HAL open science

Un système expert d'aide à la classification taxonomique de classes de descripteurs

Marc Caillet

To cite this version:

Marc Caillet. Un système expert d'aide à la classification taxonomique de classes de descripteurs. 18es Journées Francophones d'Ingénierie des Connaissances, Jul 2007, Grenoble, France. not specified. hal-00509863

HAL Id: hal-00509863

https://hal.science/hal-00509863v1

Submitted on 16 Aug 2010

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

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

Un système expert d'aide à la classification taxonomique de classes de descripteurs

Marc Caillet[1,2]

[1] Institut National de l'Audiovisuel,
4 avenue de l'Europe, 94366 Bry-sur-Marne Cedex, France
mcaillet@ina.fr

[2] INRIA Rhône-Alpes - Projet WAM
655 avenue de l'Europe, Montbonnot, 38334 St-Ismier Cedex, France

Résumé : Dans le cadre de la valorisation du patrimoine audiovisuel, nous nous intéressons à l'ingénierie d'applications multimédias qui reposent sur des descripteurs de documents audiovisuels. Ces descripteurs sont exprimés dans le langage de description objet FDL qui permet, notamment, l'expression de types de données complexes, définis sous la forme d'une aggrégation de classes de descripteurs, temporellement contrainte par des relations de Allen enrichies de cardinalités et de paramètres temporels. Comment alors, dans le cadre d'un développement objet, définir le lien de spécialisation de classes de descripteurs dont la définition contient de telles aggrégations ? Cet article présente la structure de classification de FDL, ainsi que son opérationnalisation au sein d'un système expert d'aide à la classification de classes de descripteurs audiovisuels.

Mots-clés : classes de descripteurs, documents audiovisuels, classification, subsomption, aggrégation temporelle, système expert.

1 Introduction

Avec le passage à l'ère du numérique, le processus d'indexation des documents audiovisuels connaît une profonde évolution : le document, sauvegardé sous forme de fichier informatique, rend possible l'extraction automatique de descripteurs tels que, par exemple, une détection d'applaudissements, une segmentation en locuteurs ou encore une transcription de la parole... Cette automatisation vient considérablement enrichir les possibilités offertes par l'annotation manuelle qui, dans le cadre de la gestion du patrimoine audiovisuel, consiste en la création de notices documentaires. Du fait même de la nature temporelle des documents audiovisuels, nombre de descripteurs sont localisés temporellement, cette localisation s'exprimant sous la forme d'un intervalle temporel. Certains descripteurs temporels permettent d'accéder virtuellement au document selon un point de vue particulier : une segmentation en locuteurs d'un enregistrement d'une représentation télévisée du *Misanthrope* de Molière permet, par exemple, de naviguer à l'intérieur du document de réplique d'Alceste en réplique d'Alceste.

Dans ce contexte, nous nous intéressons à l'ingénierie d'applications multimédias dont le but est de valoriser le patrimoine audiovisuel, en explorant les possibilités offertes en termes d'accès virtuel aux documents, à travers des descripteurs de ces documents. Ces applications multimédias se divisent en deux catégories : les applications de navigation dans des contenus audiovisuels à travers leur description, et les applications de génération de nouveaux contenus audiovisuels et de leur description à partir de contenus et descriptions existants (Caillet *et al.*, 2007).

Le langage de description FDL (Feria Description Language) constitue le cœur du framework FERIA (Brunie *et al.*, 2006) qui est notamment constitué de formalismes et serveurs pour la représentation et la gestion de contenus audiovisuels, entités documentaires, descriptions en vue de leur mise en œuvre au sein des applications multimédias que nous envisageons. Nous l'avons conçu dans la continuité directe des travaux qui ont abouti au langage AEDI (Audiovisual Event Description Interface, Auffret *et al.* (1999)). FDL y ajoute notamment : l'expression de la sémantique des descripteurs audiovisuels à travers leur organisation taxonomique, ainsi que l'expression de relations méréologiques temporellement contraintes. Le premier point répond à des besoins de réutilisation des descripteurs, de factorisation de comportements et de recherche d'information ; le second permet une exploitation fine de la nature temporelle des contenus audiovisuels.

Nous nous concentrons plus spécifiquement, dans cet article, sur les problèmes soulevés par l'introduction de types de données définis par une aggrégation temporellement contrainte de classes de descripteurs. Une telle aggrégation est contrainte par des relations de Allen (Allen, 1984), enrichies de cardinalités et de paramètres temporels. La structure logique d'une pièce de théâtre télévisée peut ainsi être exprimée comme un générique de début suivi de la pièce, elle-même suivie du générique de fin ; la pièce comme une suite de cinq actes successifs ; un acte comme au moins deux scènes successives. Ces aggrégations temporelles offrent une grande expressivité qui, en contre-partie, rend difficile la définition du lien de spécialisation de classes de descripteurs qui en font usage. C'est cette difficulté qui motive notre contribution qui consiste en la formalisation d'une structure de classification propre à FDL, ainsi qu'en son opérationnalisation au sein d'un système expert.

Cet article est organisé comme suit : la section 2 retrace les influences de FDL et le confronte à d'autres formalismes ; la section 3 présente le langage de description de contenus audiovisuels FDL, ainsi que la structure de classification qui organise ces descripteurs ; la section 4 propose un système expert d'aide à la classification de classe de descripteurs ; enfin, la section 5 conclut et ouvre sur des perspectives d'évolution.

2 Travaux connexes

La problématique abordée dans cet article est proche de celle, plus générale, de (Huchard, 2003) qui considère la classification de classes dans le cadre de l'ingénierie logicielle objet. À l'instar de cette auteure, l'idée de départ de nos travaux est de définir les classes de descripteurs par un ensemble de propriétés, puis de produire une meilleure classification sur la base de cette définition. (Troncy *et al.*, 2004) soulignent le manque de sémantique formelle de MPEG-7 (Multimedia Content Description In-

terface, Martínez *et al.* (2002a,b)) et proposent AVDL (AudioVisual Description Language) qui constitue une première ébauche de FDL. MPEG-7 ne permet pas non plus l'expression objet de descripteurs (Euzenat *et al.*, 2003) et, partant, rend impossible leur organisation taxonomique (Caillet *et al.*, 2005).

