Engineering Multimedia Applications on the basis of Multi-Structured Descriptions of Audiovisual Contents

Marc Caillet, Cécile Roisin, Jean Carrive, François Yvon

Engineering Multimedia Applications on the basis of Multi-Structured Descriptions of Audiovisual Contents

Marc Caillet
INA
4 avenue de l'Europe
94366 Bry-sur-Marne, France
mcaillet@ina.fr

Jean Carrive
INA
4 avenue de l'Europe
94366 Bry-sur-Marne, France
jcarrive@ina.fr

Cécile Roisin
INRIA Rhône-Alpes
655 avenue de l'Europe
38334 St-Ismier, France
cecile.roisin@inrialpes.fr

François Yvon
GET/ENST CNRS/LTCI
46 rue Barrault
75013 Paris, France
yvon@enst.fr

ABSTRACT

We focus our interest on the engineering of multimedia applications whose purpose is to exploit and make best use of the audiovisual heritage by means of prospective exploration of virtual access to audiovisual documents through multi-structured descriptions of these. Multi-structured descriptions are composed of multiple descriptors that are expressed using the FDL (Feria Description Language) object language whose expressive power is emphasized. FDL notably provides a multimedia developer with operations on descriptions and their inner descriptors, as well as temporal aggregation data types. An experimental multimedia application that makes extensive use of FDL concepts and mechanisms is outlined. It explores the use of syncing between the narrative structure of the text of a play and the narrative structure of different broadcasted performances of this play, at multiple granularity levels.

Categories and Subject Descriptors

I.2.4 [**Artificial Intelligence**]: Knowledge Representation Formalisms and Methods—*Representation languages*; I.5.4 [**Artificial Intelligence**]: Applications—*Signal processing*; H.5.1 [**Information Systems**]: Multimedia Information Systems

Keywords

Automatic speech recognition, multi-structured descriptions, audiovisual, temporal constraints, multimedia.

1. INTRODUCTION

In the age of digitization, the indexation process of documents that are parts of a cultural heritage benefits from a change in the very nature of the documents to process. These documents are stored as computer files, which makes it possible to automatically extract descriptors via computer analysis tools. As a result, the number of descriptors is potentially huge. Altogether, they form a multi-structured description of the described document; some of them bring virtual access to documents through a specific viewpoint.

In this context, we focus our interest on the engineering of multimedia applications whose purpose is to exploit and make best use of the audiovisual heritage by means of prospective exploration of virtual access to the documents through descriptors of these. Multimedia applications that fall within this scope basically divide into two categories:

1. Navigating in audiovisual documents through their descriptors. A typical scenario of such an application is: (i) the user selects a segment in a temporal segmentation, the application then plays the related audiovisual segment; (ii) while playing, the application also displays current time-dependent information that is held by temporal descriptors; possibly, information that is held by other referred descriptors as well as other audiovisual content are also displayed.

2. Generating new audiovisual documents on the basis of existing audiovisual documents and their descriptors. A typical scenario of such an application is: (i) the user selects temporal segments of existing audiovisual documents by means of temporal segmentations of these documents; (ii) the application generates a new audiovisual document which may be considered as a rough-cut that is composed of the selected segments.

We adress the engineering issue of these multimedia applications through a formalism for representing descriptions of audiovisual documents, FDL (Feria Description Language), a framework, FERIA, and the JAM![1] set of experimental multimedia applications.

This article is organized as follows. Section 2 stresses the context of our work as well as related work. Section 3 elaborates on JAM! purpose, describes the corpus on which it

[1]JAM! stands for *Jouons Avec "le Misanthrope" !*, which means *let us play with "the Misanthrope"!*

SADPI '07 Montpellier, France

Table 1: The six recordings of *The Misanthrope*

Broadcasting date	Stage director	Company
09/28/1959	Jean Piat	*Comédie Française*
01/19/1968	Maurice Sarrazin	Maurice Sarrazin & al.
02/20/1971	Pierre Dux	Jean Rochefort & al.
09/14/1977	Pierre Dux	*Comédie Française*
01/06/1980	Jean-Pierre Vincent	National Theater of Strasbourg
07/23/1980	Antoine Vitez	Company of Ivry districts

is based on and defines its functionalities. It also describes the tools that are used to get a multi-structured description of each of the six *The Misanthrope* broadcasted performances. Section 4 describes the FDL language which forms the core part of the FERIA framework. It emphazises temporal constraints of descriptors together with references between multi-structured descriptions. Section 5 then shows how these mechanisms operate in JAM!. Finally, section 6 concludes and suggests some directions for future work.

2. RELATED WORK AND CONTEXT

The multimedia applications we adress are very similar to those the Advene Model considers [5], [4]; the means to reach them notably differ on how to express descriptor semantics, on mereological relations expressiveness and on reference between descriptors. Besides, one may reuse audiovisual documents and related descriptions from one application to another; the second category of applications we delineated above also repurpose both audiovisual documents and descriptions. From this perspective, our work also bears similarities to [27] that define an algebra to handle transform and reuse operations on descriptions.

In order to support our efforts, we devised the experimental FERIA framework which basically provides multimedia applications developers with a description language, FDL, and with content, document, analysis and description servers [12]. FDL is the straightforward continuation of AEDI (Audiovisual Event Description Interface) [6] to which it mainly adds: descriptor classes semantics via an object model, improved mereological relations via temporal aggregation typing, references between descriptions and inner descriptors, and improved documentary model. [35] propose AVDL (AudioVisual Description Language), which is a first draft of FDL, and point MPEG-7 (Multimedia Content Description Interface [28] and [29]) lack of formal semantics.

Description logics form a large family of knowledge representation formalisms that stand at a crossroads of both object approaches and logic [7]. They are all based on the definition of concepts and roles, that respectively correspond to classes and properties. The subsumption and instanciation operations differ from the ones of object languages such as FDL. The subsomption operation here organizes the concepts on the basis on their instances; the instanciation calculates the concepts an object is an instance of. [14] addresses the instanciation of audiovisual sequences as a constraint satisfaction problem.

