth.

7.1.2.1 Test Case 1 for solution 2 results

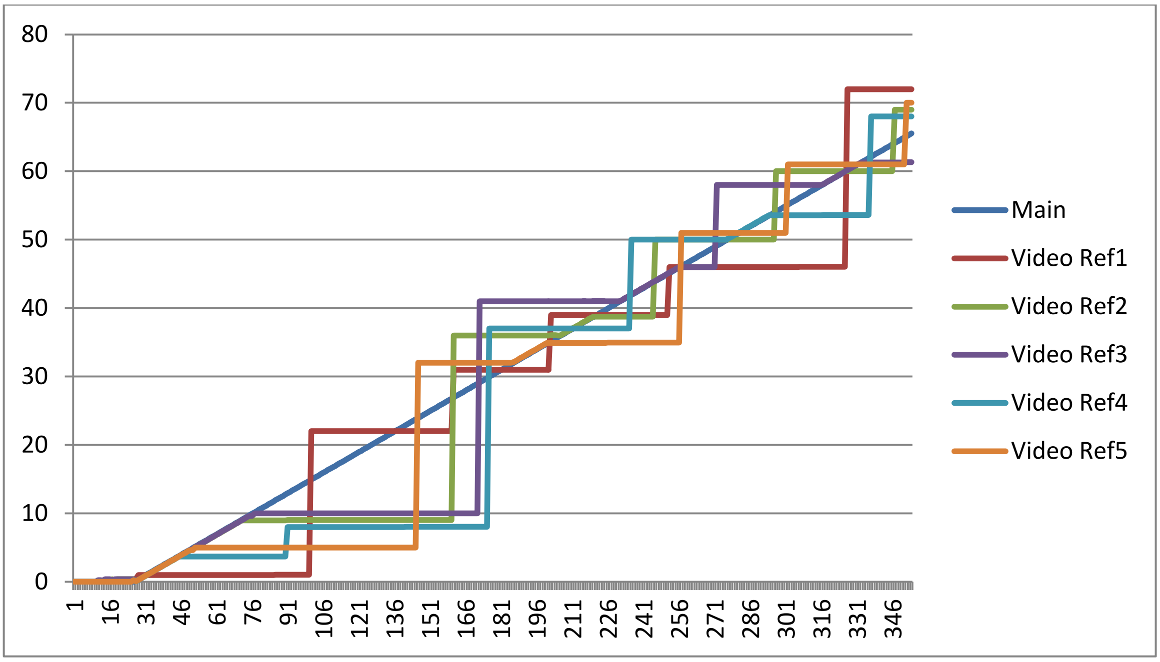


Figure 7.8: Main Video and 5 Followers synchronized, starting at 0. Bandwidth:1600Kbps

It has also been measured the mean delay and the variance of delay for the first three minute of reproduction at some time instants. The results for 1600Kpbs are:

	15 sec	30sec	45 sec	60 sec	90 sec	120 sec	180 sec
Mean Delay	7,6604	5,7608	8,5808	-1,8516	1,4948	0,1036	2,8232
Variance	16,860637	147,62649	257,7139	27,11689	46,57313	50,2486	46,44814

