

DGCIS-09.2.93.0966, ACAV

D3.1:

*Document de spécification de
modèles documentaires / feuilles de style*

Date de livraison contractuelle :	<i>30/10/2010</i>
Date de livraison effective :	<i>27/01/2011</i>
Lot :	<i>3.1</i>
Auteur et Institution :	ENCELLE Benoît (LIRIS)
Niveau de dissémination :	<i>Public</i>
Nature :	<i>Rapport</i>
Version :	<i>1.1</i>
Nombre total de pages:	<i>18</i>
Nom de fichier :	<i>ACAV-D3.1-2011-27-01-Modèles_presentation-v1.docx</i>
Mots clés :	<i>Présentation d'annotations, modèle documentaire, feuille de style, grammaire</i>
Résumé	
<i>Ce rapport présente un modèle pour la spécification de modèles de présentation d'annotations associées à des vidéos. Les éléments du modèle et une grammaire pour l'expression de modèles de présentation d'annotations y sont exposés.</i>	
The information in this document reflects only the author's views and the French ministry of industry is not liable for any use that may be made of the information contained therein. The information in this document is provided as is and no guarantee or warranty is given that the information is fit for any particular purpose. The user thereof uses the information at its sole risk and liability.	

Historique

Version	Date	Motif	Révisé par
1	2011-01-27	Première version	Benoît ENCELLE

Auteurs

Institution	Nom	Email
LIRIS- CNRS	Benoît ENCELLE	benoit.encelle@liris.cnrs.fr

Table des matières

1	Introduction	4
2	Modèle proposé pour la spécification de modèles de présentation.....	5
2.1	Définitions des différents éléments du modèles.....	5
2.1.1	Modèle de présentation	5
2.1.2	Piste d'enrichissement.....	5
2.1.3	Règle	5
2.1.4	Sélecteur.....	6
2.1.5	Actions	8
2.2	Contraintes supplémentaires sur les éléments Règle et Piste.....	12
3	Exemples de modèles de présentation d'annotations.....	13
3.1	Exemple tiré du scénario 5 :« description par des parents pour des jeunes enfants sourds ».....	13
3.1.1	Résumé.....	13
3.1.2	Modèles de présentation.....	13
3.2	Exemples tiré du scénario 6 :« description par des parents pour des jeunes enfants malvoyants ».....	14
3.2.1	Résumé.....	14
3.2.2	Modèles de présentation.....	14
4	Logique de génération d'un glossaire pour les symboles d'un modèle de présentation	15
6	Usage des modèles de présentation.....	16
7	Conclusion.....	17
8	Annexe 1 : grammaire EBNF pour la description d'un modèle de présentation .	18

1 Introduction

Ce document présente un « méta-modèle », c'est-à-dire un modèle pour la spécification de modèles de présentation d'annotations. Ce méta-modèle s'appuie sur le modèle d'annotations proposé en D2.1: « *Document de description des modèles d'annotations, spécification des vocabulaires contrôlés utilisés* ». La section suivante présente les différents éléments du modèle, en en donnant une définition semi-formelle et une formelle (représentation au format EBNF¹). La section 0 présente des exemples de modèles de présentation spécifiés à l'aide du formalisme proposé et issus des scénarios identifiés pour ACAV (cf. D1.1 : « *Scénarios détaillés* »). La section 5 présente la logique permettant de générer un glossaire associant aux différents symboles utilisés dans un modèle de présentation leur signification. La section 6 rappelle le contexte d'usage de ce méta-modèle en le situant dans la chaîne de production/diffusion d'une vidéo enrichie. La grammaire complète du méta-modèle présenté est finalement récapitulée en annexe.

¹ http://fr.wikipedia.org/wiki/Extended_Backus-Naur_Form

2 Modèle proposé pour la spécification de modèles de présentation

2.1 Définitions des différents éléments du modèles

2.1.1 Modèle de présentation

2.1.1.1 Définition

Données spécifiant la manière dont on "présente/rend" les annotations d'un package ACAV. Un modèle de présentation est composé de pistes d'enrichissement. Les modèles de présentation sont "sur les étagères" (i.e. des modèles de présentation préconfigurés sont directement utilisables par un auteur de modèles de présentation).