OWL [30] is a widely used ontological language in the context of the semantic web for intelligent access to information. It has been designed on the basis of both description logics and the RDF language [21]. Several authors combine OWL together with MPEG-7 to overcome the latter's lack of formal semantics. For instance, [22] builds an ontology that organizes MPEG-7 description tools. [34] make use of ontologies and rules to formalize a subset of the semantic constraints of MPEG-7 DAVP profile [8].

3. JAM!

Work reported in this section is jointly carried out by ENST (Telecom Paris) and INA (National Institute for Audiovisual). The main objective JAM! is twofold. First, getting a fine-grain temporal alignment — up to the level of phonemes — between the text and the audio track of multiple performances of the same play. Second, exploring what multimedia applications we are considering could gain from this alignment. A side-objective is to validate both the usefulness of multi-structured descriptions and the way the FERIA framework deals with them.

JAM! rely on the analysis of six different broadcasted performances of a famous play by Molière, *The Misanthrope*, as well as on the score of the play. The results of these analysis are expressed as FDL descriptions of the broadcasted performances. One JAM! application has currently been devised: JAM! Dual Players, a navigation application that considers two recordings at a time and allows one to synchronically play acts and scenes of both recordings.

3.1 Corpus

We opted for Molière's *The Misanthrope* because of the large number of available recordings that were aired on French television, then recorded, archived, digitized and annotated for a documentary purpose by INA. They range from the late fifties to the early eighties (*c.f.* table 1).

The whole set of performances has been supplied in MPEG-1 format. Each of the six recordings is archived into several partial files; the supplied MPEG-1 files result from the editing of the original corresponding set of archived files. Besides, these supplied resulting files were demultiplexed and decoded in order to extract the audio. As a result, six wave files were also supplied.

Although *The Misanthrope* is almost entirely composed of rhyming alexandrine couplets, it comprises some irregularities: a small passage (Act I, Scene II) is composed of octosyllables in which one of Célimène's lover, Oronte, is declaiming a sonnet of his own; and a longer one (Act V, Last Scene), a prose passage, in which two marquis, Clitandre and Acaste, are revealing to the audience the content of Célimène's incriminating epistolary correspondence. Altogether, the text of the play totals about 1800 verses.

Additionally, we picked up the text of the play on the Gallica web site[2] and then transformed it into the Text Encoding Initiative[3] standard format.

[2] `http://gallica.bnf.fr/`

[3] `http://www.tei-c.org/P4X/index.html`

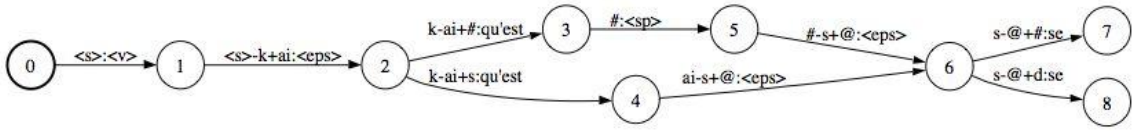

Figure 1: Part of an example of a finite-state transducer that is used during decoding. This transducer licenses two different realization of the words "qu'est-ce", with (transition 2-3-5-6) or without (transition 2-4-6) a short pause between the two words.

3.2 Analysis of broadcasted plays

JAM! relies on the results of automatic speech recognition tools that are run over each of the six recordings the corpus contains. These techniques encounter multiple issues. For instance, it is a widely acknowledged fact that various conditions are detrimental to high quality speech recognition. In the context of French classical theater performance, such conditions are background noise (footsteps, chairs, rustling paper, and so forth), musical interludes, approximations in the recitation (mistaken works, forgotten verses), expressiveness of the actors (shouts, cries, laughters, whispers, and so on), long silences between verses, words, and even, sometimes, between syllables. They all contribute to degrade the accuracy of automatic transcripts. Making use of additional text resources may dramatically improve the results [1, 17]. In the context of theater shows, the situation is even more favorable, since the text of the plays is available and contains stage directions for actors, turning the recognition problem into a much simpler alignment problem.

Besides, French classical theater, on which we are focusing, is subject to very strong constraints : they are composed of rhyming alexandrine couplets; each alexandrine is composed of exactly twelve syllables; they equally divide into two parts that are separated by a caesura which is located right between the sixth and seventh syllables. These constraints are illustrated hereafter by a short excerpt of *The Misanthrope* (in which the caesura is depicted by ||) :

Oronte
Mais ne puis-je savoir || ce que dans mon sonnet...
Alceste
Franchement, il est bon || à mettre au cabinet;[4]

Moreover, extremely detailed pronunciation prescriptions add up to these constraints. They cause the pronunciation to be quite different from what would be considered standard French, thus precluding the use of automatic speech recognition and alignment tools that are based on pronunciation dictionaries.

The alignment process then run two successive steps: (i) a metrical and phonetical analysis is performed over the text; (ii) on the basis of the results of the first step, the alignment process is performed over the audio track of each of the six recordings.

3.2.1 Metrics and phonetics

Previous studies on French classical theater texts have lead some of us to develop an automatic text annotation tool, the Metrometer [10], which implements the verse specific pronunciation rules on top of a full-fledged rule based text-to-phoneme system.

[4]Oronte: But I should like to know what there is in my sonnet to... Alceste: Candidly, you had better put it in your closet. (from the English translation supplied by `www.bibliomania.com`)

This tool has been used to produce a reference phonetic transcription for all the versified plays of the three most famous French writers of the 17th century — namely, Molière, Pierre Corneille and Jean Racine — that is a grand total of several hundred thousand verses [9]. Most of these transcriptions have been manually checked and all the deviations from the canonical alexandrine model have been recorded and/or corrected. Additional annotations, such as the syllable boundaries, the part-of-speech tags, the location of stressed syllables, etc. are also available, albeit with a much lesser degree of accuracy.

3.2.2 Temporal alignment

A specific alignment system has been designed to overcome the specificities of French classical theater play recordings. It is based on both the metrical and phonetical annotations supplied by the aforementioned Metrometer, and the Sirocco[5] speech decoding engine some of us previously developed [19] that has been enhanced in multiple ways: (i) it has been provided with a new decoding strategy that is based on static finite-state graphs (in a similar manner as, *e.g.*, [31]); (ii) the decoder is able to process unsegmented, and therefore arbitrarily long, input files. Altogether, the alignment procedure consists of the following three steps:

1. Construction of a finite-state decoding graph based on the reference phonetic transcription. To take into account the approximate respect of verse pronunciation rules by modern actors, this graph contains multiple pronunciation variants (*c.f.* figure 1).