Table 7.5: values of variance and delay for bandwidth=1600Kbps

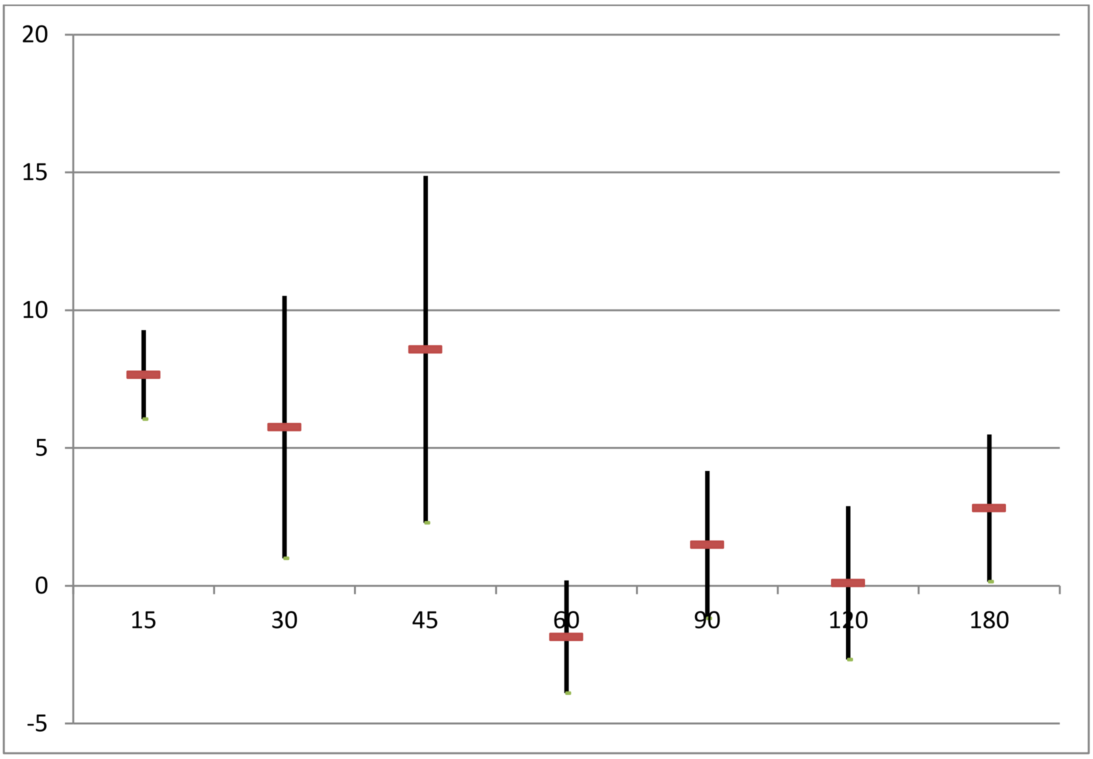


Figure7.9: 1600Kbps

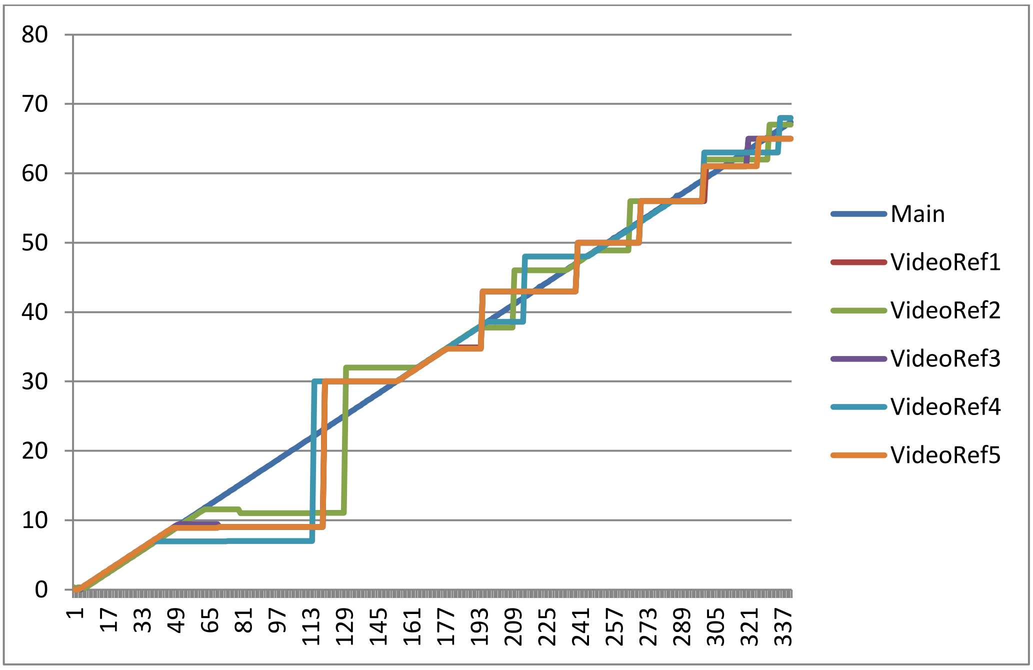


Figure 7.10 – Trends of video time codes into the first 68s in executing TC1 with Solution 2 Bandwidth:2400Kbps.

In other execution of TC1 with low bandwidth, it may happen that the master video run out of buffering, thus constraining all the other videos to stop playing and continue to perform buffering also. Those cases, less critical for the execution since the general execution time is delayed, and the other videos have more time to buffering that in the case presented in Figure 7.10.

	15 sec	30sec	45 sec	60 sec	90 sec	120 sec	180 sec
Mean Delay	3,1588	2,502	-0,3284	2,19424	1,3208	-0,2392	-0,4324
Variance	20,41004433	59,91426667	63,4053	71,58632	72,92442	34,82127	42,89639

Table 7.6: values of variance and delay for bandwidth=2400Kbps

For 2400Kbps it can be noted that: (i) in the first time instants also the master/main video did not started, (ii) the others waited for the its start, (ii) there is a progressive error reduction due to the adaptive correction of the seeking delays, while at the same time some of the videos are perfectly following the master/main video.

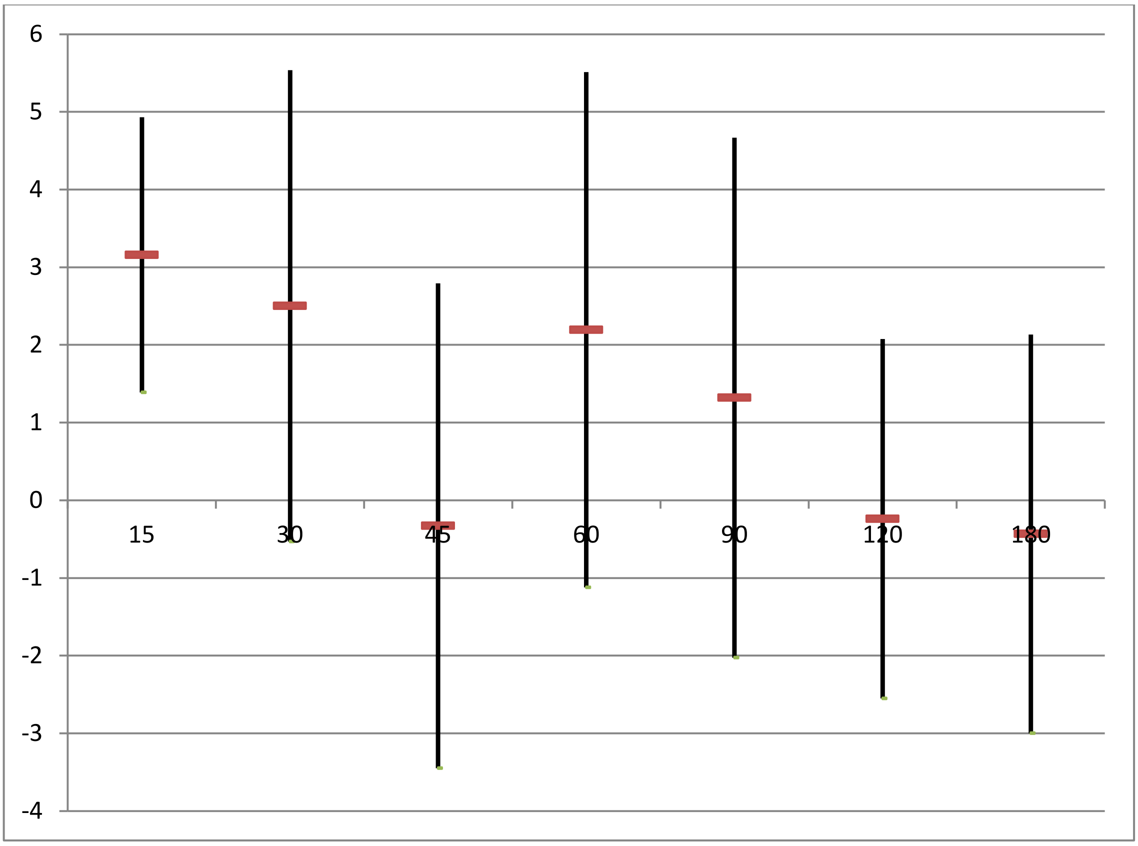


Figure 7.11: 2400Kbps

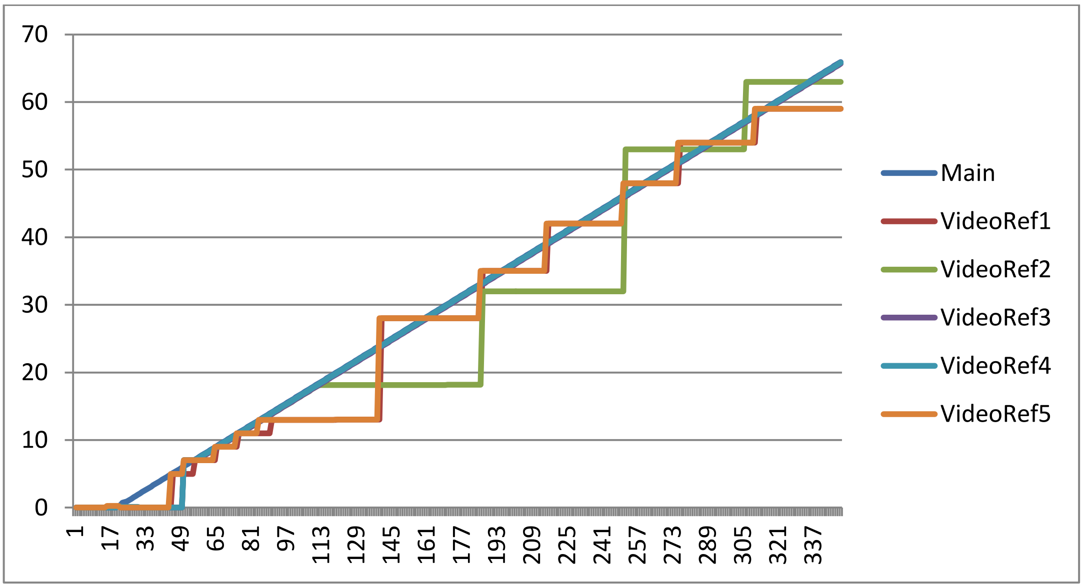


Figure 7.12: 3200 Kbps

	15 sec	30sec	45 sec	60 sec	90 sec	120 sec	180 sec
Mean Delay	1,874	3,4788	0,5312	-0,452	1,0252	-0,392	-0,7436
Variance	18,86281667	43,65129433	3,387011	4,410225	5,350684	5,213533	3,905082