Soit M un modèle de présentation, M comprend un ensemble de pistes P :

$$M = \langle P_1, \dots, P_m \rangle \text{ avec } m > 0$$

2.1.1.2 Grammaire

`PresentationModel -> Track+ ;`

2.1.2 Piste d'enrichissement

2.1.2.1 Définition

Ensemble de règles. Une piste d'enrichissement est manipulable par l'utilisateur spectateur (activation/désactivation de la piste, paramétrage (e.g. si l'auteur du modèle de présentation a explicitement indiqué qu'un ou plusieurs paramètres de types d'action instanciés sont spécifiables par l'utilisateur spectateur)).

Soit P une piste donnée, P comprend un ensemble de règles :

$$P = \langle R_1, \dots, R_n \rangle \text{ avec } n > 0$$

2.1.2.2 Grammaire

`Track -> Rule+ ;`

2.1.3 Règle

2.1.3.1 Définition

De manière générale, une règle est une manière de présenter un sous-ensemble d'annotations suivant une ou plusieurs modalités de présentation.

Soit R une règle donnée, R comprend un sélecteur S et un ensemble d'actions (A_i représente une action donnée).

$$R = \langle S, \langle A_1, \dots, A_o \rangle \rangle \text{ avec } o > 0$$

2.1.3.2 Remarque

Une règle peut également s'appliquer à l'objet vidéo. Dans ce cas, le sélecteur est l'objet « Video ». La section « Sélecteur » ci-dessous ne détaille que le sélecteur d'annotations.

2.1.3.3 Grammaire

```
Rule -> ("Video" => VideoAction ("," VideoAction)*) | (Selector => Action
("," Action)*) ;
```

2.1.4 Sélecteur

2.1.4.1 Définition

Sélectionne des annotations suivant un ensemble de contraintes de différents types. L'élément central est l'annotation.

Types de contraintes:

- Structurelles : schémas, types, tags ;
- Intrinsèques : identifiant, contenu ;
- + opérateurs : logiques (entre contraintes ET, OU), ensemblistes (union, Intersection, différence).

Un sélecteur S est une expression qui retourne :

- une collection non ordonnée d'annotations.

2.1.4.2 Grammaire

```

Selector -> Expression ;

Expression -> CompoundExpression | LocationPath ;

CompoundExpression -> "{" Expression ( ("UNION" Expression)+ | ("INTERSECTION"
Expression)+ | ("DIFFERENCE" Expression)+ ) }" ;

LocationPath -> LocationSteps ("[" Predicat "]" ) * ;

LocationSteps -> LocationStep ("/" LocationStep) * ;

LocationStep -> (Axe ":" ) ? Test ;

Axe -> "schema" | "type" | "tag" | "annotation-id" ;

Test -> CompoundTest | SimpleTest ;

CompoundTest -> "(" Test ( ( "and" Test)+ | ( "or" Test)+ ) ")" ;

SimpleTest -> "*" | id ;

id -> string ;

```

2.1.4.3 Remarques sur les *LocationSteps*

Un *LocationSteps* est composé d'un ou plusieurs *LocationStep*. Un *LocationStep* indique une contrainte structurelle (voire intrinsèque pour le dernier axe) sur un sous-ensemble d'annotations à sélectionner.

2.1.4.3.1 Types d'axe d'un *LocationStep*

Les types d'axe suivants sont proposés :

- *schema* : pour filtrer la sélection d'annotations en fonction de leur appartenance ou non à des schémas ;
- *type* : pour filtrer la sélection d'annotations en fonction de leur appartenance ou non à des types ;
- *tag* : pour filtrer la sélection d'annotations en fonction de leur appartenance ou non à des « classes d'étiquettes » ;
- *annotation-id* (par défaut si non-précisé) : pour filtrer la sélection d'annotation en fonction d'identifiant(s) d'annotation.

