Rapport du projet CPP-XSD

Vers la mise en œuvre de la norme EN 17650:2022

2 juillet 2024

VDM
Vidéo Digital Multimedia

Sommaire

Sommaire **2**
Contexte du projet **4**
La norme EN 17650 4
Contexte 4
Historique de création de la norme EN 17650 4
Description rapide du contenu de la norme EN 17650 4
Besoin d'éléments opérationnels 5
Acteurs impliqués 6
CST 6
VDM 6
Validation des fichiers XML **7**
Problèmes à résoudre 7
Validation de la conformité structurelle 7
Schematron 7
Insertion dans la chaîne XML 7
Les principaux éléments d'un schéma Schematron 8
'pattern' 8
'rule' 8
'assert' 8
Mise en œuvre du format Schematron avec le processeur Saxon et schxslt 8
CaPPuSCHino 9
Un schéma Schematron par type de fichier 9
Besoin de mutualisation 10
Mise en œuvre des schémas 11
Difficultés rencontrées 11
Modularisation des fichiers de validation avec XSLT à défaut du mécanisme natif 11
Démonstration de faisabilité d'export de fichiers CPP **12**
Rendre le CPP accessible pour faciliter son adoption 12
L'intérêt du CPP pour les laboratoires 12
La méthodologie employée : création des premiers paquetages 13
Mapping des métadonnées 13
Apport de Schematron – exemple 14
Les étapes nécessaires 15
Validation de la faisabilité 15
Propositions d'améliorations de la norme **17**
Coquilles 17
Imprécisions 17
Erreurs 17
Désaccords sur les choix faits dans la norme 18
Utilisation des fichiers PREMIS 19
Conclusion **23**

Ressources.....24
Références.....25
Annexe 1 : liste des amendements à proposer pour la révision de la norme.....26

Contexte du projet

La norme EN 17650

Contexte

Reprendre l'exploitation d'une œuvre au-delà de 20 ans après sa sortie initiale présente de nombreux défis. La faisabilité d'une ressortie dépend grandement de l'état des éléments qui ont servi à son exploitation et de la présence/absence d'indications sur sa fabrication permettant une remasterisation conforme aux souhaits de l'auteur.
Or, si la conservation à très long terme d'une œuvre sur pellicule est un sujet désormais maîtrisé et qui comporte un certain nombre d'avantages et d'inconvénients sur lesquels nous ne reviendrons pas ici, il n'existait pas, avant la publication de la norme EN 17650 en octobre 2022, de cadre précis permettant une conservation pérenne, à plus de 20 ans, du fichier numérique d'une œuvre cinématographique.

Cette norme précise comment empaqueter et organiser, à des fins de conservation, les éléments constituant la version numérique de référence de l'œuvre, mais également des informations nécessaires à sa ré-exploitation future.

Historique de création de la norme EN 17650

La transition vers le cinéma numérique a offert de nouvelles opportunités d'échanges de contenu mais aussi de nouveaux défis en termes de conservation à long terme. Bien que de nombreux fichiers soient produits, il n'existait pas de format d'archive commun. La création d'un nouveau comité technique au sein du Comité Européen de Normalisation (CEN) a été proposé par le DIN, l'institut de normalisation allemand, suivi par de nombreux autres pays européens. La commission européenne a marqué son intérêt pour le projet en l'accompagnant d'un financement.
Un nouveau comité technique, le CEN/TC 457 a été créé, afin de discuter du contenu de cette future norme et de rédiger un rapport technique pour documenter son utilisation. Le CEN est l'un des comités de normalisation technique européen avec le CENELEC et l'ETSI.
En France, les travaux étaient suivis par la commission AFNOR/CN 204 "Cinéma".
Le comité technique spécialement mis en œuvre au sein du CEN a vu son effort accompagné par la création d'un groupe projet au vu de l'importante charge de travail.
Le 29 Octobre 2022, la norme européenne sur le paquetage de conservation a été publiée en France par l'AFNOR.

Description rapide du contenu de la norme EN 17650

Le Cinema Preservation Package est un "super paquetage" qui permet d'englober différents sous-paquetages avec des métadonnées additionnelles. Un mécanisme de playlist permet de décrire l'assemblage de plusieurs sous-paquetages corrélés.
Les sous-paquetages peuvent être constitués de :

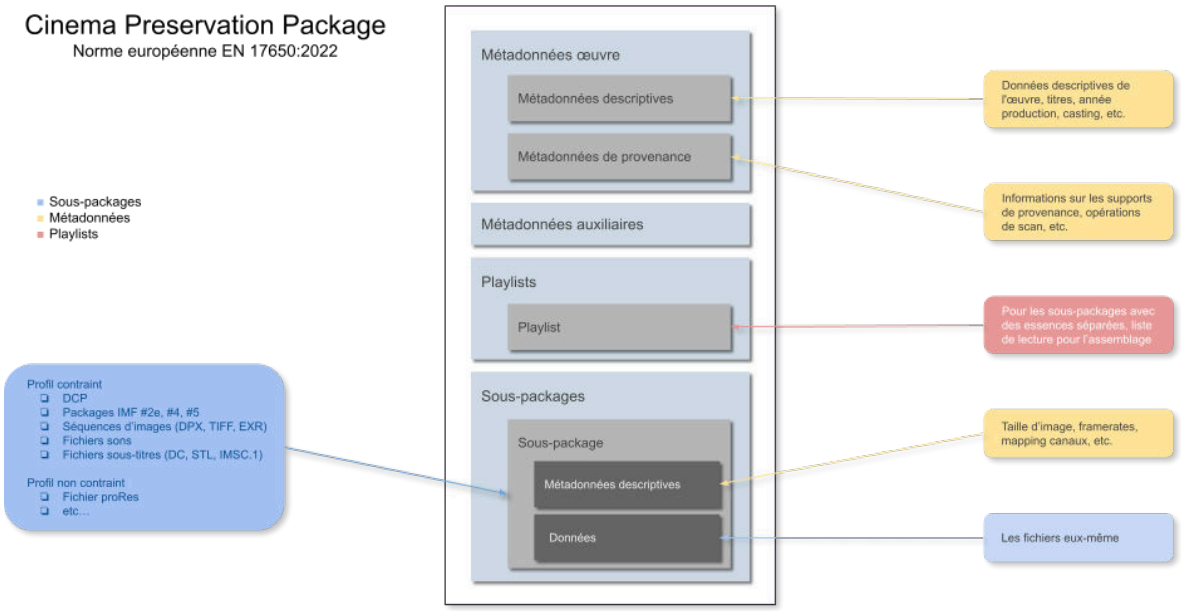
- séquences d'images, comme le DPX, le TIFF, etc.
- fichiers audio
- fichiers de sous-titres sous forme de textes ou d'images

- fichiers vidéo comprenant divers codecs audiovisuels dans des conteneurs standardisés
- paquetages à composants comme le DCP ou l'IMF

La liste des formats autorisés est limitative pour les paquetages ayant le profil "conservation à long terme" et ouverte dans les autres cas.

La norme permet de stocker pour chaque sous-paquetage des métadonnées techniques, au format EBUCore et de provenance, au format PREMIS, ces dernières permettant de garder des informations sur l'origine du contenu et les transformations appliquées.
Il permet également de stocker des informations d'identification, suivant la norme EN 15744, ainsi que des informations descriptives associées à l'œuvre ou à l'une de ses versions, également au format EBUCore.

Enfin, la structure du paquetage est décrite dans un fichier METS, qui permet de lister les sommes de contrôle pour vérifier l'intégrité, ainsi que des structures logiques permettant de décrire la manière d'articuler les sous-paquetages entre eux.



Besoin d'éléments opérationnels

Un travail de mise en œuvre opérationnelle est nécessaire pour trouver la bonne manière d'utiliser la norme.
Plusieurs aspects doivent être mis en œuvre :

- recueil des éléments média et extraction des métadonnées techniques
- compilation des métadonnées descriptives
- organisation des sous-paquetages
- réorganisation des métadonnées

Acteurs impliqués

CST

La CST est une association professionnelle de techniciens du cinéma et de l'audiovisuel. Elle est un lieu de concertation sur les métiers et les problématiques au sein des différents départements. Les groupes de travail peuvent aboutir à des recommandations techniques, qui pour certains sujets sont prolongées au niveau normatif.
L'équipe permanente de la CST soutient le travail des membres et plus largement du secteur.