Table 7.7: values of variance and delay for bandwidth=3200Kbps

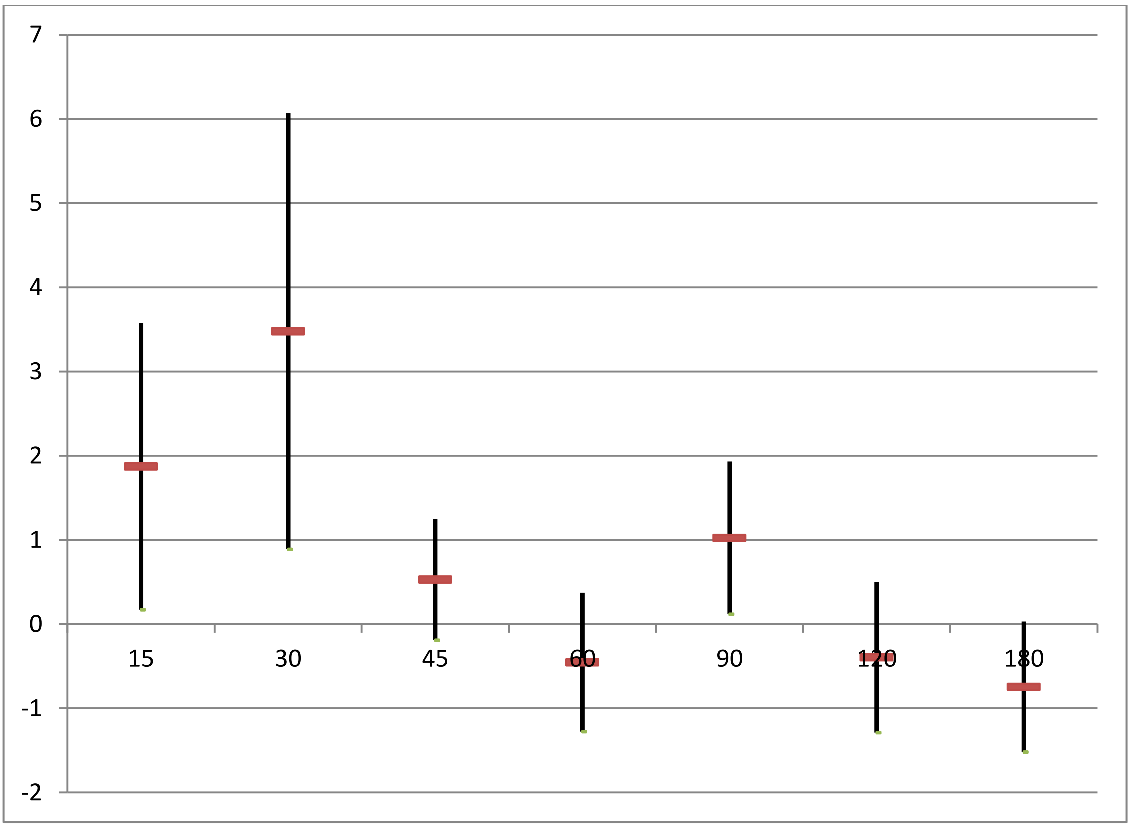


Figure 7.13: 3200Kbps

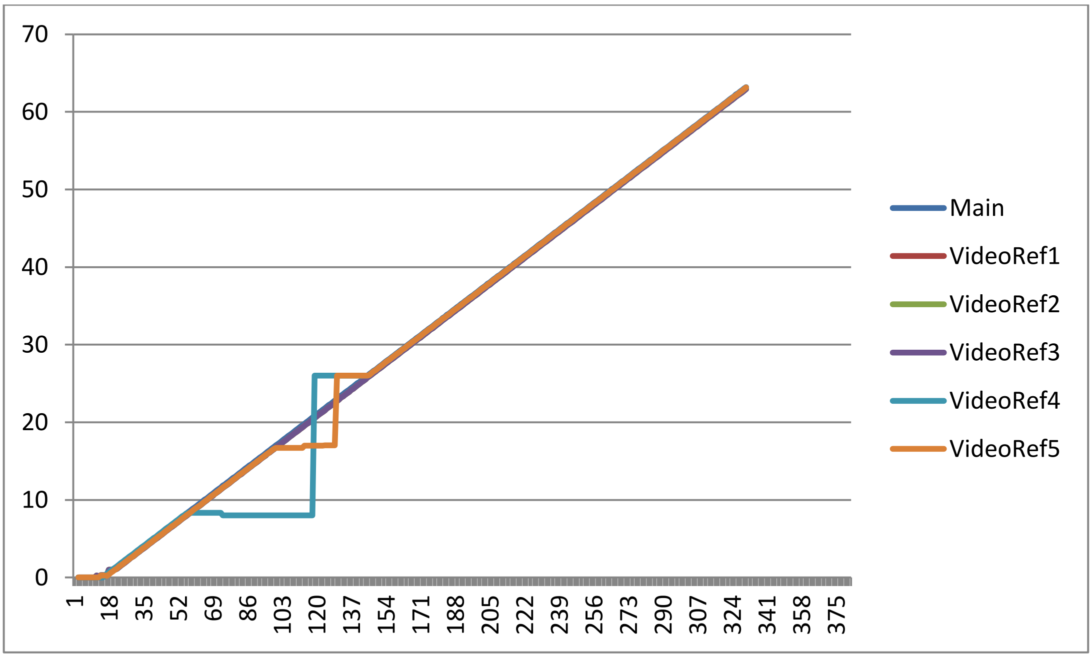


Figure7.14: 4000Kbps

For 4000Kbps data have not been taken since, the delay is minimum and the media converge in few seconds. In the graph of figure 7.14 we can see how the media are synchronized, there are only two little jumps for two media, but in general, these jumps are not performed by the user, since all the reproduction is continuous.

In order to complete the view, Table 7.8 which reports the mean error and variance for the first 70s of Solution 2 run 10 times against of TC1 with different value of network bandwidth.

S2 for TC1	Mean error	Variance
1600Kbps	2,13	39,18
2400Kbps	1,12	16,55
3200Kbps	0,56	11,60
4000Kbps	0,42	2,33

Table7.8: Values estimated for Solution 2 on TC1, in the first 70 seconds.

These graphs represent the behaviors of main video and 5 synchronized video references in the firsts 70 seconds of reproduction for solution 2. As we can see, the jumps performed to align the media references to the main media considering the delay of jump, give good results, stabilizing the media after about 1 minute. As we can see from the graphs, at that time the jumps are more precise and frequent, in order to reach a situation in which the frequency decreases because the difference among media remains constantly in the threshold. The more the bandwidth is small, the more the player jumps. Obviously in case of low bandwidth, at the beginning of the reproduction there are many jumps that make the vision not continue for the users, but considering that the tested video were more than one hour long and that the reproduction becomes quite stable after about one minute, it is possible to say that is a very good result.