2.1.4.3.2 *Forme abrégée et forme complète d'un LocationSteps, exemples de passage de formes abrégées à formes complètes*

Pour simplifier l'écriture d'un *LocationSteps*, celui-ci peut-être écrit sous une forme abrégée (cf. grammaire, la grammaire n'impose pas de spécifier des filtres pour chaque axe de localisation). Pour passer de la forme abrégée à la forme complète (et facilement interprétable) d'un *LocationSteps*, il faut remplir le patron suivant en « dépilant » sa forme abrégée de droite à gauche.

Patron : schema:*/type:*/tag:*/annotation-id:*

Exemples (forme abrégée, forme complète équivalente) :

1. `schema:VisualBase` → `schema:VisualBase/type:*/tag:*/annotation-id:*`
Sélectionne toutes les annotations liées au schema VisualBase
2. `schema:VisualBase/tag:(Niveau1 or Niveau2)` →
`schema:VisualBase/type:*/tag:(Niveau1 or Niveau2)/annotation-id:*`
Sélectionne toutes les annotations liées au schéma VisualBase et taguées « Niveau 1 » ou « Niveau2 »
3. `type:actions` → `schema:*/type:actions/tag:*/annotation-id:*`
Sélectionne toutes les annotations liées au type « actions »

2.1.4.4 Remarque sur les prédicats

Un prédicat (*predicat*) apporte une contrainte intrinsèque sur la sélection d'annotations. Plus précisément, c'est une condition sur leur contenu : un prédicat filtre l'ensemble d'annotations retournées en fonction de leur contenu. Un prédicat est assimilé à une expression régulière.

Predicat -> RegExp ;

RegExp -> string ;

2.1.4.5 Exemples de sélecteurs

Sélecteur	Signification
*	sélectionne toutes les annotations (axe par défaut annotation-id)
type:actions/*	sélectionne toutes les annotations de type "actions"
schema:visualBase/type:actions and settings	sélectionne toutes les annotations associées aux types actions ou settings du schema visualBase

2.1.5 Actions

2.1.5.1 Définition

Ensemble d'actions à réaliser pour la présentation des annotations ou sur l'objet « Video ».

Soit A une action donnée. A est définie par un type d'action T (associé à un ensemble de paramètres Pa(T)) et par un ensemble de couples $C = (Pa(T)_i, isUserModifiable_i)$ avec $1 \leq i \leq Card(Pa(T))$ et $isUserModifiable_i$ un booléen.

$$A = \langle T, C \rangle$$

$$A = \langle T, \langle (Pa(T)_1, isUserModifiable_1), \dots, (Pa(T)_p, isUserModifiable_p) \rangle \rangle$$

avec $p = Card(Pa(T))$ et $isUserModifiable_i$ un booléen, $1 \leq i \leq p$.

2.1.5.2 Types d'action

2.1.5.2.1.1 Contrôle des modalités

Les types d'actions suivants sont proposés :

- Oraliser l'annotation
- Afficher l'annotation en Braille
- Afficher l'annotation en sous-titre
- Afficher l'annotation à l'écran
- Jouer un son
- Afficher un symbole en Braille

2.1.5.2.1.2 Contrôle de la video

Les types d'actions suivants sont proposés :

- Appliquer une opération visuelle
- Appliquer une opération audio

2.1.5.3 Grammaire

```
UserMP -> ("userModifiable=" YesOrNo)?;
```

```
YesOrNo -> "yes" | "y" | "no" | "n" ;
```

```
Action -> ("Oralize" OralizeParameters) | ("DisplayBraille" BrailleParameters)
| ("DisplaySubtitle" SubtitleParameters) | ("Close-Caption" Close-
CaptionParameters) | ("Sound" SoundParameters) | ("BrailleSymbol"
BrailleSymbolParameters) ;
```

2.1.5.4 Ensemble de paramètres d'un type d'action donné

L'auteur de modèles de présentation indique, pour un type d'action donné, les paramètres manipulables par l'utilisateur spectateur. Pour une piste donnée, l'auteur du modèle de présentation peut donner par type d'action des valeurs de paramètres (l'ensemble des types d'action différents est donné en analysant l'ensemble des règles définies) et, pour chacun d'eux, peut indiquer si sa valeur est spécifiable ou non par l'utilisateur spectateur. Au niveau du rendu final, si l'utilisateur spectateur a défini des paramètres « importants », ceux-ci devront surcharger les paramètres des types d'action concernés.