Widad Amara est développeuse junior à la CST. Elle est étudiante en M2 Informatique - Parcours Ingénierie en Intelligence Artificielle (IIA) de l'Université Paris 8 Vincennes - Saint-Denis.

Hans-Nikolas Locher est directeur du développement à la CST. Il y anime l'équipe de développement logiciel. Il est également chargé du suivi normatif à l'AFNOR au sein de la commission CN 204, convenor du TC 36 / WG 4. Il était membre du groupe d'experts chargé de l'écriture de la norme EN 17650 en tant que rédacteur.

VDM

VDM est un laboratoire audiovisuel français créé en 1972, intégré au sein du Groupe Transatlantic depuis 2010. Il offre une gamme complète de services sur l'ensemble de la chaîne de d'exploitation et de conservation des œuvres audiovisuelles.

Yan Rocheteau est directeur de la fabrication au sein de VDM. Il encadre les responsables opérationnels des activités de Fabrication & Packaging, Restauration & Préservation, Voice-Over & Audiodescription, Sous-titrage. Ancien directeur technique, Yan Rocheteau dispose d'une vision globale des enjeux techniques et opérationnels liés à la préservation et à la diffusion des œuvres audiovisuelles.

Daniel Borenstein était chargé de liaison avec les archives de film dans le groupe d'experts au sein du comité CEN/TC 457. Il a été de plus mandaté par l'AFNOR pour valider la traduction française de la norme. Daniel Borenstein exerce aujourd'hui comme consultant en restauration et archivage, après avoir officié comme chef du service laboratoire/restauration au sein de la direction du patrimoine cinématographique du CNC, suite à une longue carrière dans l'image numérique et photochimique.

StudioCanal

Studiocanal est une société de production et de distribution cinématographique française, filiale du groupe Canal+. Elle est l'un des leaders européens dans la production, la distribution et la vente de films et de séries télévisées. Présent dans une dizaine de territoires par ses activités de production et distribution, StudioCanal a pleinement conscience de sa responsabilité dans la conservation du patrimoine cinématographique et audiovisuel.
Jean-Pierre Boiget est Directeur d'Exploitation & IT au sein de StudioCanal.

Validation des fichiers XML

Problèmes à résoudre

Validation de la conformité structurelle

La norme emploie intensivement des fichiers au format XML, utilisant des schémas répandus dans le domaine de l'archivage comme METS, EBUCore, PREMIS.
Bien qu'il soit possible de mettre en œuvre une validation de ces schémas, celle-ci ne s'applique qu'aux schémas initiaux. Or la norme définit par rapport à ceux-ci des sur-contraintes, que cette validation ne prend évidemment pas en compte.
Dans la norme ces sur-contraintes étant exprimées sous forme textuelles à l'aide de tableaux, il était nécessaire de les représenter sous une forme permettant la validation automatique des fichiers XML structurant les paquetages CPP.

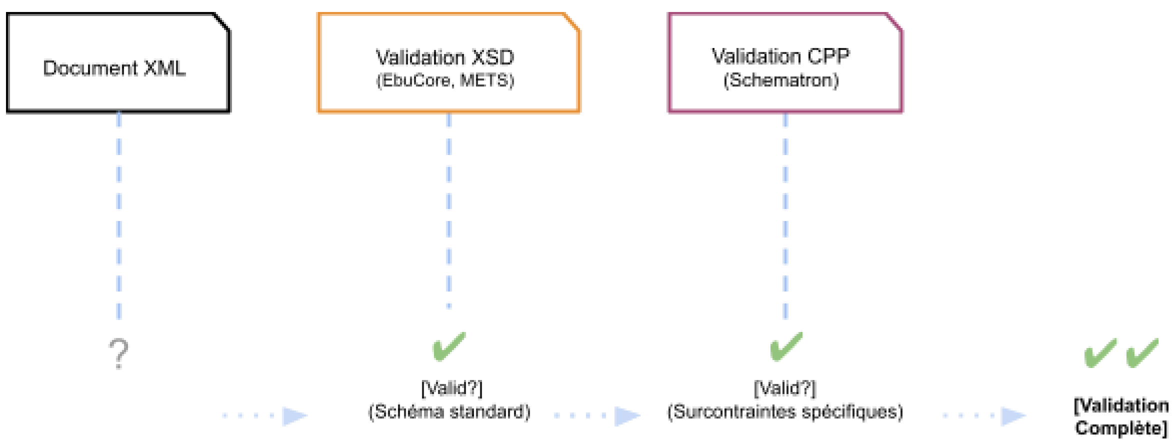
Schematron

L'approche retenue dans le projet a été d'utiliser la technologie XML normalisée *Schematron*. Une norme internationale [ISO/IEC 19757-3:2020] a été publiée et une implémentation de référence existe.

Insertion dans la chaîne XML

Schematron est une technologie complémentaire à la validation des schémas XML.
XML Schema permet de définir des fichiers avec une extension .xsd (XML Schema Definition) permettant de vérifier qu'un fichier XML est conforme à la définition du schéma.
Schematron permet d'ajouter la vérification des règles métiers.
La validation d'un fichier XML se fait en plusieurs étapes :

- vérification de la présence d'un prologue XML, de l'utilisation d'un encodage des caractères conforme à celui-ci et présence de balises ouvrantes et fermantes hiérarchiques cohérentes avec ou sans attributs ;
- validation du schéma revendiqué dans la référence aux schéma et espaces de nommages différents ;
- confirmation de règles métiers spécifiques.



Les principaux éléments d'un schéma *Schematron*

Schematron repose sur des motifs ('pattern') organisés composés de règles ('rule'), chaque 'rule' contenant des assertions ('assert') détaillées qui spécifient les conditions devant être validées pour assurer la conformité d'un document XML.

'pattern'

Ce sont des ensembles de 'rule' regroupées qui décrivent des conditions à vérifier dans un document XML. Chaque 'pattern' identifie une structure logique ou un schéma de validation spécifique appliqué à une partie ou à l'ensemble d'un document XML. Ils permettent de capturer des motifs récurrents ou des structures typiques que l'on souhaite vérifier dans le XML.

'rule'

Ce sont des unités individuelles de validation au sein d'un 'pattern'. Chaque 'rule' définit une condition spécifique que le document XML doit respecter pour être considéré comme valide. Elles peuvent être utilisées pour vérifier des exigences spécifiques, des contraintes métier ou des recommandations.

'assert'

Ce sont les déclarations spécifiques à l'intérieur d'une 'rule' qui définissent une condition précise et attendue dans le document XML. Elles expriment une exigence ou une restriction qui doit être vérifiée. Un 'assert' peut contenir des expressions XPath pour cibler les parties spécifiques du document XML et vérifier les valeurs, les relations ou d'autres critères.