Differently from the previous version, with the introduction of this type of seek, it is possible to converge to a situation in which all the media are synchronized.

On the contrary, especially for low bandwidth, the delay tends to diverge as the jumps inside the buffer are not enough to cover the difference between time codes of media reference and master.

7.1.2.2 Test Case 2 for solution 2 results

Below there are results for the test case 2. As we can see, for low bandwidth (1660Kbps-2400Kbps) the followers media reach difficultly the main time, at the beginning of play the jumps are too little, then too much ahead, but in the proximity of one minute of playing, the reproduction converges into a stable situation. Otherwise for high bandwidth, the stability is reached more easily.

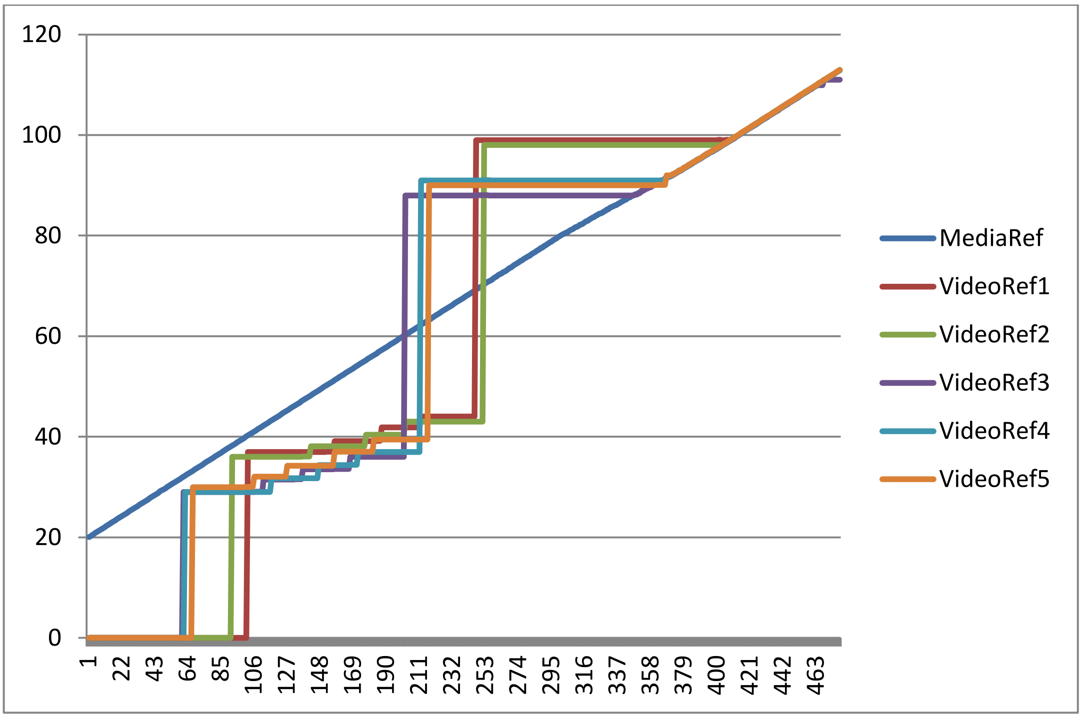


Figure 7.15: Solution 2: 1600Kbps Starting 20 sec.

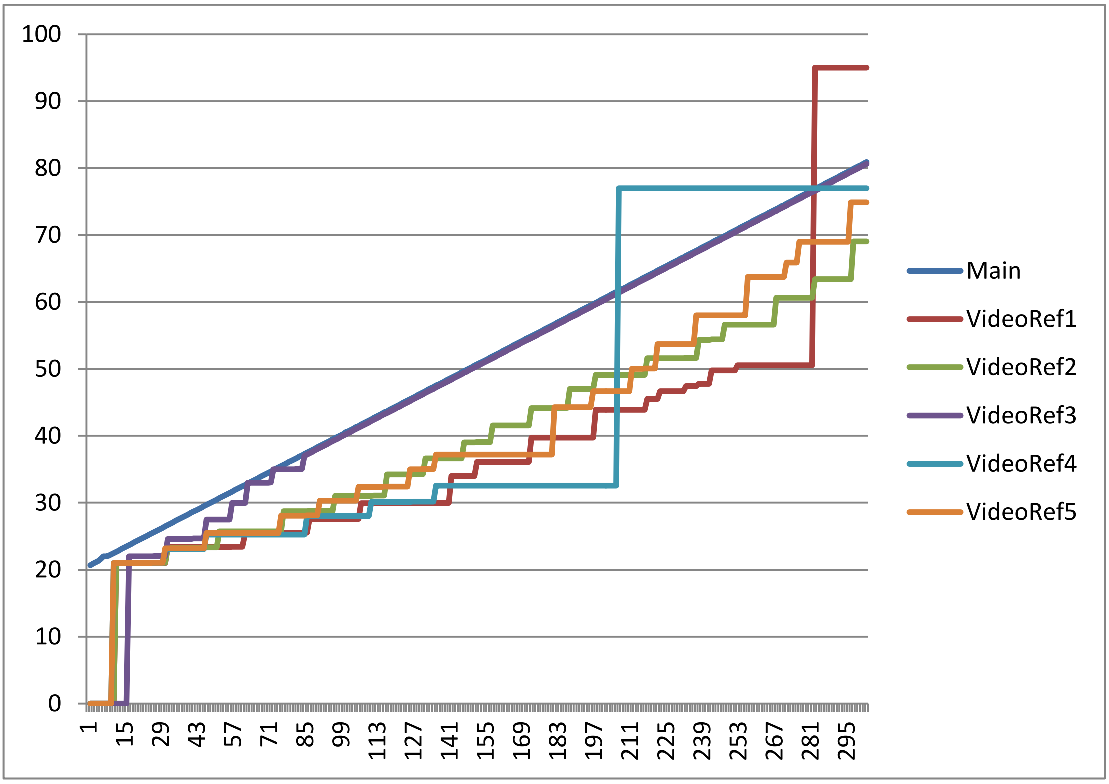


Figure 7.16: Solution 2: 2400Kbps Starting 20 sec.