2.1.5.4.1.1 Ensembles de paramètres génériques liés aux types de modalité employés

En fonction des caractéristiques des modalités de présentation employées, un ou plusieurs ensembles de paramètres génériques, présentés ci-dessous, pourront être employés.

- Modalité audio, *paramètres_audio* : panoramique, volume (%)

- Modalité temporelle, *paramètres_temporels* : caractérise une modalité présentant un message pouvant évoluer dans le temps
 - vitesse moyenne
 - seuil haut / seuil bas
 - mise en pause possible de la vidéo
- Modalité visuelle: *paramètres_visuels*: gauche, droite, haut, bas, largeur, hauteur (coin supérieur gauche)
 - *paramètres_texte_visuel* : police/ attributs/ taille/couleur

2.1.5.4.1.2 Ensemble de paramètres pour les différents types d'action envisagés

2.1.5.4.1.2.1 Contrôle des modalités

- Oraliser l'annotation (*paramètres_temporels*, voix, *fallback_to_Braille*)
- Afficher l'annotation en Braille (type de Braille, pause possible, *fallback_to_Speech*)
- Afficher l'annotation en sous-titre (*paramètres_texte_visuel*)
- Afficher l'annotation à l'écran (*paramètres_texte_visuel*, *paramètres_visuels*)
- Jouer un son (*fichier_son*, signification)
 - Rq : Le paramètre signification permet de décrire un son si celui-ci est vu comme un symbole
- Afficher un symbole en Braille (type de symbole, signification)

2.1.5.4.1.2.2 Contrôle de la vidéo

- Appliquer une opération visuelle
 - Emplacement (déterminé par l'annotation, arbitraire X%,Y%,XF%,YF%, toute l'image)
 - Opérateur
 - sur les couleurs
 - Filtre couleurs
 - Nuances de gris
 - Autres
 - zoom
 - type (fisheye, classique)
 - facteur
- Appliquer une opération audio
 - panoramique, volume