Les logiques de description constituent une grande famille de formalismes de représentation des connaissances qui se situent à la croisée des approches à objets et de la logique (Baader *et al.*, 2002; Napoli, 1997). Elles sont basées sur la définition de concepts, correspondants aux classes, et de rôles, correspondants aux propriétés. Dans ces formalismes, l'instanciation est l'opération qui détermine les concepts dont un objet est une instance, à la différence des langages à objets, tel FDL, pour lesquels elle est l'opération qui crée une instance à partir d'une classe. (Carrive *et al.*, 2000) abordent l'instanciation de séquences audiovisuelles, au sens des logiques de description, comme problème de résolution de contraintes.

OWL 1.0 (McGuinness & van Harmelen, 2004) est le dernier représentant d'une longue lignée de langages d'ontologies. Il joue un rôle considérable dans le contexte du web sémantique pour l'accès intelligent à l'information. Ce langage a été créé sur la base de logiques de description et du langage RDF (Horrocks *et al.*, 2003). Il se décompose en trois sous-langages : OWL Lite, OWL DL et OWL FULL, les deux premiers pouvant être vus comme des logiques de description. OWL 1.1 (Bechhofer *et al.*, 2006), qui n'a pas le statut de recommandation W3C, est une extension de OWL DL basée sur la logique de descriptions $\mathcal{SROIQ}$ (Horrocks *et al.*, 2006). OWL 1.0 et OWL 1.1 sont tous deux supportés par les raisonneurs Pellet[1] et FaCT++[2].

Plusieurs auteurs combinent OWL à MPEG-7 pour pallier au manque de sémantique formelle de ce dernier. (Hunter, 2001), par exemple, construit une ontologie qui organise les outils de description de MPEG-7. (Troncy *et al.*, 2006) s'attachent, quant à eux, à formaliser une partie des contraintes sémantiques, exprimées en langage naturel, du profile DAVP – *The Detailed AudioVisual Profile* (Bailer & Schallauer, 2006) – de MPEG-7 à travers ontologies et règles.

3 Structure de classification de FDL

La structure de classification de FDL est constituée de trois hiérarchies : une taxonomie de classes de descripteurs, une hiérarchie de classes de propriétés et une hiérarchie de types. Une hiérarchie ordonne un ensemble fini de classes $\mathcal{C}$ par une relation d'ordre partiel strict, notée $\prec$ (et $\preceq$, la relation d'ordre partiel associée), dite de spécialisation. Comme il est d'usage dans les langages à objets, nous désignerons par sous-classes de $c \in \mathcal{C}$ les classes qui spécialisent c, et par super-classes de c, les classes qui sont spécialisées par c. Le cadre de développement objet dans lequel nous nous situons préconise que nous nous limitions à l'héritage simple ; toute classe n'a ainsi qu'une et une seule super-classe directe.

Soient alors $\mathcal{D}$, l'ensemble des classes de descripteurs ; $\mathcal{P}$, l'ensemble des classes de propriétés ; et $\mathcal{T}$ l'ensemble des types, un type représentant une classe de données :

[1] http ://pellet.owldl.com/

[2] http ://owl.man.ac.uk/factplusplus/

Définition 1
(Super-classe directe). La fonction $sup^* : \mathcal{C} \rightarrow \mathcal{C}$ *associe à toute classe* $c_1 \in \mathcal{C}$ *sa super-classe directe :* $\forall c_1 \in \mathcal{C},\ sup^*(c_1) = c_2 \mid c_1 \prec c_2,\ \nexists c_3 \in \mathcal{C} : c_1 \prec c_3 \prec c_2$.
(Sous-classes directes). La fonction $sub^* : \mathcal{C} \rightarrow \mathcal{C}$ *associe à toute classe* $c_1 \in \mathcal{C}$ *ses sous-classes directes :* $\forall c_1 \in \mathcal{C},\ sub^*(c_1) = \{c_2 \mid c_2 \prec c_1,\ \nexists c_3 \in \mathcal{C} : c_2 \prec c_3 \prec c_1\}$.

Définition 2
(Intension d'une classe de descripteurs). La fonction $int : \mathcal{D} \rightarrow 2^{\mathcal{P}}$ *associe à toute classe de descripteurs* $d \in \mathcal{D}$ *son intension. D'autre part, la fonction* $int^* : \mathcal{D} \rightarrow 2^{\mathcal{P}}$ *associe à toute classe de descripteurs* $d \in \mathcal{D}$ *son intension propre. L'intension de* d *peut alors être calculée récursivement à partir de son intension propre et de sa super-classe directe :* $int(d) = int(sup^*(d)) \cup int^*(d)$.

Chacune de ces hiérarchies est munie d'opérations d'initialisation, d'introduction et de calcul de la relation d'ordre partiel strict, notée $\lhd$ (et $\unlhd$, la relation d'ordre partiel associée), dite de subsomption. Les algorithmes de subsomption à l'œuvre dans les logiques de description se rangent en deux grandes familles : les algorithmes de normalisation/comparaison, et les algorithmes de tableaux sémantiques (Napoli, 1997). L'algorithme de subsomption de FDL opère par comparaison ; il partage ainsi des similarités avec les algorithmes de normalisation/comparaison. Le corps de cette section définit les objets classés et décrit les opérations d'initialisation, d'introduction et de subsomption.

3.1 Classes de descripteurs, classes de propriétés et types.

Une classe de descripteur $\mathrm{d} \in \mathcal{D}$ est définie, en FDL, par une intension constituée de classes de propriétés. Une classe de propriété $\mathrm{p} \in \mathcal{P}$ est définie par un couple attribut-co-domaine, noté $\langle \mathrm{a}_p, \mathcal{C}o\mathcal{D}om_p \rangle$; le co-domaine étant le type de la valeur que peut prendre une instance de la classe de propriété, $\mathcal{C}o\mathcal{D}om \in \mathcal{T}$.