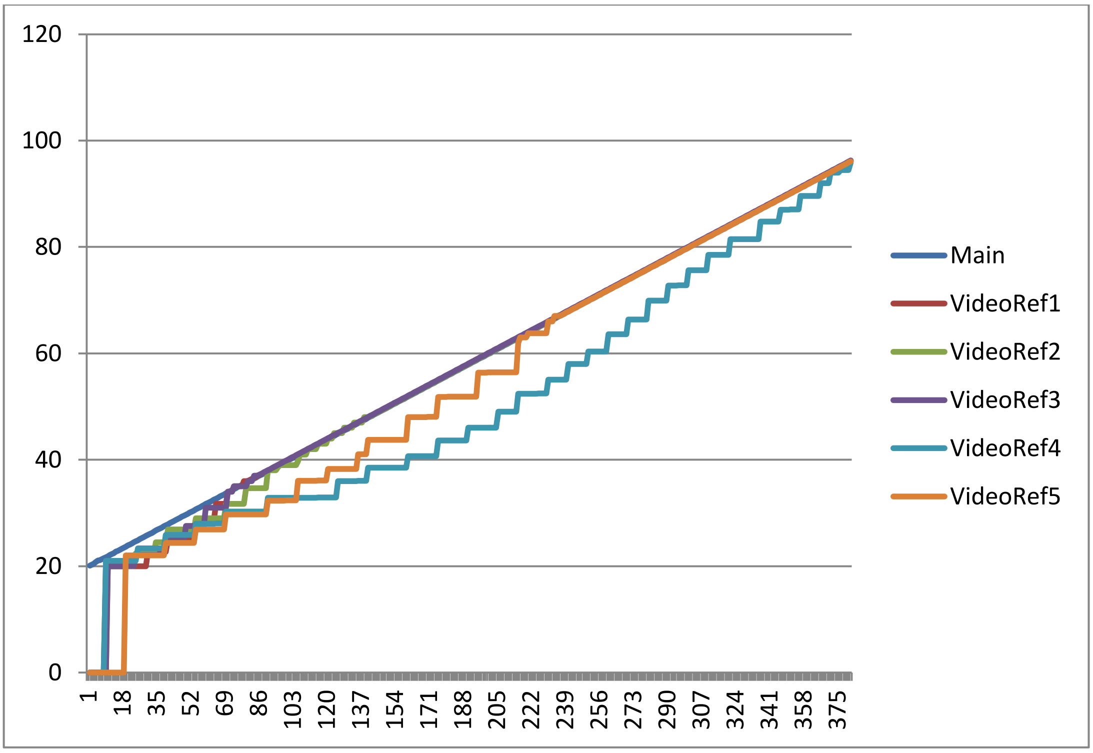


Figure 7.17: Solution 2: 3200Kbps Starting 20 sec.

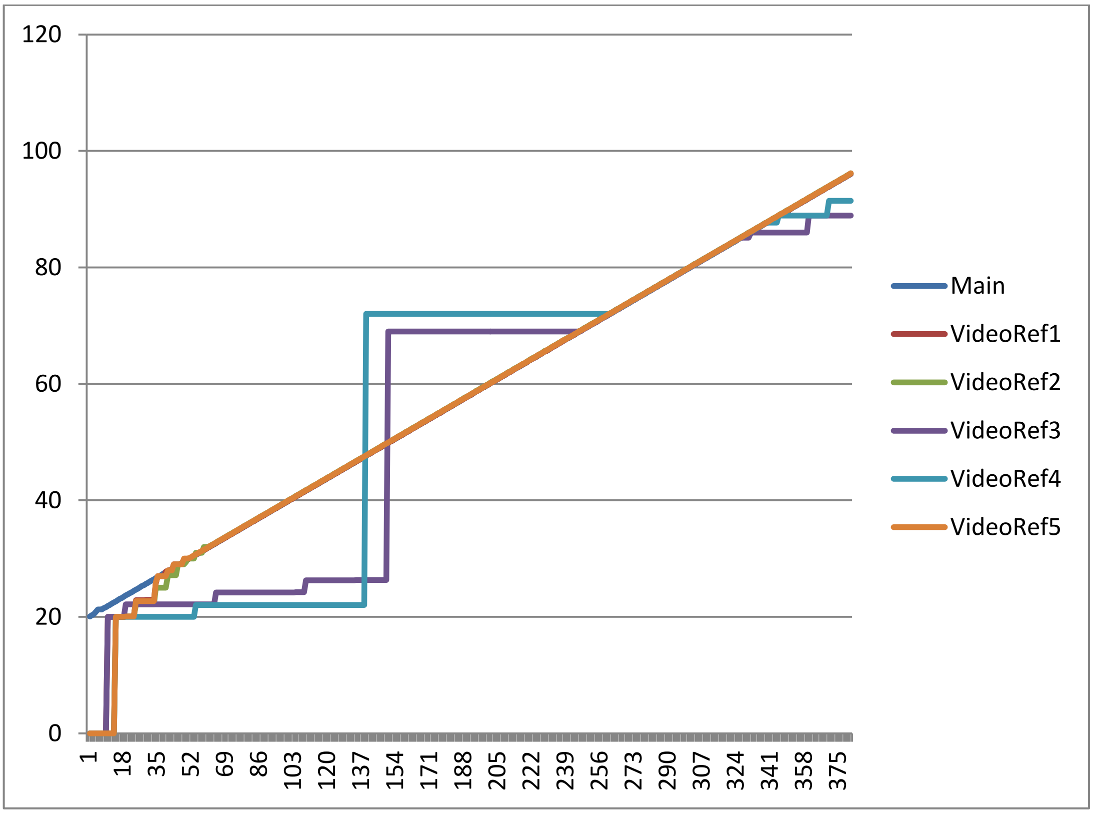


Figure 7.18: Solution 2: 4000Kbps Starting 20 sec.

8.Some Use Data of MyStoryPlayer on ECLAP

MyStoryPlayer tool presented to this article refer to its integration with ECLAP. ECLAP (European Collected Library of Performing Art) is a social network among performing art professionals with more than 170.000 content elements for more than 1.1 million. In ECLAP, the MyStoryPlayer is mainly adopted for: comparing performing art performances, presenting master classes and workshops for performing arts with most of the related content has been provided by CTFR (Centro Teatrale France Rame with content of Dario Fo Nobel Price) and CAT (Centro Teatrale Ateneo University of Rome), plus content coming from other 25 partners from all Europe.

What has been developed is a system able to keep trace of the actions performed by the users on the player, like swap, back, seek to a specific point of the timeline, reloading of new media after a query on the system, and so on. This has been performed to monitor what users do on MyStoryPlayer, in order to understand better which direction the development of the tool had to follow, according to the user behavior on the MyStoryPlayer. What has emerged in the last months is an average of about 10 clicks made by users along their navigation on the player, which is a good result, if you consider that this tool is completely new, with new features, and users need time to accustom themselves to all the proposed features.

At present, there are more than 1.000 relationships involving 415 media (see Figure 8.1 and Table 8.1 for other measures). The accesses are mainly performed by students for master classes and for playing the relationships established by teachers and researchers in the sector.

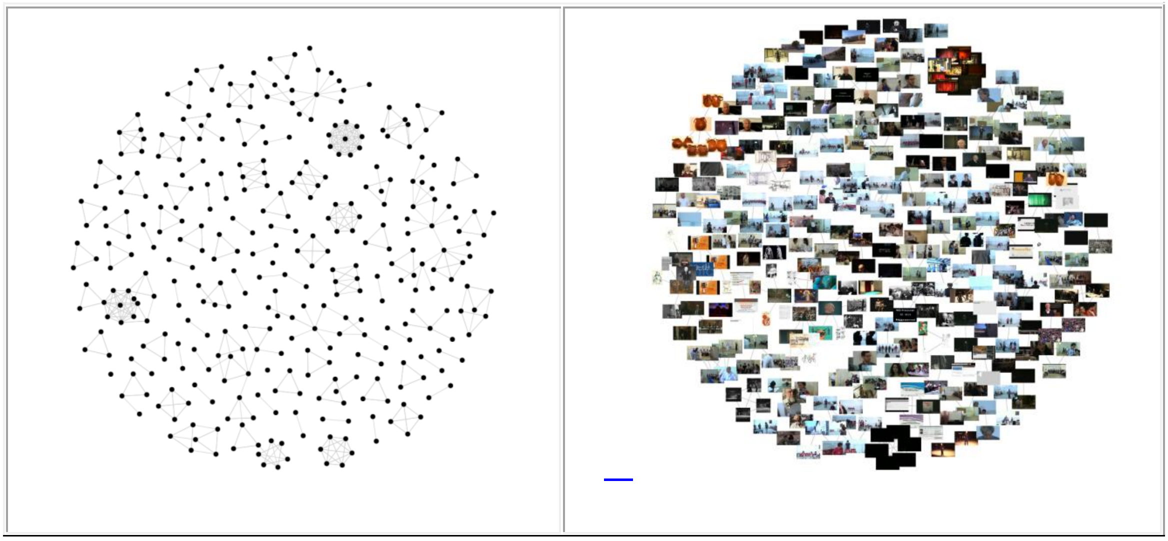


Figure 8.1 – Relationships among audiovisual elements in ECLAP. This analysis can be navigated at the URL: http://www.eclap.eu/d3/graph.html and http://www.eclap.eu/d3/graph2.html