2.1.5.4.2 Grammaire

```
// Types génériques de paramètres %
AudioParameters -> ("pan=" panValue UserMP)? ("volume=" volumeValue UserMP)?;
TemporalParameters -> ("defaultPlaybackRate=" defaultPlaybackRateValue UserMP)?
("maxPlaybackRate=" percentage UserMP)? ("minPlaybackRate=" percentage
UserMP)? ("allowVideoPause=" boolean UserMP)?;
```

```

VisualParameters -> ("left=" percentage "top=" percentage "bottom=" percentage
"right=" percentage "align=" percentage "v-align=" percentage "width="
percentage "height=" percentage UserMP)?;

TextualVisualParameters -> VisualParameters? ("font-family=" font-familyValue
UserMP)? ("font-style=" font-styleValue UserMP)? ("font-size=" font-sizeValue
UserMP)? ("font-weight=" font-weightValue UserMP)? ("color=" colorValue
UserMP)? ("background-color=" background-colorValue UserMP)?;

// Paramètres par types d'action : contrôle des modalités %
OralizeParameters -> TemporalParameters ("voice-family=" voice-familyValue
UserMP)? ("fallbackToBraille=" boolean UserMP)?;

BrailleParameters -> ("brailleType=" brailleTypeValue UserMP)?
("allowVideoPause =" boolean UserMP)? ("fallbackToSpeech=" boolean UserMP)?;

SubtitleParameters -> TextualVisualParameters ;

Close-CaptionParameters -> TextualVisualParameters VisualParameters ("width="
percentage UserMP)? ("height=" percentage UserMP)?;

SoundParameters -> ("file=" string UserMP) ("meaning=" string)?;

BrailleSymbolParameters -> ("brailleSymbolType=" brailleSymbolTypeValue UserMP)
("meaning=" string);

// Paramètres par types d'action : contrôle de la Vidéo %
// Déclaration des valeurs possibles des paramètres
panValue -> "left" | "middle-left" | "middle" | "middle-right" | "right" ;
volumeValue -> percentage | "silent" | "x-soft" | "soft" | "medium" | "loud" |
"x-loud" ;
defaultPlaybackRateValue -> "0.25" | "0.5" | "0.75" | "1" | "1.25" | "1.5" |
"1.75" | "2" ;
font-familyValue -> string ;
font-styleValue -> "normal" | "italic" | "oblique" ;
font-sizeValue -> "xx-small" | "x-small" | "small" | "medium" | "large" | "x-
large" | "xx-large" ;
font-weightValue -> "normal" | "bold" | "bolder" | "lighter" | "100" | "200" |
"300" | "400" | "500" | "600" | "700" | "800" | "900" ;
colorValue -> "aqua" | "black" | "blue" | "fuchsia" | "gray" | "green" | "lime"
| "maroon" | "navy" | "olive" | "orange" | "purple" | "red" | "silver" | "teal"
| "white" | "yellow" ;
voice-familyValue -> ((specific-voice | generic-voice ) ",")* (specific-voice
| generic-voice) ;
background-colorValue -> colorValue | "transparent";
specific-voice -> string ;
generic-voice -> "male" | "female" | "child" ;
brailleTypeValue -> "integral" | "contracted" | "computer" ;
brailleSymbolTypeValue -> string ;

// Expressions régulières
//percentage -> ;
//string -> ;

```

//boolean -> ;

2.2 Contraintes supplémentaires sur les éléments Règle et Piste

Une règle n'a qu'une seule action de même type (une règle ne peut pas avoir x actions du même type, $x \geq 2$).

$$R = \langle S, \langle A_1, \dots, A_o \rangle \rangle \text{ et } A_i.T \neq A_j.T \\ \forall i, j \text{ tel que } 1 \leq i, j \leq o \text{ et } i \text{ différent de } j$$

Dans une piste et ses règles associées, les actions de même type ont les mêmes valeurs de paramètres. (Une piste ne peut pas avoir deux règles avec y actions de même type qui ont des paramètres différents, $y \geq 2$).

$$P = \langle R_1, \dots, R_n \rangle \text{ et } \forall R_i, R_j (1 \leq i, j \leq n ; i \neq j), \\ \forall R_i.A_k, R_j.A_l (1 \leq k \leq \text{Card}(R_i.A) \text{ et } 1 \leq l \leq \text{Card}(R_j.A)) \\ \text{si } R_i.A_k.T = R_j.A_l.T \text{ alors } R_i.A_k.C = R_j.A_l.C$$

Important : ces contraintes ne peuvent pas être représentées à l'aide du formalisme EBNF mais devront être intégrées à un éditeur de modèles de présentation.

3 Exemples de modèles de présentation d'annotations

Les exemples de modèles de présentation ci-dessous sont tirés du document D1.1 : « Scénarios détaillés ». Les modèles de présentation ci-dessous sont décrits à l'aide du formalisme proposé.

3.1 Exemple tiré du scénario 5 : « description par des parents pour des jeunes enfants sourds »

3.1.1 Résumé

Gérard, père d'un enfant sourd, complète les annotations générées par le module de Speech2Text sur une vidéo (annotation de type *TranscribedSpeech*, schéma *AudioTranscription*). Gérard en profite pour ajouter deux pistes pour les bruitages et crée un schéma *Bruits* (bruits d'objets, bruits d'animaux, etc.). Par défaut *les pistes destinées aux malentendants (les locuteurs et les bruits) sont présentées sous la forme de sous-titres en bas au centre de l'écran*. Gérard lance la vidéo enrichie pour vérifier la présentation de ces nouvelles pistes en plus des locuteurs. Gérard s'aperçoit qu'il y a des chevauchements graphiques entre les locuteurs et les bruits. Il gère *les chevauchements en indiquant que les annotations sur les bruits doivent être présentées en haut à gauche de l'écran*.