Comme annoncé en introduction, nous portons notre attention sur les types de données définis par une aggrégation temporelle. Les contraintes sont exprimées par des relations de Allen enrichies de cardinalités et, pour certaines d'entre elles, de paramètres temporels, suivant ainsi (Shih *et al.*, 1996) qui relèvent l'insuffisance des relations de Allen seules pour l'expression de contraintes temporelles dans le cadre de la réalisation d'applications multimédias. Dans la suite, nous traitons en détail les cardinalités, les paramètres temporels ne posant pas de problème particulier.

Soit $\mathcal{X} \subset \mathcal{T}$, le sous-ensemble de ces types. Tout type $\mathrm{x} \in \mathcal{X}$ est défini par un graphe $G(\mathrm{N}_x, \mathrm{E}_x), \forall \mathrm{n}_x \in \mathrm{N}_x, \mathrm{n}_x \in \mathcal{D}$ ou $\mathrm{n}_x \in \mathcal{X}$. Tout arc de $\mathrm{e}_x \in \mathrm{E}_x$ est constitué d'un triplet $\langle \mathrm{Src}, \mathrm{Trgt}, \mathrm{R} \rangle$, où Src et Trgt sont respectivement la source et la cible de l'arc — Src (*resp.* Trgt) $\in \mathcal{D}$, ou Src (*resp.* Trgt) $\in \mathcal{X}$ — et R est la relation qui les relie. Pour définir la relation R, il convient de distinguer deux cas :

1. si $\mathrm{Src} = \mathrm{Trgt}$, alors R peut être définie soit par un couple $\langle \mathrm{ARs}, \mathrm{Card} \rangle$, où ARs est une disjonction de relations de Allen (soit $\mathcal{AR}$ l'ensemble des relations de Allen, $ARs \in 2^{\mathcal{AR}}$), et Card est une cardinalité telle que $\mathrm{Card} \in \mathcal{C}ard = \{*, +, +_n, ?, (\mathrm{n}), (\mathrm{n}, \mathrm{m}) \mid \mathrm{n} \in \mathbb{N}^*, \mathrm{m} \in \mathbb{N}^*\}$, où $+_n$ signifie au moins n occurrences ; soit seulement par ARs, auquel cas $\mathrm{Card} = (1)$ par défaut.
2. si $\mathrm{Src} \neq \mathrm{Trgt}$, alors R est nécessairement définie par une disjonction de relations de Allen, et $\mathrm{Card} = (1)$.

La figure 1 illustre le type de données correspondant à l'exemple de la structure logique d'une pièce de théâtre — tvplay — donné en introduction. P est constitué d'actes — A — reliés deux à deux par la relation [precedes $\vee$ meets] et ce quatre fois, ce qui donne un total de cinq actes. A est constitué de scènes — S — reliées deux à deux par la relation meets(+), ce qui donne un total d'au moins deux scènes. S est une partie de A ; S et A sont des parties de P ; enfin, S, A, P, O et C sont des parties de tvplay.

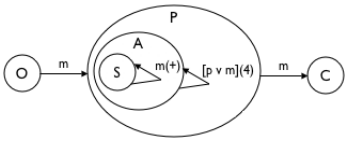

FIG. 1 – Structure logique d'une pièce de théâtre exprimée par une aggrégation temporelle.

3.2 Initialisation

La taxonomie de classes de descripteurs est initialisée par l'introduction des classes de base du langage FDL. Elles sont au nombre de cinq auxquelles s'ajoute la racine :

- RootD, racine de la taxonomie : $int^*(\text{RootD}) = \emptyset$.
- D, classe de descripteur atemporel de base, spécialise RootD, définie par trois urns permettant d'identifier de façon unique respectivement le descripteur, sa classe, l'unité documentaire décrite : $int^*(\text{D}) = \{\langle \text{D_Urn}, \text{Urn}\rangle, \langle \text{DC_Urn}, \text{Urn}\rangle, \langle \text{DE_Urn}, \text{Urn}\rangle\}$, $\text{D} \prec \text{RootD}$.
- TD, classe de descripteur temporel de base, spécialise D, définie par une localisation temporelle exprimée sous la forme d'un intervalle à borne inférieure fermée et borne supérieure ouverte : $int^*(\text{TD}) = \{\langle \text{TemporalLocation}, \text{TLoc}\rangle\}$, $\text{TD} \prec \text{D}$.
- TS, classe de segmentation temporelle de base, spécialise TD, sans intention propre : $int^*(\text{TS}) = \emptyset$, $\text{TS} \prec \text{TD}$.
- pmoTS, classe de segmentation temporelle, spécialise TS, définie par un ensemble de segments dont le type impose qu'ils soient au moins deux et reliés par la disjonction de relations [precedes $\vee$ meets $\vee$ overlaps] (*c.f.* figure 2) : $int^*(\text{pmoTS}) = \{\langle \text{Segments}, \text{pmoTS_Segments_Type}\rangle\}$, $\text{pmoTS} \prec \text{TS}$.
- pmTS, classe de segmentation temporelle, spécialise pmoTS, définie par restriction du co-domaine de la classe de propriété Segments (*c.f.* figure 2) : $int^*(\text{pmoTS}) = \{\langle \text{Segments}, \text{pmTS_Segments_Type}\rangle\}$, $\text{pmTS} \prec \text{pmoTS}$.

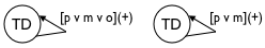

FIG. 2 – Type des segments de pmoTS et pmTS.