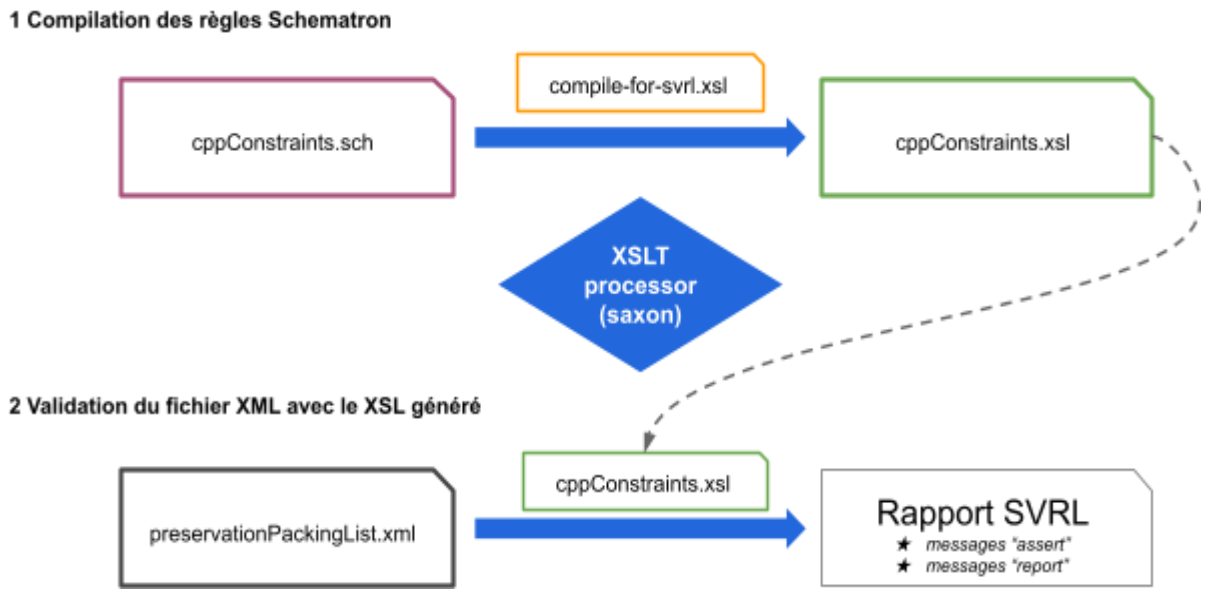
Dans le cadre du CPP, la norme EN 17650 surcontraint les schémas XML utilisés, par exemple le schéma METS peut autoriser la présence d'une balise `<File>` de manière optionnelle, mais la norme CPP renforce la présence de la balise de manière obligatoire. La validation du schéma METS sur un fichier `preservationPackingList.xml` est donc une condition nécessaire, mais pas suffisante.

Le rôle des fichiers *Schematron* développés est donc de vérifier la bonne application des sur-contraintes livrées dans les tableaux de la norme, mais aussi de vérifier des contraintes métiers plus spécifiques de la norme.

Mise en œuvre du format *Schematron* avec le processeur *Saxon* et *schxslt*

Le projet *schxslt* sur lequel nous nous sommes appuyés met en œuvre la norme *Schematron* sous la forme d'une feuille de transformation XSLT.
Le mécanisme de XSLT est d'exprimer une transformation d'un fichier XML organisé d'une certaine manière, vers un fichier XML organisé autrement. La transformation est elle-même formalisée dans un langage XML selon la spécification XSLT du w3c. On utilise un processeur XSLT qui va exécuter les instructions du fichier XSLT. Le fichier XSLT est en quelque sorte le code qui s'exécute sur la machine constituée par la processeur XSLT.

Dans le cas de l'implémentation *schxslt*, il s'agit de transformer un fichier *Schematron* en un fichier XSLT via des instructions de transformations qui sont elles-mêmes formulées en XSLT.



CaPPuSCHino

La base de code proposée par la CST pour la validation des fichiers XML du CPP est composée de fichiers XML en *Schematron*.

On retrouve bien sûr la structure des ‘pattern’, ‘rule’ et ‘assert’ de *Schematron*, les ‘pattern’ regroupant des vérifications liées entre elles.

```
<sch:pattern xmlns="http://purl.oclc.org/dsdl/schematron"
             xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
             id="commonMetsHdrConstraints">
  <!-- Constraints on metsHeader section -->
  <sch:rule context="/mets:mets/mets:metsHdr">
    <sch:assert test="matches(@CREATEDATE, '^\d{4}-\d{2}-\d{2}T(?:\d{2}:\d{2}:\d{2}(?:\.\d+)?)(?:Z|[+-]\d{2}:\d{2})?$')">
      Mandatory attribute 'CREATEDATE' in metsHdr section is missing and shall be in ISO 8601-1 format (cf. Table 1)
    </sch:assert>
    <sch:assert test="count(mets:agent[@ROLE='CREATOR' and @TYPE='OTHER']) = 1">
      Exactly one agent element with attributes ROLE='CREATOR' and TYPE='OTHER' shall be present in metsHdr section (cf. Table 2)
    </sch:assert>
    <sch:assert test="count(mets:agent/mets:name) =1">
      Exactly one name element within agent  shall be present (cf. Table 3)
    </sch:assert>
  </sch:rule>
</sch:pattern>
```

Bien que qu’il soit techniquement possible d’opérer d’autres types de vérification, il a été décidé de se focaliser sur les vérifications structurelles. D’autres types de vérification sont parfois ébauchés, sous forme non contraignante, donnant lieu à une ligne de ‘warning’ dans les rapports SVRL.

Un schéma *Schematron* par type de fichier

La norme décrivant le format CPP fait apparaître plusieurs types de fichiers XML.
La `preservationPackingList.xml` est une sorte d’index général pour l’ensemble du paquetage CPP. Les fichiers `packingList.xml` en sont l’équivalent pour chacun des sous-paquetages. Un répertoire principal à la racine permet de stocker des métadonnées descriptives, et chaque sous-paquetage contient des métadonnées descriptives propres. Ce

sont autant de types de fichiers à valider, avec des contraintes propres, mais avec aussi des contraintes communes.

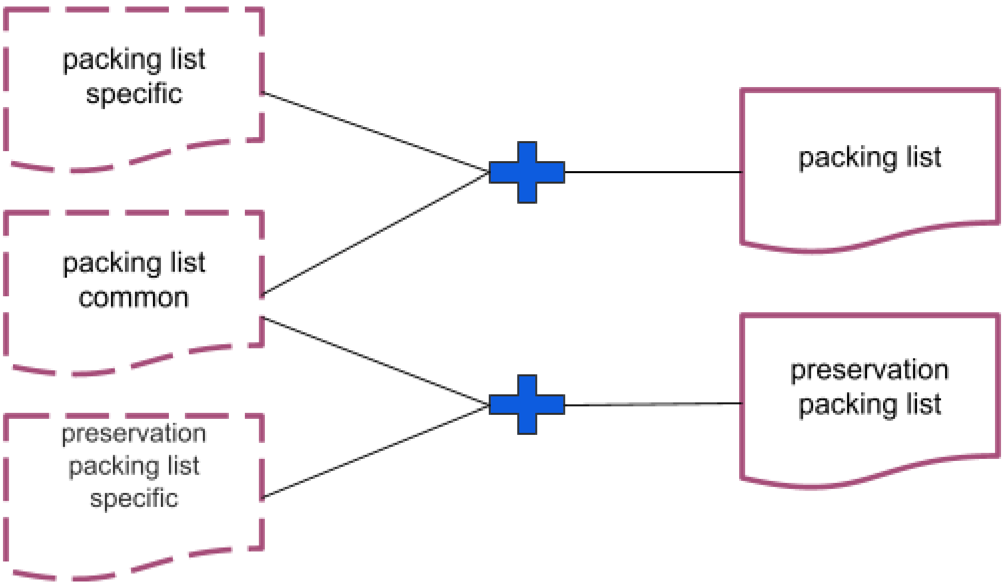
A chaque type de fichier XML dans le paquetage CPP correspond un fichier *Schematron* avec l'ensemble des règles qu'il doit vérifier.

Besoin de mutualisation

Mais certaines règles sont communes à plusieurs types de fichiers. Pour éviter une maintenance compliquée, nous avons cherché à mutualiser autant que possible l'écriture des règles.

Certaines contraintes dans les fichiers XML se répètent, ce qui se matérialise dans la norme par une factorisation avec des renvois vers des parties communes.