Analyzing the structure of relationships created by users, it is very interesting to see the produced graphs of relationships and their dimension (see Figure 8.1, where nodes are represented by media and edges by the relationships among them). In the full set of ECLAP audiovisual, there are some connected and unconnected elements. In the general set, it has been possible to identify a number of separate groups of media. Inside each group, a number of relationships have been defined, for example putting in relationship the video sequences, the video synchronizations with several additions performed to provide examples and similarity annotations and audiovisual comments. In Table 8.1, some measures have been reported to provide the reader with a general idea about the relationships established among the media. Moreover, among the largest set of media relations we have: 22 Media related by 26 Relations (single and reciprocal), described as (22M, 26R), other relevant examples are: (24M, 34R), (14M, 67R), (10M, 90R), etc. There are also a few of very simple examples (1M,1R), where the same media has been annotated with itself (for example, two different segments of the same video).

Number of defined relationships	1086
Number of media involved in some relationships	562
Only as a recipient of a relationship	46
Both as target and recipient of some relationships	272
Average number of relationships per media	1.93
Maximum number of relationships per media	11
Number of additional simple textual annotations on media segments	135

Table 8.1 – Metrics describing the usage of MyStoryPlayer model among media into the RDF database.

9. Conclusions

The work presented in this thesis addressed the problems of education and training cases in which multi-camera view are needed: performing arts and news, medical surgical action, sport actions, play instruments, speech training, etc. In most of these cases, the users (both teachers and students) need to interact to establish among audiovisual segments relations and annotations with the purpose of: comparing actions, gesture and posture; explaining actions; providing alternatives, etc. In order to keep limited the desynchronizations problem among audiovisual, most of the state of the art solutions are based on custom players and/or on specific applications that constrain to create custom streams from server side; thus leading to restrictions on the user activity on dynamically establishing relations and on accessing to the lessons via web. In this thesis, MyStoryPlayer/ECLAP solution has been presented, providing: (i) a semantic model to formalize the relationships and play among audiovisual determining synchronizations (One2One, Explosion, Reciprocal, and Sequential), (ii) a model and modality to save and share user experiences in navigating among related audiovisual, (iii) solution for shortening the production of relations, (iv) the architecture and the design of the whole system including the interaction model, and finally (v) the solution and algorithm to keep limited the desynchronizations among media in the presence of low network bandwidth. The resulting solution includes a uniform semantic model, a corresponding semantic database for the knowledge, a distribution server for semantic knowledge and media, and the MyStoryPlayer Client for web applications. The solution proposed has been validated and it is presently in use of ECLAP (European Collected Library of Performing Arts, http://www.eclap.eu) for accessing and commenting performing arts training content. The thesis also reported validation results about performance assessment and tuning about the media synchronization in critical cases. The validation test demonstrated the proposed solution is suitable for rendering multiple synchronized media via web in presence of low bandwidth, providing to the user the possibility of performing jumps (backward and forward), swap and back among media, etc. In ECLAP, the users may navigate in the audiovisual relationships, thus creating and sharing experience paths, presently several media relationships have been created and area accessible to all users and students of institutions associated to ECLAP.

Appendix

ECLAP Consortium brings together European leading national performing arts institutions, universities and research institutes. The partners belong to thirteen countries, and combine their expertise and scientific minds to achieve the ECLAP goals see partners' page on the portal:

Content Providers:

- *BELLONE, La Maison du Spectacle, Belgium*
- *Beeld en Geluid, (Sound & Vision), The Netherlands, content provider and technology provider*
- *CTFR, Dario Fo & Franca Rame Archive, Italy*
- *ESMAE-IPP,Escola Superior de Música e das Artes do Espectáculo do Porto, Portugal*
- *FIFF, Festival International de Films de Femmes de Créteil, France*
- *IKP, The Institute of Polish Culture University of Warsaw, Poland*
- *ITB, Museu de les Arts Escèniques Institut del Teatre de Barcelona, Spain*
- *MUZEUM, Muzeum, Slovenia*
- *OSZMI, Hungarian Theatre Institute, Hungary*
- *UCAM, Museum of Archaeology & Anthropology, University of Cambridge, UK*
- *UCLM,Universidad de Castilla La Mancha,Spain*
- *UG, History of Art Department at the University of Glasgow, UK*
- *UNIROMA, Centro Teatro Ateneo, University of Rome La Sapienza, Italy*
- *UVA, Department of Theatre Studies, University of Amsterdam, The Netherlands*
- *Technology Providers:*

- *DSI, Department of Systems and Informatics, University of Florence, Italy, Technology provider, technical coordinator and project coordinator*
- *NTUA, National Technical University, of Athens, Greece, Technology provider*
- *AXMEDIATECH, Axmediatech Srl, Italy, Technology provider*
- *Networking partners:*
- *FRD, Fondazione Rinascimento Digitale, Italy, dissemination and networking, UG management*
- *Affiliated partners:*
- *ArchiBraille Music, Italy*
- *FFEAC: Fondazione Fabbrica Europa, Italy*
- *IG, Grotowski Institute, Poland*
- *NINA, Narodowny Institut Audiowizualny, Poland*
- *DIGILAB, center for research in digital arts and humanities, Italy*
- *IRTEM, istituto di ricerca per il teatro musicale, Italy*
- *ANSC, Accademia Nazionale Santa Cecilia, Italy*
- *TeatroNapoletano, Italy*
- *DISAH, Department of Information Systems for Arts and Humanities, Russia*
- *DigitalMeetsCulture, Italy*
- *Cielos - Festival Cielos del Infinito (Chili)*
- *LiberLiber - Liber Liber (Italy)*