2. Decoding through this finite-state graph: verse final segments are identified on-the-fly, yielding a phone based alignment between speech and reference pronunciation (*c.f.* table 2, in which the alignment is given at the word level, and where: (i) [sp] denotes

[5]Freely available at `http://gforge.inria.fr/sirocco`

Table 2: Temporal alignment of the partial verse "*Qu'est-ce donc ? Qu'avez-vous ?*" ("What is the matter? What ails you?") from *The Misanthrope*.

start	end	phones	word
0.01	0.03	[sil]	[sil]
0.04	0.18	/kE/	qu'est
0.19	0.21	[sp]	[sp]
0.22	0.33	/s@/	ce
0.34	0.36	[sp]	[sp]
0.37	0.52	/dO k/	donc
0.53	0.56	[sp]	[sp]
0.57	0.78	/kave/	qu'avez
0.79	0.85	/vu/	vous
0.86	1.11	[sil]	[sil]

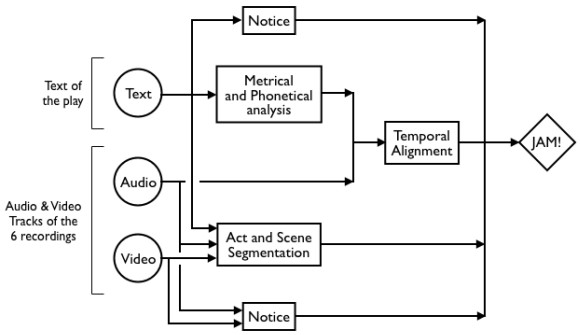

Figure 2: The analysis graph.

short pauses, (ii) [sil] denotes outer verse silences, (iii) phonemic transcriptions using the SAM Phonetic Alphabet[6]); this module uses standard preprocessing (parameterization) routines, and takes advantage of extant acoustic models (context-dependent Hidden Markov Models).

3. Various post-processing tools allow to match the time aligned transcription with the textual representation of the play at the finest possible level: each decoded phone is mapped with the related phoneme in the annotated play, allowing to match portions of the shows with portions of the play a all levels of granularity (word, verse, speaker turn, scene, and so forth).

3.2.3 Additional resources

In addition to the analysis we described above, JAM! benefits from additional resources:

- Manual temporal segmentation of each of the six recordings at both act and scene level.
- Documentary descriptive note of each of the six recordings. These notes are those that are produced by INA documentalists. They notably supply broadcasting information and the names of the television director, the stage director and the performers.
- Documentary descriptive note of the text of the play which supplies information about the characters and their role.

Figure 2 depicts the overall analysis graph that is used for JAM!. It stresses which media or which modality is analyzed, and analysis results on which some analysis tools depend on.

3.3 Applicative perspectives

Various applications that fall within the two categories we outlined in the introduction are being explored.

3.3.1 Navigation

Text alignment may allow direct access to sections of the play divided into acts and scenes. It may also allow direct access to parts of the play that feature noteworthy peculiarities, such as prose or non-alexandrine verse.

Other functionalities may fit the needs of specific audiences. Let us first consider student actors: Bringing out verses that differ from one performance to another in the way they are pronounced, bringing omitted words and omitted verses to light, underlining poor and rich rhymes as well as verse internal rhymes, and so on could be used in actor classes. In a similar fashion, assigning emotion labels to verses or lines could provide student actors with useful and lively acting examples. Moreover, full text search of lines or fragments of lines together with playing the results from each of the different recordings of a play may help student that experience difficulties with some parts of the play. Other queries would allow students to focus on more specific aspect of the play. For instance, considering the play *The Misanthrope* by Molière, a query that returns segments which contain Célimène's lines that are immediately followed by Alceste's would allow to study how Alceste reacts to what Célimène says.

From a more linguistic perspective, these recordings and related annotations constitute a very valuable resource for prosodic studies, both at macro-rhythmic level (turns, location of silences and pauses, acceleration and deceleration of the speech rate), and micro-rhythmic (location of group accents, variation of the melody, and so on).

3.3.2 Generation

The plain visualization of a play could be enriched in several ways: (i) subtitling with performers' pronunciation stressed; (ii) summarization of the performances by means of text summarization techniques performed over the play's score; (iii) synthesis of a virtual play or a virtual line on the basis of the set of the six initial performances and their description (for instance, assuming that verses are classified by performers, then by rhymes, a new play or tirade could be created by random combination of pairs of rhyming verses This technique could be improved by clustering verses according to semantic similarity. A new play could also be synthesized by picking stage actors from various recordings.)

3.3.3 Hybrid applications

The borderline between navigation and generation applications is obviously quite fuzzy. For instance, generation applications might need navigation features in order to navigate within contents that are intended to serve as a basis to the generated content.

3.4 Representation issues

The results of analysis that are performed over audiovisual contents, whether they are manual or automatic, are composed of a set of valued features of what is analyzed: for instance, face features in case of a face detection, model parameters in case of a speaker model learned over a collection of documents, temporally located segments in the case of a temporal segmentation, and so forth.

We call structured descriptor the set of valued features that result from a given analysis on a given audiovisual content. Descriptors may result from the analysis of different modalities: video track, audio track, text track, or any combination of them. Some are temporal, such as the ones that result from the temporal alignment; others are not, such as a documentary descriptive note. Some temporal descriptors bring virtual access to documents through a specific viewpoint: for instance, a speaker segmentation of a recorded play allows to navigate from one line to another of a same character. Moreover, as we work with six different recordings of *The Misanthrope*, we may navigate from one speaker

[6] `http://www.phon.ucl.ac.uk/home/sampa/index.html`

in a given recording to another speaker in another recording.

The idea here is to consider the structured descriptors that describe a given audiovisual content as a multi-structured description of this content and create references from one description to another, and references from one inner descriptor of a description to another inner descriptor in the same description.

Moreover, complex queries, such as a query that returns segments which contain Célimène's lines that are immediately followed by Alceste's (*c.f.* section 3.3.1), advocate for temporally constrained data types.