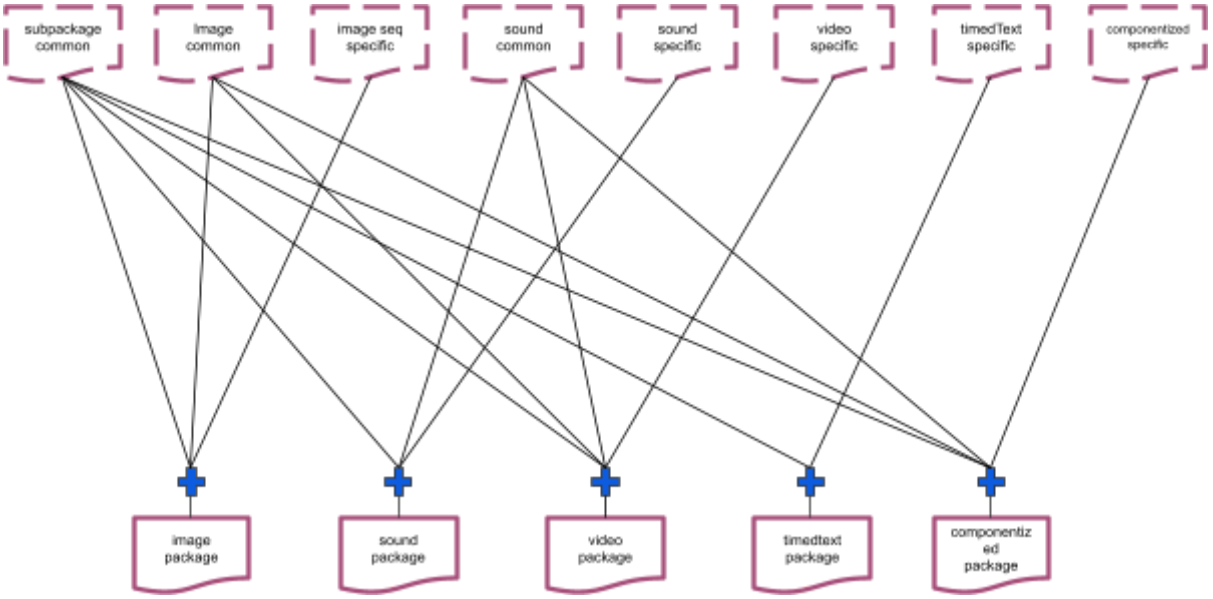
Le fichier `preservationPackingList.xml`, au format XML METS, partage par exemple une partie de structure commune avec les fichiers `packingList.xml` dans les sous-paquetages.



Pour les métadonnées techniques des sous-paquetages, on distingue trois niveaux de réutilisation :

- des règles communes de validation générale: contient les règles de validation générales applicables à tous les sous-paquetages.
- des règles partagées : certains types de sous-paquetages comme les imagePackage et les soundPackage partagent des contraintes communes avec d'autres, comme les audiovisualPackage et componentizedPackage.
- des règles spécifiques : chaque type de sous-paquetage nécessite des règles spécifiques adaptées aux exigences techniques uniques de ce type de fichier.

Cette mutualisation permet une validation précise et modulaire, facilitant l'ajout de nouveaux types de paquetages ou la modification des règles existantes en fonction de l'évolution de la norme ou des besoins métier.

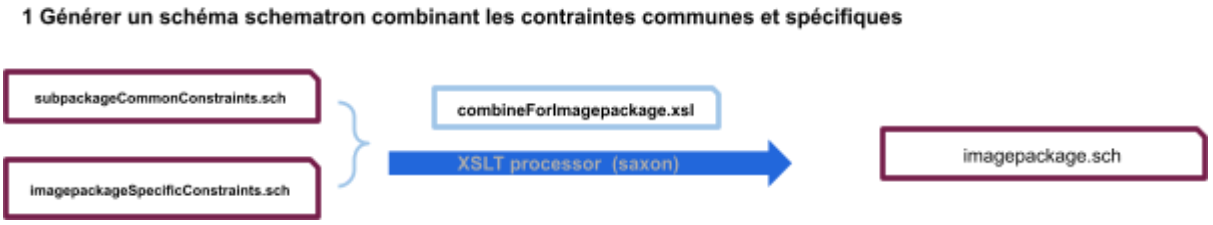


Mise en œuvre des schémas

Difficultés rencontrées

Modularisation des fichiers de validation avec XSLT à défaut du mécanisme natif

Le langage *Schematron* possède un mécanisme permettant d'inclure un sous-ensemble de '*rule'* ou de '*pattern'* présents dans un fichier tiers depuis un fichier principal. Malgré nos efforts, nous ne sommes pas parvenus à faire fonctionner correctement cette inclusion avec le processeur *Saxon* et le projet *schxslt*. Nous aurions bien entendu préféré utiliser ce mécanisme natif.
Un autre mécanisme pour gérer la modularisation a donc été mis en place, en privilégiant les technologies déjà employées. Pour éviter d'utiliser un préprocesseur tiers, il a été décidé d'utiliser XSL, le processeur XSLT étant par hypothèse déjà installé.



Cette opération d'assemblage est effectuée en appelant des fichiers XSLT listant les combinaisons à réaliser pour obtenir les fichiers complets. Il a été décidé de mettre à disposition dans le dépôt de code, à la fois les fichiers modulaires et les fichiers combinés, même si les transformations nécessaires sont documentées sous forme de texte et de scripts.

Démonstration de faisabilité d'export de fichiers CPP

Rendre le CPP accessible pour faciliter son adoption

Dès lors que les premières publications ont été faites autour du CPP, VDM s'est intéressé à ce sujet tant il vient répondre à une vraie problématique connue et identifiée de tous les acteurs du marché, mais sans réponse concrète, évidente, et adoptée par tous.

Ce genre de sujet est toujours délicat à appréhender car le marché a parfois tendance à rester prudent vis-à-vis de l'arrivée d'une nouvelle norme, à plus forte raison quand celle-ci concerne un format dont la vocation première (*assurer la pérennité d'une œuvre à très long terme*) ne représente qu'une source de coût et pas un centre de profits immédiats. Or, plus le temps passe, plus il peut s'avérer complexe et chronophage de récupérer les informations et les assets nécessaires à la constitution d'un CPP. L'opération devient ainsi plus coûteuse, mais également plus incertaine.

Il pourrait être tentant d'imposer la récupération systématique d'un set de métadonnées et d'assets de manière obligatoire lors de l'import initial dans un MAM, mais cette approche est peu réaliste tant elle apparaît contraignante. Ainsi, pour que le CPP puisse exister réellement et s'imposer, la réponse doit nécessairement être la plus « simple » et accessible possible.

L'intérêt du CPP pour les laboratoires

L'intérêt des clients ayants-droits de VDM sur le sujet du CPP est réel et croissant, et c'est pour cette raison que la laboratoire a lancé dès septembre 2022 une étude interne de la norme et le développement d'une première version de générateur de CPP.

Cette étude a permis a VDM de collecter un premier lot de remarques et de questions qui ont amenés à prendre contact avec Daniel Borenstein et Hans-Nikolas Locher pour partager ces premiers résultats et mettre en avant le besoin, selon VDM, de disposer d'un validateur officiel de CPP pour permettre des échanges de paquetages CPP entre clients et prestataires.

VDM a ainsi pu intégrer ce groupe de travail mi-2023, avec pour rôle d'apporter le retour d'expérience d'un laboratoire « historique » habitué à traiter en entrée et en sortie des contenus venant de plusieurs centaines de partenaires à travers le monde.

Pour rappel, l'objectif principal de l'étude portait sur la définition d'un ensemble de bonnes pratiques, permettant de valider la structure d'un CPP, et in-fine de pouvoir disposer d'outils pour assurer la génération « correcte » d'un CPP. Dans la mesure où de nombreux acteurs sur le marché peuvent être amenés à fabriquer des CPP, un des enjeux majeurs dans la mise en place de ces bonnes pratiques et des outils qui en découlent est de permettre un échange « facile » des CPP fabriqués.

Ainsi, en tant que laboratoire, l'objectif de VDM au travers de cette étude était de valider que l'approche proposée était fonctionnelle, fiable et automatisable. Dans le cas de l'entreprise, cela signifie donc de procéder à partir des assets et métadonnées existants dans la plateforme de media asset management : mediaspot®, en utilisant de véritables contenus.

Pour cela, le soutien de StudioCanal, partenaire de VDM, et très impliqué dans la conservation de son patrimoine cinématographique et audiovisuel, s'est avéré crucial pour pouvoir valider les travaux dans un environnement réel.