3.1.2 Modèles de présentation

Ce scénario fait apparaître un modèle de présentation : les annotations concernant les dialogues doivent être affichées sous la forme de sous-titres et les bruits en haut à gauche.

Modèle de présentation :

type : TranscribedSpeech => Subtitle

schema : Bruits => Close-Caption(x=0 y=0)

3.2 Exemples tiré du scénario 6 :« description par des parents pour des jeunes enfants malvoyants »

3.2.1 Résumé

Juliette, mère d'un enfant non-voyant annote à l'aide du schéma « Visual Base » une vidéo pour la rendre accessible aux non/malvoyants (description des décors, des personnages en présence, des actions des personnages, etc.). Une segmentation automatique des plans donne lieu à une nouvelle piste «plan » qu'elle corrige.

Elle vérifie son travail en lançant la vidéo enrichie en mode « rendu par défaut pour les non-voyants », *qui présente les annotations destinées aux non-voyants oralement* (i.e. à l'aide d'une synthèse vocale). A la maison, elle regarde la vidéo qu'elle vient de décrire avec son fils Louis qui a une certaine connaissance du Braille littéral et préfère avoir en présence de personnes voyantes et lorsque cela est possible, *un maximum de descriptions en Braille littéral.[...] et pour certaines (situées en même temps par exemple), une synthèse vocale.*

3.2.2 Modèles de présentation

Ce scénario fait apparaître deux modèles de présentation : l'un est celui par défaut pour les non-voyants/mal-voyants (braillistes ou non) et l'autre est un modèle pour braillistes.

3.2.2.1 Spécification du modèle de présentation par défaut pour les non/mal-voyants

Les annotations du schéma « Visual Base » sont présentées à l'aide d'une synthèse vocale. Nous ajoutons la présentation des changements de plan (annotations), transmis à l'aide d'un son (le bruit d'une goutte d'eau).

Modèle de présentation :

schema : VisualBase => Oralize

type : Shot => Sound(file=goutte.mp3)

3.2.2.2 Spécification du modèle de présentation pour les non/mal-voyants braillistes

Les annotations du schéma « Visual Base » sont présentées à l'aide d'une synthèse vocale. Lorsque deux annotations se chevauchent, la première dans le temps est présentée sur une plage Braille, la seconde à l'aide d'une synthèse vocale. Nous ajoutons également la présentation des changements de plan (annotations), transmis à l'aide d'un son (le bruit d'une goutte d'eau).

Modèle de présentation :

schema : *VisualBase* => *Oralize(fallbackToBraille=yes)*

type : *Shot* => *Sound(file=goutte.mp3)*

4 Logique de génération d'un glossaire pour les symboles d'un modèle de présentation

Un modèle de présentation d'annotations peut comporter des règles intégrant des actions présentant des « symboles ». Ces actions peuvent être soit de type « *Jouer un son associé à une signification* », soit de type « *afficher un symbole Braille* ».

Si une action de type « *Jouer un son* » dispose d'un paramètre *signification* (optionnel pour ce type d'action), alors cette action présentera un symbole sonore.

De ce fait, une simple analyse du modèle de présentation permettra de dégager les éléments nécessaires à la construction d'un tel glossaire, à savoir l'ensemble des symboles employés (sonores ou Braille) et leur signification.