4. FDL

FDL is an object description language that forms the core part of the FERIA framework. It has been designed to handle the requirements the previous section states. Before coming to describe how FDL deals with multi-structured descriptions and complex queries, this section outlines the main points of FDL and defines which documentary entities are describable.

4.1 An object language

FDL is an object language in which descriptor classes are intensionally defined as a set of properties that characterize the descriptor. As usual in object programming languages, the instanciation operation consists in the creation of an instance of descriptor from the corresponding descriptor class. Classes and instances are uniquely identified by a urn which is a persistent and location-independent identifier for any information resource [15].

FDL comes with a tiny core set of highly general descriptor classes that notably includes (*c.f.* figure 3):

- fdl:D is a descriptor class which intension is composed of: an identifier (fdl:id); a reference towards the upper level descriptor (fdl:UpperLevel), which is useful in case of hierarchical segmentations; a list of urns of other descriptors that refer the current descriptor (fdl:Referencers); and three urns that uniquely identify an instance descriptor, its class and the documentary entity it describes (respectively fdl:D_Urn, fdl:DC_Urn and fdl:DE_Urn).

- fdl:TD is a temporal descriptor class that specializes fdl:D with the addition of a temporal location (fdl:T_Loc) which is expressed as a temporal segment with inclusive lower bound and exclusive upper bound.

- fdl:pmoTS specializes fdl:TD with the addition of a list of temporal segments (pmoSegments). The type of pmoTS's segments property states that: (i) there are at least two segments (+); (ii) a segment can be any temporal descriptor (fdl:TD); (iii) two temporally consecutive segments must comply with either precedes, meets or overlaps Allen relations ([p ∨ m ∨ o]).

- fdl:pmTS specializes fdl:pmoTS by specializing the type of its constraining temporal aggregation type by restricting the disjunction of allowed Allen relations ([p ∨ m]).

Any user-defined descriptor class needs to specialize at least one of FDL basic class using one of the following mechanisms: (i) addition of properties; (ii) specialization of the type of inherited properties; (iii) semantic specialization (the user-defined class adds no properties neither it specializes a type, only the name differs).

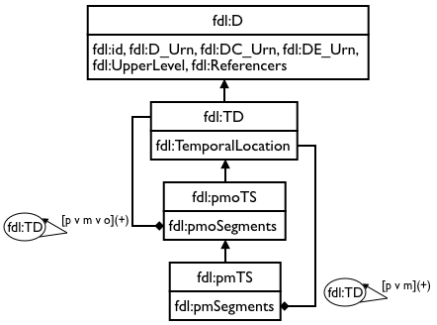

Figure 3: Excerpt of the hierarchy of FDL basic descriptor classes.

FDL relies on an abstract object metamodel that holds the descriptor classes hierarchy, and in which each descriptor class holds its instances.

4.2 Describable documentary entities

Among other formalisms and servers, FERIA provides multimedia engineers with a documentary model that defines which documentary entities are describable and how they are linked to the content they abstract.

FERIA documentary model has been elaborated: (i) on the basis of the International Federation of Library and Institutions[7] standard; (ii) taking into account specificities of digital audiovisual contents and specificities of documentary practices. The model thus divides into two abstraction levels:

- **The "media" level** that abstracts both location and possible duplicates of a digital audiovisual content which is held by a FERIA content server.

- **The "document" level** that abstracts the encoding features of the "media" level. This is especially useful when we have to describe a document which related content does not already exists. For instance, we may need to describe the evening news of a given channel at morning time, with features that are common to every evening news.

Besides, multiple describable documentary entities emerge within the "document" abstraction level: document (a homogeneous television unit that is meaningful and that can be apprehended aside from its context), excerpt (of a document), collection (of documentary entities), and track subsets.

Figure 4 pictures the path that leads from a descriptor to a content through the documentary entity the descriptor describes.

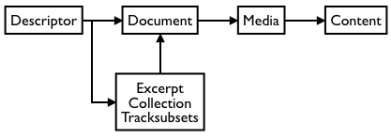

Figure 4: From a descriptor to a content.

[7] `http://www.ifla.org`

4.3 Temporal aggregation types and related classification issue

For FDL is expected to express descriptions on audiovisual contents, it has been provided with the ability to hold types that are defined as temporally constrained aggregation of descriptor classes. Given that a temporal location is defined as a temporal interval (*c.f.* section 4.1), aggregations are constrained with disjunctions of Allen relations [3] that we enriched with cardinalities and time parameters, thus notably following [33] who pointed out that Allen relations are not sufficient from the viewpoint of multimedia authoring since they do not precise timing information.

Besides, these temporal aggregation types must comply with object engineering needs. In particular, the specialization relation between descriptor classes must be defined. Which may be tricky in the case of temporal descriptor classes with a temporal aggregation typed property. To deal with this issue, FDL has been provided with a classification structure of its own. Following is a brief introduction to this structure (*c.f.* [13] for further details).

Classification structure. The classification structure is composed of three hierarchies: a descriptor class, a property class and a type hierarchies. Each of these hierarchies: (i) organizes its classes according to a specialization relation; (ii) comes with three operations, namely initialization of the hierarchy, introduction of a class and calculus of the subsumption relation. FDL classification structure together with associated operations are implemented within an expert system which purpose is to help FERIA developers to classify their own-defined classes. The descriptor class hierarchy is initialized with the introduction of FDL basic descriptor classes (*c.f.* section 4.1); property class hierarchy and type hierarchy are respectively initialized with the introduction of FDL basic property classes and FDL basic types. The subsumption calculus relies on the following basic rules:

- A descriptor class d_1 subsumes a descriptor class d_2 if and only if d_1 is a superclass of d_2, or the property classes of d_1 are superclasses of d_2's. Any change in the property classes hierarchy may imply subsumption recalculation in the descriptor class hierarchy.
- A property class p_1 subsumes a property class p_2 if and only if p_1 is a superclass of p_2, or the type of p_1 is a supertype of p_2's. Any change in the type hierarchy may imply subsumption recalculation in the property hierarchy.
- The rule that define the subsumption between two temporal aggregation types t_1 and t_2 notably states that the aggregated descriptor classes of t_1 are superclasses of the aggregated descriptor classes of t_2. Any change in the descriptor class hierarchy may imply subsumption recalculation in the type hierarchy.