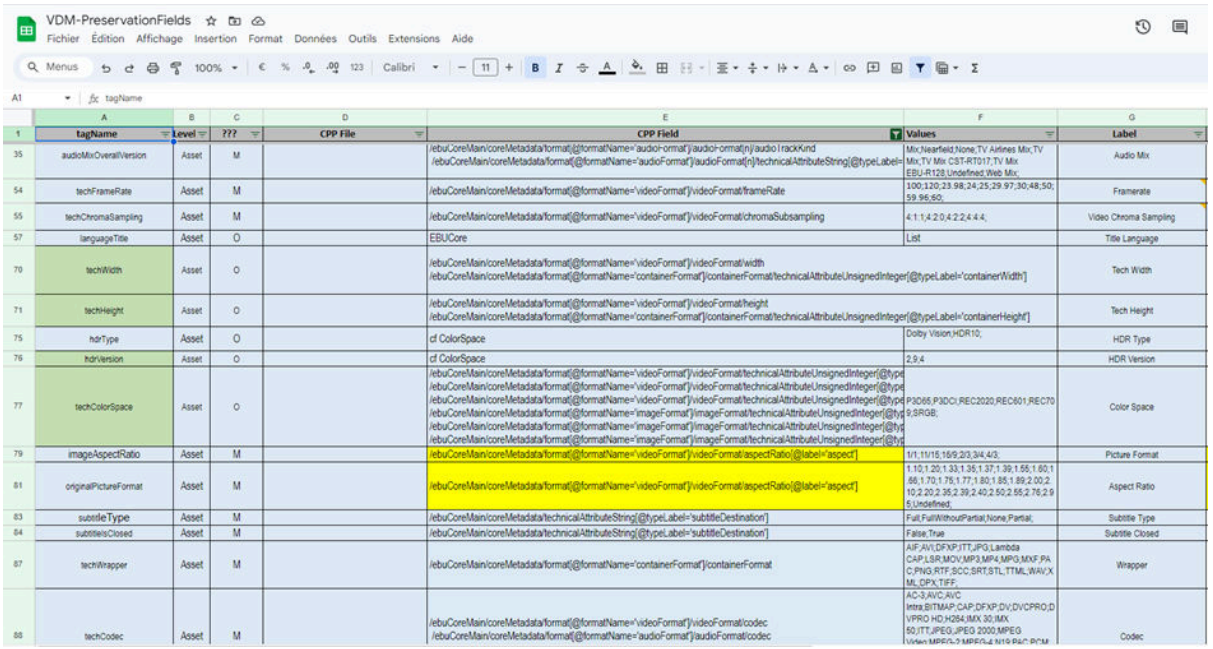
La méthodologie employée : création des premiers paquetages

Le premier constat que nous avons fait en démarrant cette étude, c'est que nous ne disposions d'aucun exemple de paquetage CPP. Aussi, notre approche a donc d'abord consisté à créer nos premiers CPP, à partir des métadonnées dont nous disposions, et en s'appuyant sur les XSD existants pour s'assurer de la conformité structurelle de ces premiers paquetages « tests » créés. Cette première étape nous a semblé indispensable pour approfondir ce sujet.

Mapping des métadonnées

Le cœur de notre travail a d'abord consisté à effectuer le mapping des métadonnées à partir de notre plateforme pour pouvoir créer des fichiers XML répondant aux normes METS, EBUCore, et PREMIS.

Ce processus de mapping est essentiel pour garantir l'interopérabilité entre différents systèmes d'information et pour assurer que les CPP générés soient conformes aux normes en vigueur.



	A	B	C	D	E	F	G
1	tagName	Level	???	CPP File	CPP Field	Values	Label
35	audioMixOverallVersion	Asset	M		/ebuCoreMain/coreMetadata/format[@formatName='audioFormat']/audioFormat[n]/audioTrackKind /ebuCoreMain/coreMetadata/format[@formatName='audioFormat']/audioFormat[n]/technicalAttributeString[@typeLabel=	Mix;Nearfield;None;TV Airlines Mix;TV Mix;TV Mix CST-RT017;TV Mix EBU-R128;Undefined;Web Mix;	Audio Mix
54	techFrameRate	Asset	M		/ebuCoreMain/coreMetadata/format[@formatName='videoFormat']/videoFormat/frameRate	100;120;23.98;24;25;29.97;30;48;50;59.96;60;	Framerate
55	techChromaSampling	Asset	M		/ebuCoreMain/coreMetadata/format[@formatName='videoFormat']/videoFormat/chromaSubsampling	4:1:1;4:2:0;4:2:2;4:4:4;	Video Chroma Sampling
57	languageTitle	Asset	O		EBUCore	List	Title Language
70	techWidth	Asset	O		/ebuCoreMain/coreMetadata/format[@formatName='videoFormat']/videoFormat/width /ebuCoreMain/coreMetadata/format[@formatName='containerFormat']/containerFormat/technicalAttributeUnsignedInteger[@typeLabel='containerWidth']		Tech Width
71	techHeight	Asset	O		/ebuCoreMain/coreMetadata/format[@formatName='videoFormat']/videoFormat/height /ebuCoreMain/coreMetadata/format[@formatName='containerFormat']/containerFormat/technicalAttributeUnsignedInteger[@typeLabel='containerHeight']		Tech Height
75	hdrType	Asset	O		cf ColorSpace	Dolby Vision;HDR10;	HDR Type
76	hdrVersion	Asset	O		cf ColorSpace	2.9.4	HDR Version
77	techColorSpace	Asset	O		/ebuCoreMain/coreMetadata/format[@formatName='videoFormat']/videoFormat/technicalAttributeUnsignedInteger[@type /ebuCoreMain/coreMetadata/format[@formatName='videoFormat']/videoFormat/technicalAttributeUnsignedInteger[@type /ebuCoreMain/coreMetadata/format[@formatName='videoFormat']/videoFormat/technicalAttributeUnsignedInteger[@type /ebuCoreMain/coreMetadata/format[@formatName='imageFormat']/imageFormat/technicalAttributeUnsignedInteger[@ty /ebuCoreMain/coreMetadata/format[@formatName='imageFormat']/imageFormat/technicalAttributeUnsignedInteger[@ty /ebuCoreMain/coreMetadata/format[@formatName='imageFormat']/imageFormat/technicalAttributeUnsignedInteger[@ty	P3D65;P3DCI;REC2020;REC601;REC709;SRGB;	Color Space
79	imageAspectRatio	Asset	M		/ebuCoreMain/coreMetadata/format[@formatName='videoFormat']/videoFormat/aspectRatio[@label='aspect']	1/1;11/15;16/9;2/3;3/4;4/3;	Picture Format
81	originalPictureFormat	Asset	M		/ebuCoreMain/coreMetadata/format[@formatName='videoFormat']/videoFormat/aspectRatio[@label='aspect']	1:1;1.20;1.33;1.35;1.37;1.39;1.55;1.60;1.66;1.70;1.75;1.77;1.80;1.85;1.89;2.00;2.10;2.20;2.35;2.39;2.40;2.50;2.55;2.76;2.95;Undefined;	Aspect Ratio
83	subtitleType	Asset	M		/ebuCoreMain/coreMetadata/technicalAttributeString[@typeLabel='subtitleDestination']	Full;FullWithoutPartial;None;Partial;	Subtitle Type
84	subtitleIsClosed	Asset	M		/ebuCoreMain/coreMetadata/technicalAttributeString[@typeLabel='subtitleDestination']	False;True	Subtitle Closed
87	techWrapper	Asset	M		/ebuCoreMain/coreMetadata/format[@formatName='containerFormat']/containerFormat	AIF;AVI;DFXP;ITT;JPG;Lambda CAP;LSR;MOV;MP3;MP4;MPG;MXF;PAC;PNG;RTF;SCC;SRT;STL;TTML;WAV;XML;DPX;TIFF;	Wrapper
88	techCodec	Asset	M		/ebuCoreMain/coreMetadata/format[@formatName='videoFormat']/videoFormat/codec /ebuCoreMain/coreMetadata/format[@formatName='audioFormat']/audioFormat/codec	AC-3;AVC-Intra;BITMAP;CAP;DFXP;DV;DVCPRO;DVPRO HD;H264;IMX 30;IMX 50;ITT;JPEG;JPEG 2000;MPEG	Codec

Une fois ce travail réalisé, nous avons réussi à générer des CPP « conformes » en s'appuyant sur les XSD mais cette vérification structurelle basique nous a semblé

insuffisante. En effet, elle ne porte que sur une validation partielle et essentiellement syntaxiale, mais ne permet pas d'aller suffisamment en profondeur.