Bibliography

- [Adowd, 2000] Abowd G. D., “Classroom 2000: An experiment with the instrumentation of a living educational environment”, http://www.research.ibm.com/journal/sj/384/abowd.html
- [AXMEDIS] AXMEDIS, “Framework and Tools Specifications”, http://www.axmedis.org
- [Bellini and Nesi, 2013] Pierfrancesco Bellini, P. Nesi, "A Linked Open Data service for performing arts", Proc. of the ECLAP 2013 conference, 2nd International Conference on Information Technologies for Performing Arts, Media Access and Entertainment, Springer Verlag LNCS, 2013
- [Bellini et al., 2006] Bellini P., P. Nesi, L. Ortimini, D. Rogai, A. Vallotti, “Model and usage of a core module for AXMEDIS/MPEG21 content manipulation tools”, IEEE International Conference on Multimedia & Expo (ICME 2006), Pages 577--580 Toronto, Canada, 9-12 July, 2006.
- [Bellini et al., 2009] P. Bellini, P. Nesi, D. Rogai, “Expressing and Organizing Real Time Specification Patterns via Temporal Logics,” Journal of Systems and Software, Elsevier, pp. 1–36, Vol.82, N.2, Feb. 2009.
- [Bellini et al., 2011] Bellini P., P. Nesi, M. Paolucci, M. Serena, "Models and tools for content aggregation and audiovisual cross annotation synchronization", Proceeding of ISM '11 Proceedings of the 2011 IEEE International Symposium on Multimedia, Pages 210-215, ISBN: 978-0-7695-4589-9, doi>10.1109/ISM.2011.41, 5-7 December 2011, Dana Point(Cam USA).
- [Bellini, Bruno, Nesi, 2011] Bellini P., I. Bruno, P. Nesi, "EXPLOITING INTELLIGENT CONTENT VIA AXMEDIS/MPEG-21 FOR MODELLING AND DISTRIBUTING NEWS", International Journal of Software Engineering and Knowledge Engineering, World Scientific Publishing Company, Vol.21, n.1, pp.3-32., 2011.
- [Bellini, Cenni, Nesi, 2012] Bellini P., D. Cenni, P. Nesi, "On the Effectiveness and Optimization of Information Retrieval for Cross Media Content", Proceeding of the

KDIR 2012 is part of IC3K 2012, International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management, 4-7 October 2012, Barcelona, Spain.

- [Bellini, Nesi, Rogai, 2007] Bellini P., P. Nesi, D. Rogai. "Exploiting MPEG-21 File Format for Cross Media Content". 13th International Conference on Distributed Multimedia Systems (DMS), San Francisco Bay, USA, 6-8 September 2007.
- [Bellini, Nesi, Serena, 2011] Pierfrancesco Bellini, Paolo Nesi, Marco Serena, Mystoryplayer: Semantic Audio Visual Annotation And Navigation Tool, proc of the 17th international conference on Distributed Multimedia Systems, Convitto della Calza, Florence, Italy, 18-20 August 2011
- [Boll and Klas, 2001] S. Boll and W. Klas. 2001. ZYX-A Multimedia Document Model for Reuse and Adaptation of Multimedia Content. IEEE Transactions on Knowledge and Data Engineering, May/June 2001, 361-382. DOI=10.1109/69.929895 http://doi.ieeecomputersociety.org/10.1109/69.929895
- [Broekstra et al., 2002] Broekstra Jeen, Arjohn Kampman, and Frank van Harmelen, "Sesame: A Generic Architecture for Storing and Querying RDF and RDF Schema", First International Semantic Web Conference (ISWC'02), Sardinia, Italy, June 9-12, 2002.
- [Bulterman and Hardman, 2005] D. C. A. Bulterman and L. Hardman, "Structured multimedia authoring," *ACM Transactions on Multimedia Computing, Communications, and Applications*, vol. 1, no. 1, pp. 89–109, Feb. 2005.
- [Bulterman, Rutledge, 2009] D.C.A Bulterman and L.W. Rutledge. SMIL3.0 - Interactive Multimedia for Web, Mobile Devices and DAISY Talking Books. Springer-Verlag, 2009. ISBN: 978-3-540-78546-0
- [Burnet et al., 2005] Burnett, I.S., Davis, S.J., Drury, G. M., "MPEG-21 digital item declaration and Identification-principles and compression", IEEE Transactions on Multimedia, Vol.7, N.3, pp.400-407, 2005.
- [Chilamkurti et al., 2010] Chilamkurti N, Zeadally S, Soni R, Giambene G, Wireless multimedia delivery over 802.11e with cross-layer optimization techniques. Multimed Tools Appl 47:189–205. doi:10.1007/s11042-009-0413-6, 2010

- [Dakss et al., 1998] Dakss J., Stefan Agamanolis, Edmond Chalom, V. Michael Bove, Jr., ``Hyperlinked Video,'' Proc. SPIE Multimedia Systems and Applications, v. 3528, 1998.
- [DC] DC: Dublin Core Metadata Initiative, http://dublincore.org/
- [ECLAP] http://www.eclap.eu,
- [Gaggi and Danese, 2011] Ombretta Gaggi and Luca Danese, A SMIL player for any web browser, Proc. Of Distributed Multimedia Systems, DMS2011, Florence, Italy, pp.114-119, 2011.
- [Gao et al., 2011] Bo Gao, Jack Jansen, Pablo Cesar, and Dick C.A. Bulterman. 2011. Accurate and low-delay seeking within and across mash-ups of highly-compressed videos. In Proceedings of the 21st international workshop on Network and operating systems support for digital audio and video (NOSSDAV '11). ACM, New York, NY, USA, 105-110. DOI=10.1145/1989240.1989266 http://doi.acm.org/10.1145/1989240.1989266
- [Hausenblas, 2008] Hausenblas Michael: "Non-linear interactive media productions" Multimedia Systems, Vol 14, Number 6, December 2008, pg 405-413 DOI: 10.1007/s00530-008-0131-3
- [Kahan et al., 2001] Kahan, J., Koivunen, M., Prud'Hommeaux, E., and Swick, R., "Annotea: An Open RDF Infrastructure for Shared Web Annotations", in Proc. of the WWW10 International Conference, Hong Kong, May 2001 http://www10.org/cdrom/papers/488/index.html http://www.w3.org/2001/Annotea/
- [Klamma et al., 2006] Klamma, R., Marc Spaniol, Dominik Renzel, "Virtual Entrepreneurship Lab 2.0: Sharing Entrepreneurial Knowledge by Non-Linear Story-Telling", RWTH Aachen University, Germany, Journal of Universal Knowledge Management), Springer, 1(3):174–198, 2006.
- [Koivunen et al., 2003] Koivunen, M., Swick, R., and Prud'hommeaux, E., "Annotea Shared Bookmarks", 2003, In Proc. of KCAP 2003, http://www.w3.org/2001/Annotea/Papers/KCAP03/annoteabm.html
- [Kosovic et al., 2004] Kosovic D., Ronald Schroeter, Jane Hunter, "FilmEd: Collaborative Video Annotation, Indexing and discussion tool over Broadband

networks", Proceedings of the 10th International Multimedia Modelling, Page: 346 , 2004.