JAM!-related classification examples. Let tvshow denote the temporal descriptor class that represents the logical structure of a general television show. tvshow is defined by a property that holds temporal segments which in turn is defined by a temporal aggregation type. tvshow is composed of opening credits — O — that precede closing credits — C (*c.f.* figure 5). Both O and C are specializations of fdl:TD.

Finally, let tvplay denote the temporal descriptor that represents a broadcasted play: opening credits that meets the

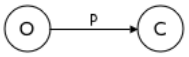

Figure 5: Type of tvshow's segments.

play itself which in turn meets closing credits; let P denote the temporal descriptor that represents the logical structure of the play: an act that precedes or meets another act four times, which sums up to exactly five acts; let A denote the temporal descriptor that represents the logical structure of an act: a scene that meets another scene at least one time; and let S denote the temporal descriptor that represents a scene (*c.f.* figure 6), S is a semantic specialization of fdl:TD.

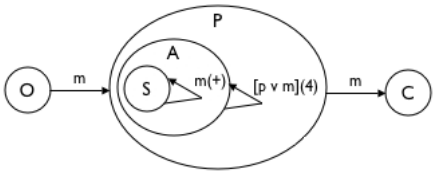

Figure 6: Temporal aggregation type of tvplay's segments.

In a first approach, every temporal aggregation type is assigned the root type — RootT — as its superclass, the related properties are assigned the root property — RootP — as their superclass, and both tvshow and tvplay are assigned fdl:TD as their superclass. Hereafter is an excerpt of an interaction between the expert system and the user at the introduction of the type of tvshow's segments property:

```
=== (T) === Type tvshowSegments_Type (#) ===
Current subsumer (*) - Other possible subsumers (+):
(*) RootT
|  ...
|  (+) pmoTS_Segments_Type
|  |  (+) pmTS_Segments_Type
|  (#) tvshowSegments_Type
```

In this example, tvshowSegments_Type is subsumed by pmTS_Segments_Type because: (i) O and C are specializations of fdl:TD; (ii) the relation p is more specific than [p ∨ m](+). In a similar way, tvplaySegments_Type is subsumed by tvshowSegments_Type because: the relation p can be inferred between O and C by transitivity of the m relation [2]. tvplaySegments_Type is subsumed by pmTS_Segments_Type because: (i) O, C, P, A and S are all specializations of fdl:TD;

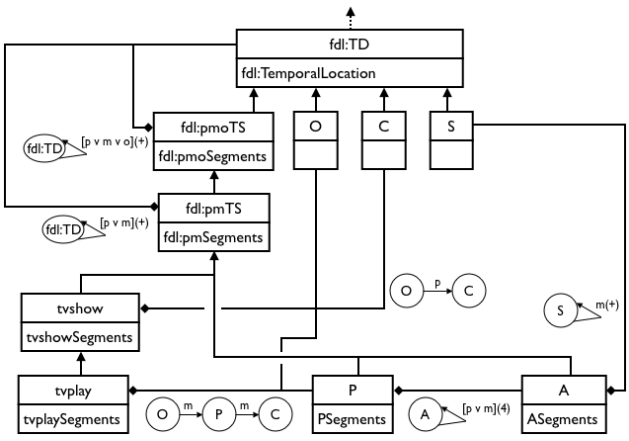

Figure 7: Excerpt of the FDL hierarchy with JAM!-related descriptor classes classified.

(ii) because all relations are more specific than [p ∨ m]; (iii) the cardinalities of tvplaySegments_Type sums up to + which is the same as the cardinality of pmTS_Segments_Type.

Depending on the user's choices, the resulting descriptor class taxonomy could be as depicted by figure 7.

It notably states that a tvshow is more specific than the pmTS FDL basic temporal segmentation, and that the tvplay is more specific than tvshow, and thus more specific than pmTS, which is very useful for the JAM! application (*c.f.* section 5).

FDL also bears similarities with the MediaObject Schema [18] that addresses interactive television documents structuring issues, as well as with the Madeus authoring environment [23] that constrains temporal scenarios with Allen relations in order to ensure its consistency at editing time.

4.4 Multi-structured descriptions

A multi-structured FDL description of a documentary entity is a set of descriptors that describe either a whole documentary entity or a track subset of it. As stated by section 4.2, a documentary entity is either a document (that abstracts an audiovisual content), an excerpt of a document, or a collection of documentary entities. The focus here is on how to efficiently use multi-structured descriptions and descriptors they are composed of, whether they are temporal or not, in a FERIA multimedia application.

The multimedia applications FERIA considers are all based on accessing audiovisual content through its description. They differ from graph-architected applications in which: (i) an arc from one node to another defines a navigation link; (ii) each node includes one or more components and links between them that manage their relative behavior. Links are statically specified by the developer at authoring time; they directly point moments in time in the audiovisual content. The Amsterdam Hypermedia Model [20], for instance, has been designed to handle such links; it served as a basis to the W3C Synchronized Multimedia Integration Language[8] timing model. Mexitl [11] is an interval temporal logic that extends Allen's with, notably, modal operators and numerical parameters. Its purpose is to represent temporal constraints between components, or subdivisions of components, in multimedia documents.

Conversely, in a FERIA application, links are dynamically specified by the descriptors that are used by the application at runtime. Links are references: (i) from one descriptor, whether it is temporal or not, to another; (ii) from one descriptor to the audiovisual content through documentary abstractions (as depicted by figure 4). To allow interactions between descriptors and between contents, any multi-structured FDL description is provided with operations on either two distinct descriptions or on the descriptors a description is composed of. Links between components, that can be specified using the Amsterdam Hypermedia Model, are undirectly created by means of descriptors.

4.4.1 Hierarchical cross-reference

This first operation is meant to ease the navigation within a hierarchical segmentation. For example, let us consider the hierarchical temporal segmentation of a recorded play at act and scene level: the first level is composed of a list of segments whose type is a temporal aggregation that constrains a play to be composed of exactly five temporally successive acts; the second level of the hierarchy is also a temporal segmentation that is composed of a list of segments whose type is a temporal aggregation that constrains an act to be composed of at least two temporally successive scenes. Thus, every act implicitly refers to its inner scenes; conversely; every scene refers to the act it is part of through the fdl:UpperLevel property which is inherited from fdl:D.