Nous avons remonté cette information lors de nos différentes réunions de travail et c'est ainsi que la CST a proposé l'utilisation et l'intégration de *Schematron*.

Apport de *Schematron* – exemple

La norme indique que le fichier EBUCore d'un audiovisualPackage doit contenir un élément « format » dont le formatName est « containerFormat »

Table 55 — <coreMetadata> mandatory children for Audiovisual Package

Topic	EBUCore element	Element Arity	Attribute or other subelement	Subelement mandatory	Value(s)
Package metadata	format	1	@formatName	x	"audiovisualPackageFormat"
			#children	x	see 11.3.1.6.2
Container metadata	format	1	@formatName	x	"containerFormat"
			#children	x	see 11.3.1.6.3
VideoStreams metadata	format	1..n	@formatName	x	"videoFormat"
			#children	x	see 11.3.1.6.5
AudioStreams metadata	format	0..n	@formatName	x	"audioFormat"
			#children	x	see 11.3.1.6.7

Dans l'extrait ci-dessous, l'élément « format » est erroné

```
▼<format formatName="container_Format">
  ▼<containerFormat containerFormatName="QuickTime">
     <technicalAttributeString
     typeLabel="containerStandardReference">urn:cen.eu:en17650:2022:ns:metadata#UnknownStandard</technicalAttributeString>
     <technicalAttributeUri
     typeLabel="containerStandardReference">urn:cen.eu:en17650:2022:ns:metadata#UnknownStandard</technicalAttributeUri>
   </containerFormat>
   <mimeType typeLabel="video/quicktime"/>
 </format>
```

Le document XML est cependant conforme au schema EBUCore

```
2024-06-18 13:55:49.8879 [INFO ] EBUCore - Package XSD Validation OK for "C:\CPP_Samples\MONSIEUR-N\audiovisualPackage
_c2ec52ea-1534-4f16-bde6-0af647749008\metadata\techMD_c2ec52ea-1534-4f16-bde6-0af647749008_0000.xml"
```

La validation *Schematron* complémentaire permet de mettre l'erreur en évidence.

```
2024-06-18 13:55:58.8273 [INFO ] Lauching Java Schematron validation process...
>> Filepath = "java.exe"
>> Parameters = -jar "C:\CPP_Samples\schemas\EN17650\schematron\SaxonHE12-4J\saxon-he-12.4.jar" -s:"C:\CPP_Samples\MONSIEUR-N\audiovisualPackage_c2ec52ea-1534-4f16-bde6-0af647749008\metadata\techMD_c2ec52ea-1534-4f16-bde6-0af647749008_0000.xml" -xsl:"C:\CPP_Samples\schemas\EN17650\Schematron\audiovisualSubpackage_techMD_rules.xsl" -o:"C:\CPP_Samples\output\schematronReports\techMD_c2ec52ea-1534-4f16-bde6-0af647749008_0000-package-ebucore.xml"
2024-06-18 13:56:00.5173 [INFO ] Java : Process terminé (ExitCode = 0)
2024-06-18 13:56:00.5173 [ERROR] EN17650 - CppPackageEBUCore schematron validation failed for "C:\CPP_Samples\MONSIEUR-N\audiovisualPackage_c2ec52ea-1534-4f16-bde6-0af647749008\metadata\techMD_c2ec52ea-1534-4f16-bde6-0af647749008_0000.xml"
> schematron - 1 failed Asserts or Warnings detected :
Pattern : name      =
Pattern : id        =
Pattern : documents = file:/C:/CPP_Samples/MONSIEUR-N/audiovisualPackage_c2ec52ea-1534-4f16-bde6-0af647749008/metadata/techMD_c2ec52ea-1534-4f16-bde6-0af647749008_0000.xml
Rule : context      = /ebucore:ebuCoreMain/ebucore:coreMetadata
Assert : location   = /Q{urn:ebu:metadata-schema:ebucore}ebuCoreMain[1]/Q{urn:ebu:metadata-schema:ebucore}coreMetadata[1]
Assert : test       = count(ebucore:format[@formatName='containerFormat']) = 1
Assert : text       = Exactly one Container Format element within the coreMetadata is required (cf. Table 55).
```

Nous avons alors, pendant plusieurs semaines, itéré autour de ces schémas que nous intégrions progressivement au fur et à mesure qu'ils s'enrichissaient.

Les étapes nécessaires

Nous avons donc réalisé des Cinema Preservation Packages à partir des assets et des métadonnées de la plateforme mediaspot®. Cette démarche a permis de valider la faisabilité de la génération de CPP dans un environnement de production « réel ».
Le prototypage s'est effectué en quatre étapes :

1. Collecte des schémas XML référencés dans la norme : les fichiers XSD publiés et accessibles sur internet permettent de « construire » la structure d'un CPP.

2. Création automatique des classes C# correspondantes

3. Codage de la création des CPP : utilisation des classes générées pour créer les CPP, en suivant les schémas METS, EBUCore, PREMIS et les spécificités additionnelles de la norme EN 17650.

4. Validation des CPP générés : assurer la conformité des XML avec les schémas XSD pour garantir la validité technique des paquetages + les XML Schematron.

Une prochaine étape consistera à intégrer l'ensemble de ces schémas de validation XML dans un seul et même outil « open-source » (à l'image de ce qu'a fait Netflix avec *Photon* pour l'IMF)

Validation de la faisabilité

La fabrication d'un CPP répond à un besoin critique de préservation et de conservation à long terme des œuvres numériques. Pour un laboratoire, intégrer le CPP représente une évolution naturelle des services et répond à l'évolution des besoins des ayants-droits. Nous sommes convaincus que ce format contribuera à la normalisation des échanges entre les acteurs de notre marché, facilitant la préservation des œuvres audiovisuels.
Cette étude a permis de démontrer la faisabilité d'une mise en pratique concrète de la norme, à partir d'un environnement de production réel, à la fois pour la génération d'un paquetage CPP et pour la vérification d'un paquetage CPP extérieur.

Enfin, les ajustements restants concerneront également la sélection et le choix des éléments en eux-mêmes pour le contenu du paquetage et des sous-paquetages. Ce choix pouvant se faire par l'ayant-droit directement, ou bien en co-construction avec un expert au sein d'une structure spécialisée dans la préservation.

Propositions d'améliorations de la norme

Quelques incohérences ont été constatées dans le texte de la norme lors de la mise en œuvre, on peut en déduire des retours à conserver pour une prise en compte lors d'une prochaine révision de la norme.

Pour conserver le rapport lisible, tous les cas relevés n'ont pas été résumés ici, une liste complète des erreurs découvertes figure en annexe.

Fort de cette liste, la CST pourra proposer à la commission CN 204 de l'AFNOR de porter ces commentaires et de demander une révision de la norme sans attendre la revue systématique au bout de 5 ans. Quelque soit la décision de calendrier, lors de la prochaine révision de la norme, les commentaires seront alors adressés par le comité CEN TC 457.

Coquilles

Quelques coquilles évidentes ont été relevées, où l'erreur ne fait pas de doute et où la correction est assez directe, en voici quelques exemples :

- au paragraphe 8.3.4.2 l'expression xpath "`/mets:mets/mets:dmdSec/`" ne devrait pas comporter de '/' à la fin.
- au paragraphe 8.4.4.1 le titre fait référence à un élément parent erroné, au lieu de "Constraints on the <mets> root element" il faut lire "Constraints on the <metsHdr> element"
- table 63 la valeur du `typeLink` proposée est "`urn.eu:en17650:2022:ns#componetizedType`" et il faut lire "`urn.eu:en17650:2022:ns#componentizedType`"

Imprécisions

- les éléments `normalPlayTime` et `editUnitNumbers` sont mutuellement exclusifs, et la norme les cite comme obligatoire, ce qui entraîne une confusion avec le schéma EBUCore

Erreurs

Des erreurs un peu plus ennuyeuses se sont glissées dans la rédaction de la première édition de la norme.