Soit $\mathcal{B}_\mathcal{D} = \{\text{D}, \text{TD}, \text{TS}, \text{pmoTS}, \text{pmTS}\}$, l'ensemble des classes de descripteurs de base. La hiérarchie des classes de propriétés est initialisée par l'introduction des classes de propriétés de base du langage FDL : ce sont les propriétés de l'intention propre respective de chacune des classes de descripteur de l'ensemble $\mathcal{B}_\mathcal{D}$. Il en est de même pour la hiérarchie de types, initialisée par l'introduction des co-domaines des propriétés de base auxquelles sont adjoints les types de données simples usuels (string, integer, long, float, double, boolean).

3.3 Introduction

Une classe de descripteur d est introduite : (i) en spécifiant explicitement sa super-classe directe qui est nécessairement une classe de base ou une sous-classe d'une classe de base ; et (ii) en spécifiant son nom et son intension propre. Une classe de propriété p est introduite : en spécifiant explicitement sa super-classe directe, et en spécifiant un nom utilisé comme attribut et un co-domaine. Enfin, un type t est introduit, soit explicitement en spécifiant super-classe directe, nom et définition ; soit implicitement avec sa définition donnée en valeur du co-domaine d'une propriété.

3.4 Subsomption

Une classe de descripteur d_1 subsume une classe de descripteur d_2 si et seulement si l'une des règles suivantes est vérifiée :

- **(D1)** d_1 est une super-classe de d_2 : $\mathrm{d}_2 \prec \mathrm{d}_1 \Rightarrow \mathrm{d}_2 \lhd \mathrm{d}_1$.
- **(D2)** l'intension de d_1 subsume l'intension de d_2. Soit la fonction injective f : $int(\mathrm{d}_1) \rightarrow int(\mathrm{d}_2)$ qui à toute propriété $\mathrm{p}_1 \in int(\mathrm{d}_1)$ associe une propriété $\mathrm{p}_2 \in int(\mathrm{d}_2)$ telle que $\mathrm{p}_1 \preceq \mathrm{p}_2$: $(\forall \mathrm{p}_1 \in int(\mathrm{d}_1),\ f(\mathrm{p}_1) \neq \emptyset) \Rightarrow int(\mathrm{d}_2) \lhd int(\mathrm{d}_1) \Rightarrow \mathrm{d}_2 \lhd \mathrm{d}_1$.

Une classe de propriété p_1 subsume une classe de propriété p_2 si et seulement si l'une des règles suivantes est vérifiée :

- **(P1)** p_1 est une super-classe de p_2 : $\mathrm{p}_2 \prec \mathrm{p}_1 \Rightarrow \mathrm{p}_2 \lhd \mathrm{p}_1$.
- **(P2)** le type de $\mathrm{p}_1 := \langle \mathrm{a}_1, \mathrm{t}_1 \rangle$ est une super-classe du type de $\mathrm{p}_2 := \langle \mathrm{a}_2, \mathrm{t}_2 \rangle$: $t_2 \prec \mathrm{t}_1 \Rightarrow \mathrm{p}_2 \lhd \mathrm{p}_1$.

Un type défini par une aggrégation temporelle t_1 subsume un type t_2 lui aussi défini par une aggrégation temporelle si et seulement si la règle suivante est vérifiée : **(T1)** t_1 et t_2 sont respectivement définis par les aggrégations temporelles $G(\mathrm{N}_1, \mathrm{E}_1)$ et $G(\mathrm{N}_2, \mathrm{E}_2)$, et l'ensemble d'arcs E_1 subsume l'ensemble d'arcs E_2 : Le graphe de la structure n'est pas nécessairement complet, et il peut être nécessaire d'inférer des relations implicites. Soit alors la fonction i qui, à tout couple $(\mathrm{e} := \langle \mathrm{Src}, \mathrm{Trgt}, \mathrm{R} \rangle, \mathrm{G}(\mathrm{N}, \mathrm{E}))$ associe un ensemble constitué d'arcs tels que $\forall \mathrm{n}_1 \in \mathrm{N}, \forall \mathrm{n}_2 \in \mathrm{N} : \mathrm{n}_1 \neq \mathrm{n}_2, \mathrm{n}_1 \preceq \mathrm{Src}, \mathrm{n}_2 \preceq \mathrm{Trgt}$ et il existe un chemin dans G(N,E) de n_1 à n_2, et dont la relation est calculée par transitivité des relations qui se trouvent sur le chemin (Allen, 1983).
Soit la fonction injective $h : \mathrm{E}_1 \rightarrow \mathrm{E}_2$ qui à tout arc $\mathrm{e}_1 \in \mathrm{E}_1$ associe un arc $\mathrm{e}_2 \in \mathrm{E}_2$ tel que $\mathrm{e}_2 \preceq \mathrm{e}_1$: $(\forall \mathrm{e}_1 \in \mathrm{E}_1,\ h(\mathrm{e}_1) \neq \emptyset$ ou $(\forall \mathrm{e}_1\prime, h(\mathrm{e}_1\prime) = \emptyset$ et $\exists\, \mathrm{e}_2\prime \in i(\mathrm{e}_1\prime, G(\mathrm{N}_2, \mathrm{E}_2)), \mathrm{e}_2\prime \lhd \mathrm{e}_1\prime)) \Rightarrow \mathrm{E}_2 \lhd \mathrm{E}_1 \Rightarrow \mathrm{t}_2 \lhd \mathrm{t}_1$.

Le calcul de la relation de subsomption d'arcs d'aggragations temporelles doit tenir compte de l'étendue des relations exprimables (disjonction de relations de Allen et cardinalités). Il nécessite l'introduction d'opérations sur ces structures.

Définition 3
(Opérations sur les structures temporelles). Soit $d : \mathcal{X} \mapsto 2^{\mathcal{D}}$ la fonction qui associe à toute aggrégation temporelle l'ensemble des classes de descripteurs qu'elle contraint ; soit $a : \mathcal{X} \mapsto 2^{2^{\mathcal{AR}}}$ la fonction qui associe à toute aggrégation temporelle l'ensemble des relations de Allen par lesquelles elle contraint les classes de descripteurs ; et soit $card : \mathcal{X} \mapsto \mathcal{C}ard$ la fonction qui associe à toute aggrégation temporelle sa cardinalité.