- [Layaida et al., 2002] Nabil Layada, Loay Sbry-Ismail, Cecile Roisin, "Dealing with Uncertain Durations in Synchronized Multimedia Presentations", Multimedia Tools and Applications, Kluwer Academic Publishers, vol.18, pp.213–231, December 2002, 213-231. DOI=10.1023/A:1019944800320 http://dx.doi.org/10.1023/A:1019944800320
- [Lee and Park, 2010] Lee I, Park J, A scalable and adaptive video streaming framework over multiple paths. Multimed Tools Appl 47:207–224. doi:10.1007/s11042-009-0414-5, 2010
- [LigneDeTemps] Ligne de Temps http://www.iri.centrepompidou.fr/outils/lignes-de-temps/
- [Lombardo and Damiano, 2012] Lombardo Vincenzo, Rossana Damiano, "Semantic Annotation of narrative media objects", *Multimedia Tools and Applications*, vol. 59, no. 2, pp. 407–439, May 2011.
- [Meixner and Hoffmann, 2012] B. Meixner and J. Hoffmann, "Intelligent download and cache management for interactive non-linear video," *Multimedia Tools and Applications*, Jun. 2012.
- [Meixner and Kosh, 2012] Britta Meixner and Harald Kosch. 2012. Interactive non-linear video: definition and XML structure. In Proceedings of the 2012 ACM symposium on Document engineering (DocEng '12). ACM, New York, NY, USA, 49-58. DOI=10.1145/2361354.2361367 http://doi.acm.org/10.1145/2361354.2361367
- [MPEG-21 DIS] MPEG-21 DIS: http://mpeg.chiariglione.org/technologies/mpeg-21/mp21-dis/index.htm
- [MPEG-7] MPEG-7: http://mpeg.chiariglione.org/standards/mpeg-7/mpeg-7.htm
- [MXF] AAF/MXF: Multimedia Exchange Format, http://www.ist-nuggets.tv/mxf
- [NCL] ITU-T Rec. H.761, Nested Context Language (NCL) and Ginga-NCL for IPTV Services, Geneva, Apr. 2009. Available at http://www.itu.int/rec/T-REC-H.761. Last access on June 25th 2012.
- [Neuschmied, 2007] H. Neuschmied, "MPEG-7 Video Annotation Tool Media Analyze – Pre-processing Tool MPEG-7 Video Annotation Tool," pp. 5–7, 2007.

- [OpenAnnotation] of W3C, http://www.openannotation.org/ , http://www.openannotation.org/spec/core/
- [OverlayTV] OverlayTV: http://www.overlay.tv/
- [Pereira and Ebrahimi, 2002] F. Pereira, T. Ebrahimi, eds. (2002), "The MPEG-4 Book" IMSC Press, Prentice Hall.
- [RDF] semantic Web Standard, http://www.w3.org/RDF/
- [Schroeter et al., 2003] Schroeter, R., J. Hunter, D. Kosovic, "Vannotea – A Collaborative Video Indexing, Annotation and Discussion System for Broadband Networks", proc of Knowledge Markup and Semantic Annotation Workshop, K-CAP 2003, USA, October, 2003.
- [SCORM] Advanced Distributed Learning, "The Sharable Content Object Reference Model (SCORM) - Version 1.3 - WD," March 2003
- [SESAME] Sesame: http://www.openrdf.org/doc/sesame/users/index.html
- [Shen et al., 2009] Shen, E., Lieberman, H., and Davenport, G. (2009). "What's next?: emergent storytelling from video collection". In CHI '09: Proceedings of the 27th international conference on Human factors in computing systems, pages 809{818, New York, NY, USA. ACM.
- [Shipman et al., 2008] Shipman Frank, Girgensohn Andreas WilcoxLynn: "Authoring, Viewing, and Generating Hypervideo: An Overview of Hyper-Hitchcock", 2008 ACM Trans. Multimedia Commun. Appl. 5,2, Article 15, 19 pages.
- [Smith, Lugeon, 2000] Smith J. R., B. Lugeon, "A Visual Annotation Tool for Multimedia Content Description", Proc. SPIE Photonics East, Internet Multimedia Management Systems, November 2000.
- [SPARQL] SPARQL Query Language for RDF, http://www.w3.org/TR/rdf-sparql-query/
- [Steves et al., 2000] Steves M. P., M. Ranganathan, E.L. Morse: "SMAT: Synchronous Multimedia and Annotation Tool," Proceedings of the International Conference on System Science, September, 2000.
- [Tsai et al., 2010] Tsai MF, Shieh CK, Ke CH, Deng DJ, Sub-packet forward error correction mechanism for video streaming over wireless networks. Multimed Tools Appl 47:49–69. doi:10.1007/s11042-009-0406-5, 2010

- [Ursu et al., 2008] Ursu Marian F., Maureen Thomas, Ian Kegel, Doug Williams, Mika Tuomola, Inger Lindstedt, Terence Wright, Andra Leurdijk, Vilmos Zsombori, Julia Sussner, Ulf Myrestam, and Nina Hall. 2008. Interactive TV narratives: Opportunities, progress, and challenges. ACM Trans. Multimedia Comput. Commun. Appl. 4, 4, Article 25 (November 2008).
- [YouTube] YouTube Creating or editing annotations.Website: https://support.google.com/youtube/?hl=en-GB#topic=2676319
- [Zhai et al., 2005] Zhai G., G. Fox, M. Pierce, W. Wu, H. Bulut, “eSports: Collaborative an Synchronous Video Annotation System in Grid Computing Environment”, Proc. of Seventh IEEE International Symposium on Multimedia, 12-14 December, CA, USA, 2005.
- [EDM] EDM, Europena. (2010). Europeana Data Model Primer. Technical report. August 2010. Retrieved April 30, 2011, from http://version1.europeana.eu/web/europeana-project/technicaldocuments/ http://pro.europeana.eu/documents/900548/bb6b51df-ad11-4a78-8d8a-44cc41810f22
- [Mayer-Patel, Gotz, 2007] Mayer-Patel K., David Gotz, "Scalable, Adaptive Streaming for Nonlinear Media," IEEE MultiMedia, vol. 14, no. 3, pp. 68-83, July-Sept. 2007, doi:10.1109/MMUL.2007.63.
- [LIDO] LIDO - Lightweight Information Describing Objects, Richard Light, Gordon McKenna, Regine Stein, Axel Vitzthum, ATHENA project deliverable D3.3, November 2009, http://www.athenaeurope.org/getFile.php?id=535
- [EAD] http://www.loc.gov/ead/
- [METS] http://www.loc.gov/standards/mets/

Publications of the author

- P. Bellini, P. Nesi, M. Serena, "Mystoryplayer: Semantic Audio Visual Annotation And Navigation Tool". DMS 2011, Firenze 18-20 Agosto, 2011.
- P. Bellini, I. Bruno, D. Cenni, P. Nesi, M. Paolucci, M. Serena, Semantic Model for Cultural Heritage Social Network and Cross Media Content for Multiple Devices AI*IA, workshop beni culturali, Palermo 2011.
- P. Bellini, P. Nesi, M.Paolucci, M. Serena, MyStoryPlayer, Collections, Playlists: Content aggregator tools in Eclap Project, Demo paper at DMS 2011, Firenze 18-20 Agosto, 2011.
- P.Bellini, P.Nesi, M.Paolucci, M.Serena, ECLAP: European Collected Library of Performing Art, Demo paper at NEM 2011, Torino 27-29 settembre 2011.
- P.Bellini, P.Nesi, M.Paolucci, M.Serena, Models and tools for aggregating and annotating content on ECLAP, IEEE International Symposium on Multimedia (ISM2011), Dana Point (California) 5-7 dicembre 2011.
- P. Bellini, P. Nesi, M. Serena, "MyStoryPlayer: experiencing multiple audiovisual content for education and training", submitted to Multimedia Tools and Applications Journal.