En voici quelques exemples :

- dans le répertoire `playlist` du paquetage, il est facultatif d'ajouter des playlists. Pour autant, dans le fichier de métadonnées racine `preservationPackingList.xml`, il est obligatoire d'ajouter au moins une ligne de description `<file>` pour dans la catégorie des fichiers `playlist`, ce qui n'est pas logique et donc probablement une erreur de mise en cohérence.
- un erreur similaire existe pour les fichiers `ancillaryData`
- un élément `<format>` n'est pas listé dans les éléments enfants obligatoires

Désaccords sur les choix faits dans la norme

- la convention de nommage des fichiers n'autorise pas toujours le caractère espace souligné ("_"), ce qui pose des problèmes car il est systématiquement utilisé comme séparateur, ce qui oblige à renommer assez inutilement les fichiers
- la caractérisation d'un sous-élément '`role`' de l'élément '`contributor`', désignant une personne nommée au générique d'un film, est de façon optionnelle '`cast`' ou '`credit`', ce qui rend la validation des listes techniques et artistiques incohérente.

Utilisation des fichiers PREMIS

Le premier exercice a consisté à élaborer le plus court XML PREMIS possible pour ne transporter que l'ancien nom du fichier média.
Ci-dessous un exemple pour transporter le nom original d'un document word.

```
<premis:premis xmlns:premis="http://www.loc.gov/premis/v3"
               xmlns:xlink="http://www.w3.org/1999/xlink"
               xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
                                        xsi:schemaLocation="http://www.loc.gov/premis/v3
https://www.loc.gov/standards/premis/premis.xsd"
               version="3.0">
   <premis:object xsi:type="premis:file">
       <premis:objectIdentifier>
           <premis:objectIdentifierType>local</premis:objectIdentifierType>
           <premis:objectIdentifierValue>file-123</premis:objectIdentifierValue>
       </premis:objectIdentifier>
       <premis:objectCharacteristics>
           <premis:format>
               <premis:formatDesignation>
                   <premis:formatName>Microsoft Word Document</premis:formatName>
               </premis:formatDesignation>
           </premis:format>
       </premis:objectCharacteristics>
       <premis:originalName>original_document.docx</premis:originalName>
   </premis:object>
</premis:premis>
```

Un autre exemple de fichier XML PREMIS a été construit pour modéliser un cas d'usage fictif.
On suit la numérisation d'une seule bobine d'un film composé de plusieurs bobines :

- une bobine du film "Ocean's Eleven"
- capture des détails de la source physique
- du processus de numérisation,
- de l'objet numérique résultant
- et de toutes les parties impliquées dans le processus.

Il fournit un enregistrement complet pour la préservation, le suivi de la provenance et les références futures.

```
<premis:premis xmlns:premis="http://www.loc.gov/premis/v3"
   xmlns:xlink="http://www.w3.org/1999/xlink"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
                                        xsi:schemaLocation="http://www.loc.gov/premis/v3
https://www.loc.gov/standards/premis/premis.xsd"
   version="3.0">
         <premis:object   xsi:type="premis:intellectualEntity">   <!--add   the   object
intellectualEntity type for the film-->
    <!--Please note that adding this object may be redundant in relation to the content
of the descMD.xml file-->
```

```
        <premis:objectIdentifier>
            <premis:objectIdentifierType>local</premis:objectIdentifierType>
            <premis:objectIdentifierValue>ocean11</premis:objectIdentifierValue>
        </premis:objectIdentifier>
        <premis:linkingEventIdentifier>
            <premis:linkingEventIdentifierType>UUID</premis:linkingEventIdentifierType>

<premis:linkingEventIdentifierValue>8c6714b3-ab18-44e2-8507-80691ad1dafa</premis:linking
EventIdentifierValue>
        </premis:linkingEventIdentifier>
    </premis:object>
      <premis:object  xsi:type="premis:representation">    <!--physical  source  object  (a
reel)-->
        <premis:objectIdentifier>
            <premis:objectIdentifierType>local</premis:objectIdentifierType>
            <premis:objectIdentifierValue>ocean11-03A</premis:objectIdentifierValue>
        </premis:objectIdentifier>
        <premis:significantProperties>
            <premis:significantPropertiesType>Base</premis:significantPropertiesType>

<premis:significantPropertiesValue>triacetate</premis:significantPropertiesValue>
        </premis:significantProperties>
        <premis:significantProperties>
            <premis:significantPropertiesType>optical</premis:significantPropertiesType>

<premis:significantPropertiesValue>digital</premis:significantPropertiesValue>
        </premis:significantProperties>
        <premis:significantProperties>
            <premis:significantPropertiesType>length</premis:significantPropertiesType>
            <premis:significantPropertiesValue>300</premis:significantPropertiesValue>
        </premis:significantProperties>
        <premis:significantProperties>
                                                <premis:significantPropertiesType>length
unit</premis:significantPropertiesType>
            <premis:significantPropertiesValue>meter</premis:significantPropertiesValue>
        </premis:significantProperties>
        <premis:significantProperties>
                                            <premis:significantPropertiesType>number    of
reels</premis:significantPropertiesType>
            <premis:significantPropertiesValue>12</premis:significantPropertiesValue>
        </premis:significantProperties>
        <premis:linkingEventIdentifier>
            <premis:linkingEventIdentifierType>UUID</premis:linkingEventIdentifierType>

<premis:linkingEventIdentifierValue>8c6714b3-ab18-44e2-8507-80691ad1dafa</premis:linking
EventIdentifierValue>
        </premis:linkingEventIdentifier>
    </premis:object>
    <premis:object xsi:type="premis:representation"> <!--digitalized object -->
        <premis:objectIdentifier>

<premis:objectIdentifierType>urn:cen.eu:en17650:2022:ns:metadata#resourceId</premis:obje
ctIdentifierType>

<premis:objectIdentifierValue>imagePackage_6aac082c-008f-4827-becb-a4697d2383a7</premis:
```

```
objectIdentifierValue>
       </premis:objectIdentifier>
       <premis:relationship>
           <premis:relationshipType>representation</premis:relationshipType>
           <premis:relationshipSubType>digital</premis:relationshipSubType>
           <premis:relatedObjectIdentifier>

<premis:relatedObjectIdentifierType>local</premis:relatedObjectIdentifierType>

<premis:relatedObjectIdentifierValue>ocean11-03A</premis:relatedObjectIdentifierValue>
           </premis:relatedObjectIdentifier>
       </premis:relationship>
       <premis:linkingEventIdentifier>
           <premis:linkingEventIdentifierType>UUID</premis:linkingEventIdentifierType>

<premis:linkingEventIdentifierValue>8c6714b3-ab18-44e2-8507-80691ad1dafa</premis:linking
EventIdentifierValue>
       </premis:linkingEventIdentifier>
   </premis:object>
   <premis:event>
       <premis:eventIdentifier>
           <premis:eventIdentifierType>UUID</premis:eventIdentifierType>

<premis:eventIdentifierValue>8c6714b3-ab18-44e2-8507-80691ad1dafa</premis:eventIdentifie
rValue>
       </premis:eventIdentifier>
       <premis:eventType>digitization</premis:eventType>
       <premis:eventDateTime>2019-11-14T11:00:50.632264+00:00</premis:eventDateTime>
       <premis:eventDetailInformation>
                                   <premis:eventDetail>Film   reel   digitization   using
Scan-proZ39</premis:eventDetail>
       </premis:eventDetailInformation>
       <premis:eventOutcomeInformation>
           <premis:eventOutcome>success</premis:eventOutcome>
           <premis:eventOutcomeDetail>
                     <premis:eventOutcomeDetailNote>Scan completed successfully with no
visible artifacts or quality issues</premis:eventOutcomeDetailNote>
           </premis:eventOutcomeDetail>
       </premis:eventOutcomeInformation>
       <premis:linkingAgentIdentifier>
           <premis:linkingAgentIdentifierType>local</premis:linkingAgentIdentifierType>

<premis:linkingAgentIdentifierValue>scan-01</premis:linkingAgentIdentifierValue>
       </premis:linkingAgentIdentifier>
       <premis:linkingAgentIdentifier>
           <premis:linkingAgentIdentifierType>local</premis:linkingAgentIdentifierType>

<premis:linkingAgentIdentifierValue>lab-01-VDM</premis:linkingAgentIdentifierValue>
       </premis:linkingAgentIdentifier>
       <premis:linkingAgentIdentifier>
           <premis:linkingAgentIdentifierType>local</premis:linkingAgentIdentifierType>

<premis:linkingAgentIdentifierValue>mediaSpot-user-01</premis:linkingAgentIdentifierValu
e>
           <premis:linkingAgentRole>operator</premis:linkingAgentRole>
```

```
        </premis:linkingAgentIdentifier>
    </premis:event>
    <premis:agent>
        <premis:agentIdentifier>
            <premis:agentIdentifierType>local</premis:agentIdentifierType>
            <premis:agentIdentifierValue>scan-01</premis:agentIdentifierValue>
        </premis:agentIdentifier>
        <premis:agentName>Scan-proZ39</premis:agentName>
        <premis:agentType>software</premis:agentType>
        <premis:agentVersion>Scan-proZ39.1.2</premis:agentVersion>
    </premis:agent>
    <premis:agent>
        <premis:agentIdentifier>
            <premis:agentIdentifierType>local</premis:agentIdentifierType>
            <premis:agentIdentifierValue>lab-01-VDM</premis:agentIdentifierValue>
        </premis:agentIdentifier>
        <premis:agentName>VDM Laboratory</premis:agentName>
        <premis:agentType>organization</premis:agentType>
    </premis:agent>
    <premis:agent>
        <premis:agentIdentifier>
            <premis:agentIdentifierType>local</premis:agentIdentifierType>
            <premis:agentIdentifierValue>mediaSpot-user-01</premis:agentIdentifierValue>
        </premis:agentIdentifier>
        <premis:agentName>John Doe</premis:agentName>
        <premis:agentType>person</premis:agentType>
        <premis:agentNote>Scanning technician</premis:agentNote>
    </premis:agent>
</premis:premis>
```