Un arc $\mathrm{e}_1 := \langle \mathrm{Src}_1, \mathrm{Trgt}_1, \langle \mathrm{ARs}_1, \mathrm{Card}_1 \rangle \rangle$ subsume un arc $\mathrm{e}_2 := \langle \mathrm{Src}_2, \mathrm{Trgt}_2, \langle \mathrm{ARs}_2, \mathrm{Card}_2 \rangle \rangle$ si et seulement si l'une des règles suivantes est vérifiée :

- Les nœuds de e_1 et de e_2 sont des classes de descripteurs, la source (*resp.* la cible) de e_1 est une super-classe de la source (*resp.* la cible) de e_2 :
  (E1) $\mathrm{Src}_1 \in \mathcal{D}$, $\mathrm{Trgt}_1 \in \mathcal{D}$, $\mathrm{Src}_2 \in \mathcal{D}$ et $\mathrm{Trgt}_2 \in \mathcal{D}$, $\mathrm{Src}_2 \preceq \mathrm{Src}_1$, $\mathrm{Trgt}_2 \preceq \mathrm{Trgt}_1$, $\mathrm{ARs}_2 \subseteq \mathrm{ARs}_1$, $\mathrm{Card}_2 \leq \mathrm{Card}_1 \Rightarrow \mathrm{e}_2 \lhd \mathrm{e}_1$;
- Les nœuds de e_1 sont des classes de descripteurs, l'un au moins des nœuds de e_2 est une agrégation temporelle, la source (*resp.* la cible) de e_1 est une super-classe de la source (*resp.* la cible) de e_2 ou de l'ensemble des classes de descripteurs contraintes par la source (*resp.* la cible) de e_2 :
  - **(E2)** $\mathrm{Src}_1 \in \mathcal{D}$, $\mathrm{Trgt}_1 \in \mathcal{D}$, $\mathrm{Src}_2 \in \mathcal{D}$ et $\mathrm{Trgt}_2 \in \mathcal{X}$, $\mathrm{Src}_2 \preceq \mathrm{Src}_1$, $\forall \mathrm{d_t} \in d(\mathrm{Trgt}_2)$: $\mathrm{d_t} \preceq \mathrm{Trgt}_1$, $a(\mathrm{Trgt}_2) \cup \mathrm{ARs}_2 \subseteq \mathrm{ARs}_1$, $card(\mathrm{Trgt}_2) + 1 \leq \mathrm{Card}_1 \Rightarrow \mathrm{e}_2 \lhd \mathrm{e}_1$;
  - **(E3)** $\mathrm{Src}_1 \in \mathcal{D}$, $\mathrm{Trgt}_1 \in \mathcal{D}$, $\mathrm{Src}_2 \in \mathcal{X}$ et $\mathrm{Trgt}_2 \in \mathcal{D}$: se déduit de **(E2)** ;
  - **(E4)** $\mathrm{Src}_1 \in \mathcal{D}$, $\mathrm{Trgt}_1 \in \mathcal{D}$, $\mathrm{Src}_2 \in \mathcal{X}$ et $\mathrm{Trgt}_2 \in \mathcal{X}$, $\forall \mathrm{d_s} \in d(\mathrm{Src}_2)$: $\mathrm{d_s} \preceq \mathrm{Src}_1$, $\forall \mathrm{d_t} \in d(\mathrm{Trgt}_2)$: $\mathrm{d_t} \preceq \mathrm{Trgt}_1$, $a(\mathrm{Src}_2) \cup a(\mathrm{Trgt}_2) \cup \mathrm{ARs}_2 \subseteq \mathrm{ARs}_1$, $card(\mathrm{Src}_2) + card(\mathrm{Trgt}_2) + 1 \leq \mathrm{Card}_1 \Rightarrow \mathrm{e}_2 \lhd \mathrm{e}_1$;
- L'un au moins des nœuds de e_1 est une aggrégation temporelle, les nœuds de e_2 sont des classes de descripteurs, la source (*resp.* la cible) est une super-classe de la source (*resp.* la cible) de e_2, ou l'ensemble des classes de descripteurs contraintes par la source (*resp.* la cible) de e_1 ne contient qu'une unique classe de descripteur qui est une super-classe de la source (*resp.* la cible) de e_2 :
  - **(E5)** $\mathrm{Src}_1 \in \mathcal{D}$, $\mathrm{Trgt}_1 \in \mathcal{X}$, $\mathrm{Src}_2 \in \mathcal{D}$ et $\mathrm{Trgt}_2 \in \mathcal{D}$, $\mathrm{Src}_2 \preceq \mathrm{Src}_1$, $d(\mathrm{Trgt}_1) = \{\mathrm{d_t}\}$, $\mathrm{Trgt}_2 \preceq \mathrm{d_t}$, $\mathrm{ARs}_2 \subseteq \mathrm{ARs}_1 \cup a(\mathrm{Trgt}_1)$, $\mathrm{Card}_2 \leq 1 + card(\mathrm{Trgt}_1) \Rightarrow \mathrm{e}_2 \lhd \mathrm{e}_1$;
  - **(E6)** $\mathrm{Src}_1 \in \mathcal{X}$, $\mathrm{Trgt}_1 \in \mathcal{D}$, $\mathrm{Src}_2 \in \mathcal{D}$ et $\mathrm{Trgt}_2 \in \mathcal{D}$: se déduit de **(E5)** ;
  - **(E7)** $\mathrm{Src}_1 \in \mathcal{X}$, $\mathrm{Trgt}_1 \in \mathcal{X}$, $\mathrm{Src}_2 \in \mathcal{D}$ et $\mathrm{Trgt}_2 \in \mathcal{D}$, $d(\mathrm{Src}_1) = \{\mathrm{d_s}\}$, $\mathrm{Src}_2 \preceq \mathrm{d_t}$, $d(\mathrm{Trgt}_1) = \{\mathrm{d_t}\}$, $\mathrm{Trgt}_2 \preceq \mathrm{d_t}$, $\mathrm{ARs}_2 \subseteq \mathrm{ARs}_1 \cup a(\mathrm{Src}_1) \cup a(\mathrm{Trgt}_1)$, $\mathrm{Card}_2 \leq 1 + card(\mathrm{Src}_1) + card(\mathrm{Trgt}_1) \Rightarrow \mathrm{e}_2 \lhd \mathrm{e}_1$;
- La source (*resp.* la cible) de e_1 et la cible (*resp.* la source) de e_2 sont des aggrégations temporelles :
  - **(E8)** $\mathrm{Src}_1 \in \mathcal{D}$, $\mathrm{Trgt}_1 \in \mathcal{X}$, $\mathrm{Src}_2 \in \mathcal{X}$ et $\mathrm{Trgt}_2 \in \mathcal{D}$: se déduit de **(E3)** et **(E5)** ;
  - **(E9)** $\mathrm{Src}_1 \in \mathcal{X}$, $\mathrm{Trgt}_1 \in \mathcal{D}$, $\mathrm{Src}_2 \in \mathcal{D}$ et $\mathrm{Trgt}_2 \in \mathcal{X}$ se déduit de **(E6)** et **(E2)** ;
- Les sources respectives de e_1 et e_2 sont des classes de descripteurs, les cibles respectives de e_1 et e_2 sont des aggrégations temporelles, la source de e_1 est une super-classe de la source de e_2, et l'ensemble des arcs de la cible de e_1 subsume l'ensemble des arcs de la cible de e_2 :
  (E10) $\mathrm{Src}_1 \in \mathcal{D}$, $\mathrm{Trgt}_1 \in \mathcal{X}$, $\mathrm{Src}_2 \in \mathcal{X}$ et $\mathrm{Trgt}_2 \in \mathcal{X}$ se déduit de **(E3)** et **(E8)** ;
- Les règles **(E11)** à **(E16)**, qui traitent toutes les autres permutations de domaines des nœuds de e_1 et e_2, se déduisent de **(E10)** et de l'une des règles de **(E1)** à **(E7)**.