5

6 Usage des modèles de présentation

Le processus proposé d'usage d'un modèle de présentation, situé dans package ACAV, est le suivant (cf. figure 1) :

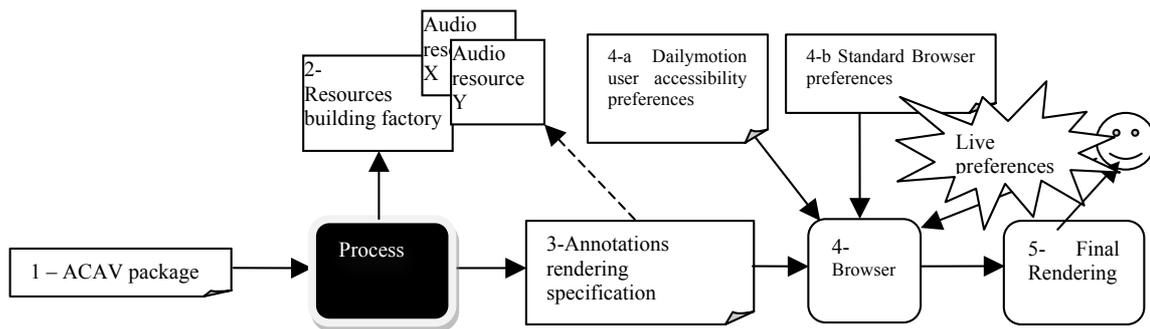


Figure 1. Processus global d'usage des modèles de présentation

Un package ACAV (1) contient les annotations et la description d'un modèle de présentation (le format utilisé pour la sérialisation des modèles de présentation reste à définir). Un processus (Process) va interpréter le modèle de présentation spécifié dans le package et demander la construction des ressources « statiques » (2) - i.e. certains enrichissements (e.g. fichiers mp3 pour les descriptions en synthèse vocale).

Ce processus va également générer un document (3) contenant les annotations et les informations de présentation (e.g. des liens vers les ressources statiques (e.g. fichiers mp3)), destiné à être en partie interprétable par un navigateur.

Ensuite, la vidéo pourra être lue parallèlement à ses enrichissements, qui pourront être personnalisés en fonction :

- premièrement des préférences « sur le moment » de l'utilisateur (i.e. live preferences) ;
- deuxièmement des préférences « statiques » associées à cet utilisateur (celles associées à son compte Dailymotion) ;
- et troisièmement des préférences indiquées dans le navigateur.

7 Conclusion

Ce document présente un modèle pour la spécification de modèles de présentation d'annotations. Les éléments du modèle sont présentés et une grammaire EBNF est indiquée pour l'expression de modèles de présentation. Des exemples illustrent l'usage des éléments du modèle et des éléments de la grammaire proposée. La logique de création d'un glossaire récapitulant l'ensemble des symboles pouvant être employés au sein d'un modèle de présentation est expliquée. Enfin, le contexte d'usage de ces modèles de présentation est indiqué, en les situant dans la chaîne de production/diffusion d'une vidéo enrichie.

Les éléments et la grammaire présentés constituent une première proposition qui pourra bien entendu évoluer. De plus, une interface graphique proposée à l'utilisateur final pour l'expression de modèles de présentation, s'appuyant sur cette grammaire, doit être développée.

8 Annexe 1 : grammaire EBNF pour la description d'un modèle de présentation

Une grammaire formelle (sous la forme EBNF (i.e. Extended Backus-Naur Form) est proposée pour l'expression de modèles de présentation.