4.4.2 Temporal cross-reference

Temporal cross-reference is needed to be performed in situations where we need to know which temporal descriptors are temporally included in a given temporal interval. A first example is: while a recorded theater play is being played, we may need to know which are the current speaker, act and scene. Figure 8 shows a temporal cross-reference between the player's current time and partial temporal segmentations at speaker (the first four speaker turns) and act and scene level (the first act and its three inner scenes).

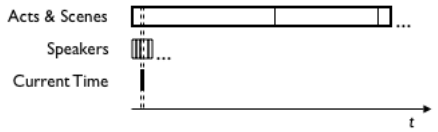

Figure 8: Temporal cross-reference at current time.

Another example is: we may need to retrieve which stage performers are playing in a given scene. Figure 9 shows a temporal cross-reference between the (partial) act and scene temporal segmentation and the (partial) temporal segmentation at speaker level.

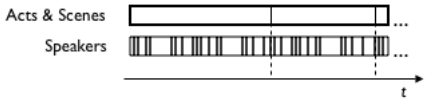

Figure 9: Temporal cross-reference between two temporal descriptors.

The temporal cross-reference operation thus performs comparisons between a given temporal location, which is expressed as an interval with inclusive lower bound and exclusive upper bound, and the temporal location of instances of one or more temporal descriptor classes.

4.4.3 Inter-descriptions cross-reference

The inter-descriptions cross-reference operation is performed in situations in which we need to sync multiple recordings on the basis of temporal segmentations. For instance, let us consider two different recordings of *The Misanthrope* and their respective act and scene segmentation, we may need to sync both of the recordings at act or scene level. Figure 10 pictures act and scene segmentations of two different recordings of *The Misanthrope* that have different act, scene and overall lengths.

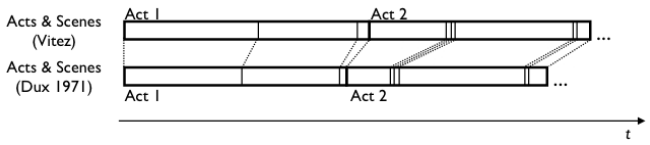

Figure 10: Syncing two different recordings at scene level.

[8]SMIL: http://www.w3.org/TR/2005/REC-SMIL2-20051213/smil-timing.html

Given that the two act and scene segmentation are instances of the same descriptor class, the inter-descriptions cross-reference operation can be performed by comparisons between the rank of temporal segmentations' inner segments.

4.4.4 Reference towards a non temporal descriptor

The third operation consists in referencing a non temporal descriptor from any other descriptor, whether it is temporal or not. For instance, this operation may be performed in order to actually retrieve the text of a scene from its temporal alignment (*c.f.* figure 11). This is a situation where a temporal descriptor refers to a non temporal one.

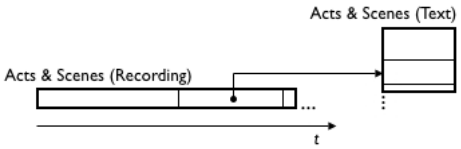

Figure 11: From temporal to non temporal.

This operation may also be performed between two non-temporal descriptors. For instance (*c.f.* figure 12), an actor that is included in a documentary descriptive note of a given play recording may refer to a character that is included in the documentary descriptive note of the text of the play, thus allowing us to retrieve which character is played by a given actor.

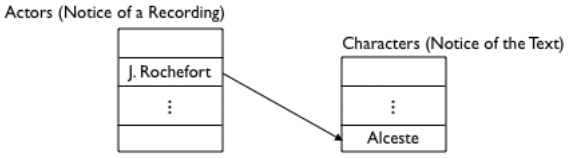

Figure 12: From temporal to non temporal.

For this operation to be held, the definition of the source descriptor class contains a property which type is a reference towards the targeted descriptor class. Each source descriptor instance then holds the urn of the targeted descriptor instance.

4.4.5 From non temporal to temporal

Lastly, we may need to play a recording starting from a given point that has been selected from the text of the play (as depicted by figure 13); this is performed thanks to a reference from a non temporal descriptor (a logical segmentation of the text of the play) to a temporal descriptor (a logical segmentation of the recorded play).

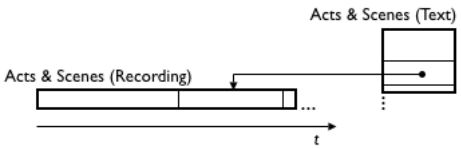

Figure 13: From non temporal to temporal.

This operation is processed by accessing the descriptor gathered by the fdl:Referencers property every descriptor inherits from fdl:D (c.f. figure 3). This property is dynamically valued while the descriptors arebeing loaded. Additionally, this operation can be performed between two non temporal descriptors.

5. FDL IN JAM!

Currently, we have developed a prototypic application that considers two recordings at a time and allows to synchronically play acts and scenes of both recordings (*c.f.* figure 14).

This application relies on temporal segmentations at speaker, act and scene levels, as well as on the documentary descriptive note of each of the recorded plays and the documentary descriptive note of the text of the play. Their FDL expression, using the XML syntax, comes from various processes:

- The text of the play has been transformed from the TEI format to an FDL descriptor that represents a logical segmentation of the text at act, scene, line and line item level. This transformation has been processed by using an XSLT stylesheet.
- The act and scene segmentation of each of the play, that notably contains a reference towards their related text, results from a manual annotation that has directly been expressed in FDL.
- The documentary descriptive note of the play also results from an FDL manual annotation.
- The documentary descriptive note of each of the recorded plays has been manually transformed into an FDL descriptor on the basis on a text file which comes from INA archive system.
- The temporal alignment process (*c.f.* section 3.2) aligns the text together with the recorded plays at various granularity level, from line level up to the phoneme level. We deduced a temporal segmentation at speaker level from the line level. As the results of this process are expressed in an XML format that is specific to the temporal alignment tool, we transformed them into FDL descriptors by using an XSLT[9] stylesheet.