4 Un système expert d'aide à la classification taxonomique de classes de descripteurs

La structure de classification décrite dans la section précédente est mise en œuvre au sein d'un système expert (s.e.), développé en Prolog, dont l'objectif est d'aider le concepteur d'applications multimédias FERIA à classifier au mieux ses classes de des-

cripteurs. Toutes les règles de subsomption ont été implémentées et testées, à l'exception des règles (**E5**) à (**E16**) pour lesquelles le travail reste à faire. Enfin, le s.e. est, pour l'heure, une application indépendante : il n'a pas encore été intégré au framework FERIA.

Le principe général de fonctionnement de ce s.e. est le suivant : après initialisation (*cf.* section 3.2), l'utilisateur introduit ses propres types, classes de propriétés et classes de descripteurs selon les modalités énoncées dans la section 3.3, en particulier en indiquant une super-classe. Le s.e. calcule alors les autres subsumants possibles et, le cas échéant, les propose à l'utilisateur en lui demandant de faire un choix. Afin de faciliter la décision, il justifie explicitement les résultats de ses calculs. Cette justification consiste, pour l'essentiel, en l'explicitation et la clarification des traces d'exécution de Prolog.

4.1 Algorithme de classification

La classification suit les mêmes principes qu'il s'agisse de classes de descripteurs, de classes de propriétés ou de types : calcul de tous les subsumants possibles ; proposition des subsumants à l'utilisateur avec justification, pour chaque subsumant, du résultat de son calcul ; choix de l'utilisateur, mise-à-jour de la hiérarchie concernée ; répercussion du résultat de la classification sur les autres hiérarchies. L'algorithme ci-dessous détaille cette opération :

Algorithme 1
$classification(c) :=$

1. *Calcul de tous les subsumants possibles, déterminés suivants les règles définies à la section 3.4. Soit* $subsumer$ *la fonction qui, à toute classe, associe l'ensemble de ses subsumants ; si* $subsumer(c) \setminus sup^*(c) \neq \emptyset$ *:*
2. *Soit* $subsumer^+(c)$ *le choix de l'utilisateur. Si* $subsumer^+(c) \neq sup^*(c)$ *:*
3. *Si* $c \in \mathcal{D}$*, alors* $int^*(c) = int^*(c) \setminus int(sup^*(c)) \cap int^*(c)$ *;* $\forall c\prime \in sub(c) : c\prime \neq c, classification(c)$ *;* $\forall t \in \mathcal{T} : t := c$ *ou* $t := ref(c), classification(t)$*.*
   Si $c \in \mathcal{P}$*, alors* $\forall d \in \mathcal{D} : c \in int^*(d), classification(d)$ *;* $\forall t \in \mathcal{T} : t := rec(P)$ *et* $c \in P, classification(t)$*.*
   Si $c \in \mathcal{T}$*, alors* $\forall p \in \mathcal{P} : p := \langle a, c\rangle, classification(p)$*.*

4.2 Un exemple : structures logiques d'émissions télévisées

Prenons un exemple simple pour illustrer le fonctionnement du système expert. Soit la classe de descripteur tvshow qui représente une émission de télévision — tvshow — comme étant un générique de début — O — précédant un générique de fin — C : $int^*(\text{tvshow}) = \{\langle \text{tvshow_Segments}, G(\{\text{O}, \text{C}\}, \{\langle \text{O}, \text{C}, \text{precedes}\rangle\})\rangle\}$, où $int^*(\text{O}) = int^*(\text{C}) = \emptyset$, et $sup^*(\text{O}) = sup^*(\text{C}) = \text{TD}$; $sup^*(\text{tvshow}) = \text{TD}$, comme super-classe supposée. La figure 3 illustre la aggrégation temporelle de tvshow_Segments.

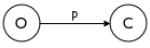

FIG. 3 – Type des segments de tvshow.