Conclusion

Le premier objectif du projet était de simplifier la mise en œuvre de la norme et il nous semble que la mise à disposition des schémas *Schematron* remplit une première étape importante. Nous avons pu vérifier par la mise en pratique côté laboratoire, que les schémas aidaient au développement de l'application de composants logiciels permettant d'exporter des packages au format CPP.

Les travaux mettent également en évidence l'utilisabilité du format défini dans la norme, même si quelques ajustements sont nécessaires. Le fait de devoir éprouver un nouveau texte est normal et nous pouvons nous réjouir de contribuer à en augmenter la maturité.

Des commentaires sont prêts pour une révision. Nous pouvons les porter à l'AFNOR pour déclencher une révision, ou attendre d'autres commentaires de la part des partenaires historiques du projet.

Une des démarches laissées en suspens au terme de la rédaction initiale de la norme était de traduire la norme et ses évolutions par un registre des métadonnées nécessaires et de leurs valeurs possibles pour décrire les différents aspects techniques et descriptifs couverts par la norme. Un tel registre pourrait être opéré par la CST, avec le soutien du CNC.

La concomitance de ce travail avec la révision de la norme CST-RT-043 a pu montrer une large convergence des besoins d'échange des métadonnées et de la standardisation des échanges. Le projet vient donc confirmer une démarche de standardisation des modèles de métadonnées au cœur des démarches de mise en place de dataspaces sectoriels, portés par la CST avec le projet TAMIS et sa participation au consortium TEMS.

Enfin, nous appelons de nos vœux la mise à disposition de solutions logicielles par les acteurs du marché. Nous sommes conscient qu'une des difficulté est de devoir à la fois maîtriser des aspects de mastering et la disponibilité des métadonnées. Il nous semble, là aussi, que la mise en place d'API standardisées pourrait simplifier la mise en place de solutions.

Ressources

CaPPuSCHino	https://gitlab.com/cst-cpp/cappuschino
Exemples XML	A venir

Références

[EN 17650:2022]	Un cadre pour la conservation numérique des œuvres cinématographiques - Le paquetage de conservation cinéma
[EN 15744:2009]	Film identification — Minimum set of metadata for cinematographic works
[EN 15907:2010]	Film identification - Enhancing interoperability of metadata - Element sets and structures
[ISO/IEC 19757-3:2020]	Information technology - Document Schema Definition Languages (DSDL) - Part 3: Rule-based validation using Schematron
schxslt	https://github.com/schxslt/schxslt

Annexe 1 : liste des amendements à proposer pour la révision de la norme

MB/NC	Line number	Clause/Subclause	Paragraph/ Figure/ Table/	Type of comment	Comments	Proposed change	Observations of the secretariat
AFNOR	1	8.3.4.2		ed	"/mets:mets/mets:dmdSec/" There is an error within the XPath expression. The final backslash shall be removed to correctly target the element.	"/mets:mets/mets:dmdSec"	
AFNOR	1 and 2		Table 9	te	Ancillary files arity=1 Playlist files arity=1 The standard erroneously requires the presence of Ancillary and Playlist files in fileSec by setting their arity to 1. These elements shall be optional.	Ancillary files arity=0..1 Playlist files arity=0..1	
AFNOR	1, 2, and 4		Table 11	te	Ancillary data arity=1 Playlist arity=1 Checker reports arity=1 The standard erroneously requires the presence of Ancillary data, Playlist, and Checker reports by setting their arity to 1. These elements shall be optional as the presence of target files is itself optional.	Ancillary data arity=0..1 Playlist arity=0..1 Checker reports arity=0..1	
AFNOR		8.4.4.1		ed	Constraints on the <mets> root element The title of the section contains an error. The attributes packingListKind, packageProfile, and packageFormatVersion are within <metsHdr>, not the <mets> root element.	Constraints on the <metsHdr> element	
AFNOR	1	8.4.4.8		ed	XPath is: "/mets:mets/mets:structMap/mets:div[@TYPE="ancillaryData"]" The XPath provided for ancillary data div is incorrect due to a <mets:div> missing element.	XPath is: "/mets:mets/mets:structMap/mets:div/mets:div[@TYPE="ancillaryData"]"	
AFNOR	1	8.4.4.9		ed	XPath is: "/mets:mets/mets:structMap/mets:div[@TYPE="data"]" The XPath provided for data div is incorrect due to a <mets:div> missing element.	XPath is: "/mets:mets/mets:structMap/mets:div/mets:div[@TYPE="data"]"	
AFNOR	1		Table 41	te	The <format> element EBUCore is missing for Image Package Format.	Add the <format> element EBUCore for Image Package Format.	
AFNOR	2 and 4		Table 48	te	normalPlayTime and editUnitNumber are both mandatory according to Table 48 The normalPlayTime and editUnitNumber sub-elements are erroneously listed with arity=1 in Table 48, leading to confusion.	It should be explicitly stated below the table that according to EBUCore, either normalPlayTime or editUnitNumber is mandatory, not both.	
AFNOR	1		Table 63	te	"urn.eu:en17650:2022:ns #componetizedType" The typeLink attribute of the ebucore:technicalAttributeString element is incorrectly specified.	Correct the typeLink attribute value of ebucore:technicalAttributeString to: "urn.eu:en17650:2022:ns #componentizedType".	

AFNOR		6.3.2		ge	Root folder name According to the standard, the use of underscores (_) in folder naming is prohibited, which is illogical.	Update the folder naming convention to allow underscores (_).	
AFNOR			Table 36	te	the values ‘credits’ and ‘cast’ for the typeLabel attribute in contributor/role sub element currently are optional values	the values ‘credits’ and ‘cast’ for the typeLabel attribute in contributor/role shall be mandatory to facilitate the implementation of the validation constraint	