Temporal segmentations are displayed by means of a rectangular control that is located just under each of the two players. User-friendly keyboard short-cuts make it easy to: (i) linearly navigate from act to act, or from scene to scene, (ii) hierarchically navigate from one act to its scenes, or from one scene to the act it is included in; (iii) zoom in and out. Figure 14 shows, at zoom factor 1, the three scenes of the first act followed by the remaining four acts. This control is designed to display only temporal segmentations of type pmTS (*c.f.* section 4.1). As a consequence, the segmentation selector only proposes segmentations of type pmTS. For we are able to infer that the act and scene temporal segmentation specializes pmTS (*c.f.* section 4.3), the segmentation selector is able to propose it.

Act and scene segmentation of both recordings semantically cross-refer to one another. This results in the ability to synchronically move from one segment to another: each action that is being made on one of these controls is immediately reflected on the other. Moreover, the text of the play is temporally aligned with the recordings at act and scenes level thanks to the reference that is held by the act and scene temporal segmentation towards their related text. The temporal cross-reference between, on the one hand, each player's current time, and on the other hand, the respective temporal segmentation at speaker level and at act and scene level

[9] `http://www.w3.org/TR/xslt`

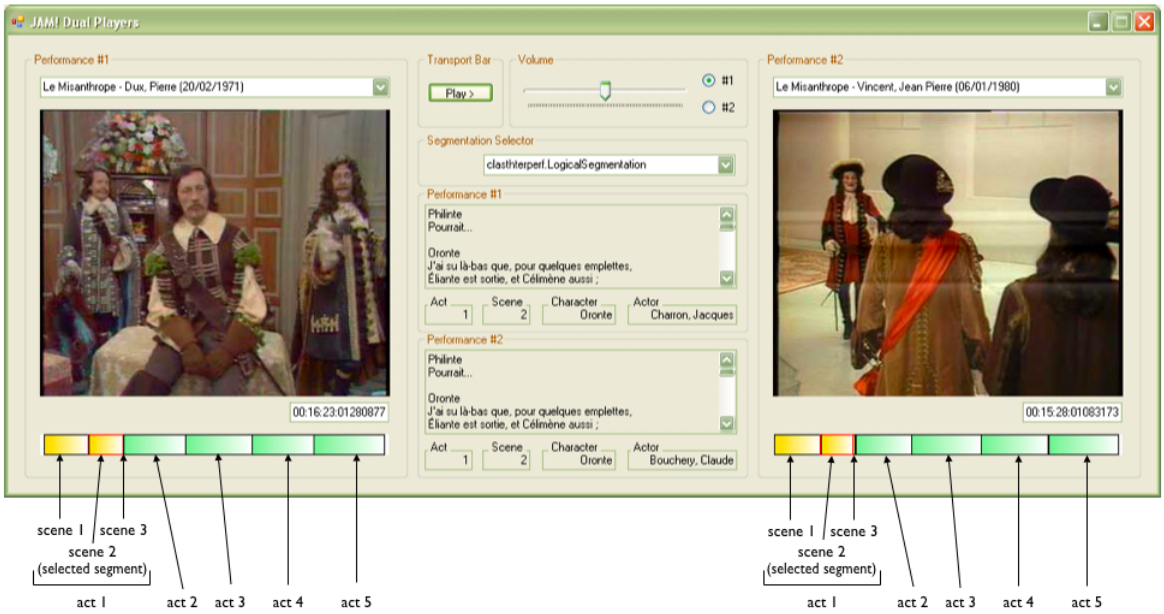

Figure 14: An example of a JAM! prototypic application.

makes it possible to know, at any moment, which actor is speaking and which are the current act and scene. Finally, a reference from the actor in the documentary descriptive note of a recorded play towards the related character in the documentary descriptive note of the text allows to display the name of the character that is being played by the speaking actor.

Just like JAM! highly benefits from *The Misanthrope* score, [26] propose enhanced video-based applications relying on additional resources. Experiences with films and their scripts have also been performed like in [32] where the focus is on the authoring of the alignment. Automatic lip sync has been being studied for a while. [25] survey solutions based on the estimation of face model parameters; one possible direction takes a preliminary step during which syncing between the text and the audio signal is processed, as it is the case in our experiment. [24] apply the aformentionned Mexitl formalism to the lip-sync problem.

6. CONCLUSION AND FUTURE WORK

This paper describes the experimental JAM! multimedia application that explores multimedia functionalities that could be engineered on the basis of broadcasted classical theater and fine-grain alignment of text and audio tracks. It shows the relevance of our approach that considers: (i) multi-structured descriptions of audiovisual documents and operations on them that allow temporal, non temporal and semantic cross-reference; (ii) temporal aggregation types that constrain descriptor classes with Allen relations and cardinalities.

Future work ranges from JAM! to analysis tools. The very next step will consist in the implementation of two additional functionalities: navigation from one segment to another at a finer grain, up to the phoneme level, and generation of a new play on the basis of actors that will be picked up in distinct existing recorded plays. Such a new play is to be considered as a rough-cut which is likely to be enriched. We plan to explore this enrichment by using the LimSee3 authoring model [16]. Descriptor instances are planned to be introduced in FDL classification structure in order to perform complex queries answering, such as a query for segments which contain Célimène's lines that are immediately followed by Alceste's. Theoretical issues are currently under study.

Computation of intonational markers (such as duration, energy, fundamental frequency) is planned to performed on the basis of the temporal alignment. These markers will then allow us to measure phenomena that are related to the respect of the metrical rhythm: realization of optional mute-E, realization of optional liaison consonants, and vocalic realizations of glides (respect of the diaresis). We are also interested in measuring the acoustical correlates of other metrical events, notably the intonational marking of the caesura and of the rhyme.

7. AKNOWLEDGMENTS

Part of this work is supported by the European Commission under contract FP-020726, Knowledge Space of semantic inference for automatic annotation and retrieval of multimedia content (K-Space). The authors would also like to thank Jun-Yi Zhao for the excellent work on XSLT transformations he did during an internship.

8. REFERENCES

[1] A. Allauzen and J.-L. Gauvain. Adaptation automatique du modèle de langage d'un système de transcription de journaux parlés. *Traitement Automatique du Langage*, 44(1), 2003.

[2] J. F. Allen. Maintaining knowledge about temporal intervals. *Communications of the ACM*, 26(11):832 – 843, 1983.

[3] J. F. Allen. A general model of action and time. *Artificial Intelligence*, 23(2), 1984.