Introduits dans le système, O et C sont classifiés comme sous-classes de TD n'ayant pas d'autre subsumant possible. À l'introduction de la classe de propriété tvshow_Segments qui définit intensionnellement tvshow, le système crée le type correspondant à son co-domaine en le nommant tvshowSegments_Type et propose, justifications à l'appui, les subsumants suivants (extrait de l'interaction entre le s.e. et l'utilisateur) :

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

RootT is a subsumer of tvshowSegments_Type because:
        RootT is the user-defined superclass of tvshowSegments_Type.
pmoTS_Segment is a subsumer of tvshowSegments_Type because:
        the temporal structure of the former subsumes the latter's.
pmTS_Segment is a subsumer of tvshowSegments_Type because:
        the temporal structure of the former subsumes the latter's.
```

tvshowSegments_Type est subsumé par pmTS_Segments_Type car : (i) O et C spécialisent TD ; (ii) la relation p est plus spécifique que la relation [p $\vee$ m](+). Choisissons pmTS_Segments_Type pour super-classe directe de tvshowSegments_Type, le s.e. propose alors, sur le même mode interactif, un choix similaire pour la propriété tvshowSegments : RootP, pmoTS_Segments, pmTS_Segments. Choisissons là encore la plus spécifique, pmTS_Segments. Le s.e. effectue alors la classification de la classe de descripteur tvshow. Ci-dessous, un extrait de l'interaction avec l'utilisateur, au cours de laquelle le s.e. justifie ses propositions et donne la répartition des propriétés en fonction de chacun des choix possibles :

```
=== (DC) === Descriptor class tvshow (#) ===
Current subsumer (*) - Other possible subsumers (+):
RootD
|  D
|  |  (*) TD
|  |  |  (+) TS
|  |  |  |  (+) pmoTS
|  |  |  |  |  (+) pmTS
|  |  |  (+) C
|  |  |  (+) O
|  |  (#) tvshow