Cette grammaire est prédictive (LL(1)) afin par exemple proposer à l'auteur de modèle de présentation, en cours d'écriture d'une règle, l'ensemble des possibilités correctes (c.à.d. respectant la grammaire) ou pour directement compléter sa production (si un seul choix de poursuite est possible).

```
PresentationModel -> Track+ ;

Track -> Rule+ ;

Rule -> ("Video" "=" VideoAction ("," VideoAction)*) | (Selector "=" Action
("," Action)*) ;

Selector -> Expression ;

Expression -> CompoundExpression | LocationPath ;

CompoundExpression -> "{" Expression ( ("UNION" Expression)+ | ("INTERSECTION"
Expression)+ | ("DIFFERENCE" Expression)+ ) "}" ;

LocationPath -> LocationSteps ("[" Predicat "]"*)* ;

LocationSteps -> LocationStep ("/" LocationStep)* ;

LocationStep -> (Axe ":")? Test ;

Axe -> "schema" | "type" | "tag" | "annotation-id" ;

Test -> CompoundTest | SimpleTest ;

CompoundTest -> "(" Test ( ( "and" Test)+ | ( "or" Test)+ ) ")" ;

SimpleTest -> "*" | id ;

id -> string ;

Predicat -> RegExp ;

RegExp -> string ;

UserMP -> ("userModifiable=" YesOrNo)?;

YesOrNo -> "yes" | "y" | "no" | "n" ;

Action -> ("Oralize" OralizeParameters) | ("DisplayBraille" BrailleParameters)
| ("DisplaySubtitle" SubtitleParameters) | ("Close-Caption" Close-
CaptionParameters) | ("Sound" SoundParameters) | ("BrailleSymbol"
BrailleSymbolParameters) ;

// Types génériques de paramètres %

AudioParameters -> ("pan=" panValue UserMP)? ("volume=" volumeValue UserMP)?;

TemporalParameters -> ("defaultPlaybackRate=" defaultPlaybackRateValue UserMP)?
("maxPlaybackRate=" percentage UserMP)? ("minPlaybackRate=" percentage
UserMP)? ("allowVideoPause=" boolean UserMP)?;

VisualParameters -> ("left=" percentage "top=" percentage "bottom=" percentage
"right=" percentage "align=" percentage "v-align=" percentage "width="
percentage "height=" percentage UserMP)?;
```

```

TextualVisualParameters -> VisualParameters? ("font-family=" font-familyValue
UserMP)? ("font-style=" font-styleValue UserMP)? ("font-size=" font-sizeValue
UserMP)? ("font-weight=" font-weightValue UserMP)? ("color=" colorValue
UserMP)? ("background-color=" background-colorValue UserMP)?;

// Paramètres par types d'action : contrôle des modalités %
OralizeParameters -> TemporalParameters ("voice-family=" voice-familyValue
UserMP)? ("fallbackToBraille=" boolean UserMP)?;
BrailleParameters -> ("brailleType=" brailleTypeValue UserMP)?
("allowVideoPause =" boolean UserMP)? ("fallbackToSpeech=" boolean UserMP)?;
SubtitleParameters -> TextualVisualParameters ;
Close-CaptionParameters -> TextualVisualParameters VisualParameters ("width="
percentage UserMP)? ("height=" percentage UserMP)?;
SoundParameters -> ("file=" string UserMP) ("meaning=" string)?;
BrailleSymbolParameters -> ("brailleSymbolType=" brailleSymbolTypeValue UserMP)
("meaning=" string);
// Paramètres par types d'action : contrôle de la Vidéo %
// Déclaration des valeurs possibles des paramètres
panValue -> "left" | "middle-left" | "middle" | "middle-right" | "right" ;
volumeValue -> percentage | "silent" | "x-soft" | "soft" | "medium" | "loud" |
"x-loud" ;
defaultPlaybackRateValue -> "0.25" | "0.5" | "0.75" | "1" | "1.25" | "1.5" |
"1.75" | "2" ;
font-familyValue -> string ;
font-styleValue -> "normal" | "italic" | "oblique" ;
font-sizeValue -> "xx-small" | "x-small" | "small" | "medium" | "large" | "x-
large" | "xx-large" ;
font-weightValue -> "normal" | "bold" | "bolder" | "lighter" | "100" | "200" |
"300" | "400" | "500" | "600" | "700" | "800" | "900" ;
colorValue -> "aqua" | "black" | "blue" | "fuchsia" | "gray" | "green" | "lime"
| "maroon" | "navy" | "olive" | "orange" | "purple" | "red" | "silver" | "teal"
| "white" | "yellow" ;
voice-familyValue -> ((specific-voice | generic-voice ) ",")* (specific-voice
| generic-voice) ;
background-colorValue -> colorValue | "transparent";
specific-voice -> string ;
generic-voice -> "male" | "female" | "child" ;
brailleTypeValue -> "integral" | "contracted" | "computer" ;
brailleSymbolTypeValue -> string ;
// Expressions régulières
//percentage -> ;
//string -> ;
//boolean -> ;

```