[4] O. Aubert, P.-A. Champin, and Y. Prié. Integration of semantic web technology in an annotation-based

hypervideo system. In *Workshop on Semantic Web Annotations for Multimedia (SWAMM'06)*, Edinburgh, UK, 2006.

[5] O. Aubert and Y. Prié. Advene: active reading through hypervideo. In *16th ACM Conf. on Hypertext and Hypermedia*, pages 235–244, Salzburg, Austria, 2005.

[6] G. Auffret, J. Carrive, O. Chevet, and T. Dechilly. Audiovisual-based hypermedia authoring: using structured rerpresentations for efficient access to av documents. In *ACM Hypertext '99*, Darmstadt, Germany, 1999.

[7] F. Baader, D. Calvanese, D. McGuiness, D. Nardi, and P. Patel-Schneider, editors. *The Description Logic Handbook*. Cambridge University Press, 2002.

[8] W. Bailer and P. Schallauer. The detailed audiovisual profile: Enabling interoperability between mpeg-7 based systems. In *Proceedings of 12th Multimedia Modellinig Conference*, pages 217–224, Beijing, China, 2006.

[9] V. Beaudoin. *Mètres et Rythmes du Vers Classique. Corneille, Racine.* Champion, 2002.

[10] V. Beaudoin and F. Yvon. The metrometer: a tool for analysing french verse. *Literary and Linguistic Computing*, 11(1), 1996.

[11] H. Bowman, H. Cameron, P. King, and S. Thompson. Mexitl: Multimedia in executable interval temporal logic. Technical report, Computing Laboratory, University of Kent, 1997.

[12] V. Brunie, J. Carrive, and L. Vinet. Ingénierie des documents audiovisuels : le projet feria. *TSI*, 25(4):469 – 496, Mai 2006.

[13] M. Caillet. Un système expert d'aide à la classification taxonomique de classes de descripteurs. In *Ingénierie des Connaissances*, Grenoble, France, July 2007.

[14] J. Carrive, F. Pachet, and R. Ronfard. Clavis - a temporal reasoning system for classification of audiovisual sequences. In *Proceedings of RIAO*, Paris, France, 2000.

[15] L. Daigle, D. van Gulik, R. Iannella, and P. Faltstrom. *Uniform Resource Names (URN) Namespace Definition Mechanisms*, October 2002.

[16] R. Deltour and C. Roisin. The limsee3 multimedia authoring model. In *DocEng '06*, Amsterdam, The Netherlands, 2006.

[17] J. Glass, T. Hazen, S. Cyphers, I. Malioutov, and R. Barzilay. Progress in spoken lecture processing. In *Int. Conf. on Spoken Language Processing*, Pittsburgh, U.S.A., 2006.

[18] R. Goularte, E. dos Santos Moreira, and M. da Graça C. Pimentel. Structuring interactive tv documents. In *DocEng '03*, Grenoble, France, 2003.

[19] G. Gravier, F. Yvon, B. Jacob, and F. Bimbot. Sirocco, un système ouvert de reconnaissance de la parole. In *XXIVe Journées d'Études sur la Parole (JEP'02)*, Nancy, France, 2002.

[20] L. Hardman, D. C. A. Bulterman, and G. van Rossum. The amsterdam hypermedia model: adding time and context to the dexter model. *Communications if the ACM*, 37(2):50 – 62, 1994.

[21] I. Horrocks, P. F. Patel-Schneider, and F. van Harmelen. From shiq and rdf to owl: The making of a web ontology language. *Journal of Web Semantics*, 1(1):7–26, 2003.

[22] J. Hunter. Adding multimedia to the semantic web - building an mpeg-7 ontology. In *1st Int. Semantic Web Working Symposium SWWS'01*, Stanford, California, USA, 2001.

[23] M. Jourdan, N. Layaïda, C. Roisin, L. Sabry-Ismaïl, and L. Tardif. Madeus, an authoring environment for interactive multimedia documents. In *ACM Multimedia '98*, Bristol, UK, 1998.

[24] P. King, H. Cameron, H. Bowman, and S. Thompson. Synchronization in multimedia documents. *LNCS*, 1998.

[25] J. Lewis. Automated lip-sync: Backgrounds and techniques. *Visualization and Computer Animation*, 2, 1991.

[26] K.-Y. Liu and H.-Y. Chen. Exploring media correlation and synchronization for navigated hypermedia documents. In *13th annual ACM international conference on Multimedia*, Singapore, 2005.

[27] C. L. Madhwacharyula, M. Davis, P. Mulhem, and M. S. Kankanhalli. Metadata handling: A video perspective. *ACM Transactions on Multimedia Computing, Communications and Applications*, 2(4), 2006.

[28] J. M. Martínez, R. Koenen, and F. Pereira. Mpeg-7: The generic multimedia content description standard, part 1. *IEEE Multimedia*, 9(2):78–87, Avril-Juin 2002.

[29] J. M. Martínez, R. Koenen, and F. Pereira. Mpeg-7: The generic multimedia content description standard, part 2. *IEEE Multimedia*, 9(3):83–93, Juillet-Septembre 2002.

[30] D. L. McGuinness and F. van Harmelen. Owl web ontology language overview, http://www.w3.org/tr/owl-features/, 2004.

[31] M. Mohri, F. C. N. Pereira, and M. Riley. Weighted finite-state transducers in speech recognition. *Computer, Speech and Language*, 16(1):69–88, 2002.

[32] R. Ronfard and T. T. Thuong. A framework for aligning and indexing movies with their script. In *IEEE Int. Conf. on Multimedia and Expo*, Baltimore, Maryland, 2003.

[33] T. K. Shih, L.-J. Hwang, and J.-Y. Tsai. Formal model of temporal properties underlying multimedia presentations. In *Multimedia Modeling*, 1996.

[34] R. Troncy, W. Bailer, M. Hausenblas, and R. Schlatte. Enabling multimedia metadata interoperability by defining formal semantics of mpeg-7 profiles. In *1st Int. Conf. on Semantic and Digital Media Technologies (SAMT'06)*, pages 41–55, Athens, Greece, December 2006.

[35] R. Troncy, J. Carrive, S. Lalande, and J.-P. Poli. A motivating scenario for designing an extensible audio-visual description language. In *CORIMEDIA'04*, Sherbrooke, Canada, Octobre 2004.