TD is a subsumer of tvshow because:
        TD is the user-defined superclass of tvshow.
        if you choose TD as the subsumer of tvshow, the properties distribution
        will be as follows:
                properties inherited from D: [D_Urn, DC_Urn, TP_Urn, DE_Urn]
                properties inherited from TD: [TemporalLocation]
                properties of tvshow: [tvshow_Segments]
...
pmTS is a subsumer of tvshow because:
        the properties of TS1 are either included in the property set of tvshow,
        or some are included and the others are super property classes
        of properties of tvshow. if you choose ...
```

Poursuivons notre exemple avec l'introduction des classes qui expriment la structure logique d'une pièce de théâtre français du 17ème siècle télévisée (structure évoquée en introduction à cet article) : tvplay est une segmentation temporelle dont la liste des segments est contrainte comme décrit à la figure 1, où P, A et S spécialisent TD.

tvplaySegments_Type est subsumé par tvshowSegments_Type, par application de la règle

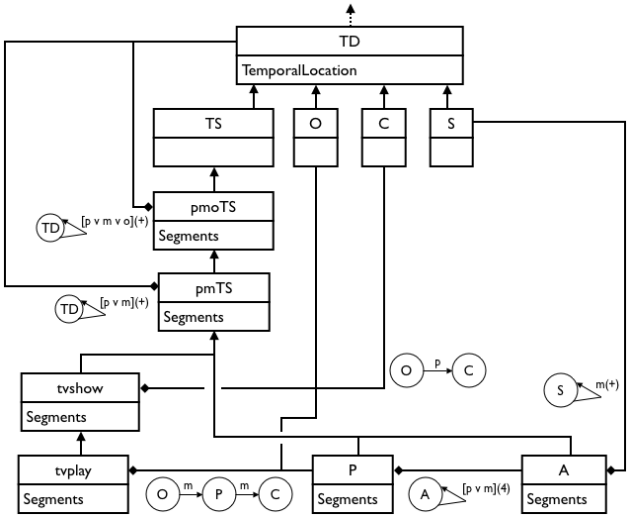

FIG. 4 – Extrait de la hiérarchie FDL après inférence des relations de subsomption.

(T1), car la relation p peut être inférée entre O et C par transivité de la realtion m. tvplaySegments_Type est subsumé par pmTS_Segments_Type, par application des règles **(E2)** à **(E4)**, car : (i) O, C, P, A et S spécialisent TD ; (ii) toutes les relations de tvplaySegments_Type sont plus spécifiques que [p ∨ m] ; (iii) le résultat de l'addition des cardinalités de tvplaySegments_Type est égal à (+), cardinalité de pmTS_Segments_Type. Ce qui permet d'inférer que tvplay spécialise tvshow. La figure 4 présente la hiérarchie résultant de l'interaction avec le s.e. sous forme de pseudo-UML.

Complexifions maintenant l'exemple avec l'introduction de classes de descripteurs qui expriment, de façon grossière, la structure temporelle logique d'un journal télévisé tvnews : elle est composée d'un générique de début — O — suivi du corps du journal — J — et du générique de fin — C ; le corps du journal étant lui-même composé de deux ou trois séquences R_Seq composées d'annonce de reportage R_A et de reportage R, suivies d'une séquence B_Seq composée d'une ou deux brèves B, le tout se répétant cinq fois (*c.f.* figure 5). À l'introduction, toutes ces classes ont TD pour super-classe ; tvnewsSegments_Type est subsumé par tvshowSegments_Type et pmTS_Type pour les mêmes raisons que tvplaySegments_Type, et par application des mêmes règles.

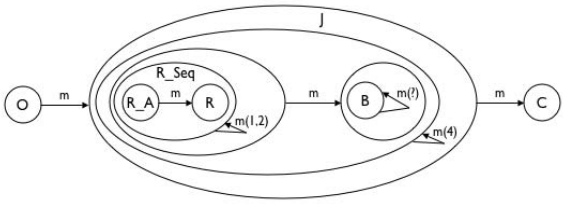

FIG. 5 – Structure logique temporelle d'un journal télévisé (tvnewsSegments_Type).

5 Conclusion et perspectives

Cet article a présenté la structure de classification, et son opérationnalisation au sein d'un système expert, des classes de descripteurs de documents audiovisuels qu'il est possible d'exprimer avec le langage FDL. Une des particularités de FDL est qu'il permet d'exprimer des types de données définis par une aggrégation temporelle qui contraint des classes de descripteurs à l'aide de relations de Allen enrichies de cardinalités et de paramètres numériques. FDL constitue le cœur du framework FERIA. Une première version du langage est actuellement opérationnelle ; elle est à l'œuvre dans plusieurs projets, (Caillet *et al.*, 2007) en particulier.

Les travaux vont tout d'abord se poursuivre, au niveau du système expert, d'une part avec l'implémentation des règles de subsomption d'arcs (**E5**) à (**E16**) (*c.f.* section 3.4), et d'autre part, avec l'amélioration des justifications fournis par le s.e. de façon à les rendre plus précises. Ils se développeront ensuite conjointement, au niveau de la structure de classification et de son opérationnalisation, avec la modification et/ou l'ajout de règles de subsomption d'arcs de façon à prendre en compte des structures de plus grande complexité. Un autre axe consiste à étudier l'intérêt d'établir une hiérarchie *part-of* de classes de descripteurs.

L'étape suivante verra l'introduction des instances de descripteurs, non pas pour effectuer un calcul d'instanciation au sens des logiques de description, mais dans le but de répondre à des requêtes complexes. Par exemple, supposons que nous disposions d'une segmentation en Alceste et d'une segmentation en Célimène d'une représentation télévisée du *Misanthrope* de Molière. Une requête pourrait consister à demander tous les segments contenant une réplique de Célimène suivie immédiatement d'une réplique d'Alceste. Une telle requête peut s'exprimer sous la forme d'une classe de descripteurs définie, notamment, par une propriété dont le type est une aggrégation temporelle. Répondre à la requête revient alors à instancier cette classe sur la base des instances respectives des classes de descripteurs représentant les segmentations en Alceste et Célimène.

Références

ALLEN J. F. (1983). Maintaining knowledge about temporal intervals. *Communications of the ACM*, **26**(11), 832 – 843.

ALLEN J. F. (1984). A general model of action and time. *Artificial Intelligence*, **23**(2).

AUFFRET G., CARRIVE J., CHEVET O. & DECHILLY T. (1999). Audiovisual-based hypermedia authoring : using structured rerpresentations for efficient access to av documents. In *ACM Hypertext '99*, Darmstadt, Germany.

F. BAADER, D. CALVANESE, D. MCGUINESS, D. NARDI & P. PATEL-SCHNEIDER, Eds. (2002). *The Description Logic Handbook*. Cambridge University Press.

BAILER W. & SCHALLAUER P. (2006). The detailed audiovisual profile : Enabling interoperability between mpeg-7 based systems. In *Proceedings of 12th Multimedia Modellinig Conference*, p. 217–224, Beijing, China.

BECHHOFER S., CALVANESE D., GIACOMO G. D., HORROCKS I., LUTZ C., MOTIK B., PARSIA B., SATTLER U. & PATEL-SCHNEIDER P. F. (2006). Owl 1.1 web ontology language, http ://owl1_1.cs.manchester.ac.uk/.

BRUNIE V., CARRIVE J. & VINET L. (2006). Ingénierie des documents audiovisuels : le projet feria. *TSI*, **25**(4), 469 – 496.

CAILLET M., CARRIVE J. & ROISIN C. (2005). Description des documents audiovisuels : s'affranchir des limitations de mpeg-7. In *MetSI05*, p. 69–78, Grenoble, France.

CAILLET M., CARRIVE J., ROISIN C. & YVON F. (2007). Engineering multimedia applications in the basis of multi-structured descriptions of audiovisual contents. In *Proceedings of SADPI'07*.

CARRIVE J., PACHET F. & RONFARD R. (2000). Clavis - a temporal reasoning system for classification of audiovisual sequences. In *Proceedings of RIAO*, Paris, France.

EUZENAT J., NAPOLI A. & BAGET J.-F. (2003). Xml et les objets (objectif xml). *L'Objet*, **9**(3), 11–37.

HORROCKS I., KUTZ O. & SATTLER U. (2006). The even more irresistible sroiq. In *10th Int. Conf. on Principles of Knowledge Representation and Reasoning (KR'06)*.

HORROCKS I., PATEL-SCHNEIDER P. F. & VAN HARMELEN F. (2003). From shiq and rdf to owl : The making of a web ontology language. *Journal of Web Semantics*, **1**(1), 7–26.

HUCHARD M. (2003). Classification de classes dans les approches à objets.

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

MARTÍNEZ J. M., KOENEN R. & PEREIRA F. (2002a). Mpeg-7 : The generic multimedia content description standard, part 1. *IEEE Multimedia*, **9**(2), 78–87.

MARTÍNEZ J. M., KOENEN R. & PEREIRA F. (2002b). Mpeg-7 : The generic multimedia content description standard, part 2. *IEEE Multimedia*, **9**(3), 83–93.

MCGUINNESS D. L. & VAN HARMELEN F. (2004). Owl web ontology language overview, http ://www.w3.org/tr/owl-features/.

NAPOLI A. (1997). *Une Introduction aux Logiques de Description*. Rapport interne 3314, INRIA.

SHIH T. K., HWANG L.-J. & TSAI J.-Y. (1996). Formal model of temporal properties underlying multimedia presentations. In *Multimedia Modeling*.

TRONCY R., BAILER W., HAUSENBLAS M. & SCHLATTE R. (2006). Enabling multimedia metadata interoperability by defining formal semantics of mpeg-7 profiles. In *1st Int. Conf. on Semantic and Digital Media Technologies (SAMT'06)*, p. 41–55, Athens, Greece.

TRONCY R., CARRIVE J., LALANDE S. & POLI J.-P. (2004). A motivating scenario for designing an extensible audio-visual description language. In *CORIMEDIA'04*, Sherbrooke, Canada.