INSA de Lyon
20, Avenue Albert Einstein
69100 Villeurbanne - France

Numéro de Document : **L-011-02**

Version : **2**

ÉLABORATION D'UN CAHIER DES CHARGES POUR LA DEFINITION D'UN MODELE DE DOCUMENT POUR LES THESES NUMERIQUES DE DOC'INSA (CITHER2002)

-

RAPPORT DE FIN DE PFE

Responsable du projet : **Doc'INSA**

Département responsable du projet : **IF**

Préparé, vérifié et approuvé par		**Signature**	**Date**
Stagiaire PFE	M. Ioannitis Sébastien	_____	__ / __ / __
Directrice Doc'INSA	Mme Joly Monique	_____	__ / __ / __
Responsable PFE	Mme Rumpler Béatrice	_____	__ / __ / __
Employé Doc'INSA	Mlle Boudia Dalila	_____	__ / __ / __
Employé Doc'INSA	M. Brochet Gilles	_____	__ / __ / __

Tous droits réservés. Les informations contenues dans ce document sont la propriété de l'INSA de Lyon et ce dernier est fourni sans l'assurance qu'il ne contienne d'erreurs ou d'oublis. En sus, son contenu, même partiel, ne peut être reproduit, utilisé ou divulgué sauf sous l'accord de ses propriétaires, lequel accord faisant l'objet d'une autorisation. Ces restrictions s'étendent à tous types de médias comprenant notamment les supports magnétiques et électroniques.

INFORMATIONS DE CONTROLE DU DOCUMENT

N°	Date	Auteur	Description
1.	14-sept-02		Création du document.
2.	14-sept-02	M. Ioannitis Sébastien	Rédaction
3.	15-juin-02	M. Ioannitis Sébastien	Rédaction
4.	17-juin-02	M. Ioannitis Sébastien	Rédaction
5.	18-juin-02	M. Ioannitis Sébastien	Rédaction
6.	19-juin-02	M. Ioannitis Sébastien	Rédaction Préparation pour le pack 1
7.			

Tableau 1 – Informations de contrôle du document

Table des matières

Liste des figures

Liste des tableaux

1. OBJET DU DOCUMENT

1.1 IDENTIFICATION DU DOCUMENT

Ce document prend part dans le projet « Élaboration d'un cahier des charges pour la définition d'un modèle de document pour les thèses numériques de Doc'INSA », celui-ci étant un sous-projet CITHER (Consultation en texte Intégral des THèses en Réseau) de Doc'INSA.

Ce document offre une vue synthétique de tous les livrables produits lors de ce projet. Il présente d'abord le cahier des charges dans ses généralités. Il décrit ensuite les recommandations/standards ainsi que les outils utilisés, et expose certains choix effectués pour ou contre certains d'entre eux, notamment concernant la DTD utilisée pour l'archivage des thèses. Enfin, il décrit la chaîne de traitements.

Étant donné que le prototype de ladite chaîne de traitements est en cours de développement et que le PFE a été prolongé afin de terminer les développements entrepris, la documentation concernant ce prototype sera fournie ultérieurement, certains choix techniques n'étant pas encore définitifs ou les développements n'étant pas encore assez avancés.

1.2 DESCRIPTION DU SYSTEME

Le but du système peut se résumer en deux points. D'une part il permet aux doctorants de leur apporter un cadre structuré pour la rédaction de leur travail de thèse au travers de modèles de documents, d'un outil pour l'envoi de celui-ci à Doc'INSA ainsi que d'une bonne assistance. D'autre part, il permet de prendre en considération de nouveaux besoins en termes d'archivage, en ce sens que tous les documents doivent être transformés puis stockés dans un format d'archivage pivot facilitant non seulement leur archivage à long terme, mais aussi leur restitution sous un autre format dans le but d'être consultés au travers du web.

2. PRESENTATION DU CAHIER DES CHARGES

Dans cette partie, nous présentons le cahier des charges dans ses grandes lignes en décrivant tous les paquetages des cas d'utilisation définis et en exposant brièvement le but des cas d'utilisation qui les composent. Donnons tous d'abord les acteurs interagissant avec le système.

2.1 ACTEURS INTERNES À DOC'INSA

1. Éditeur/archiviste : cet acteur correspond à la personne en charge de la réception des thèses, de leur traitement, et de leur archivage, ceci dans le but de les rendre disponibles aux internautes ;

2. Site web : cet acteur modélise le site web de Doc'INSA mettant à disposition des internautes les travaux de thèse sous leur forme numérique, ceux-ci ayant été réalisés par les doctorants ;

3. Extranet : cet acteur modélise l'interface entre le doctorant et l'éditeur/archiviste. Cet acteur prend son rôle dans le cadre de la soumission du travail de thèse du doctorant à Doc'INSA et des informations associées.

2.2 ACTEURS EXTERNES À DOC'INSA

1. Doctorant : cet acteur correspond à la personne produisant un travail de thèse dans le but d'obtenir le grade de Docteur. Cet acteur rédige sa thèse et la soumet avant la soutenance à l'acteur éditeur/archiviste ;

2. Internaute : cet acteur correspond à toute personne consultant les thèses en ligne disponibles sur le site de Doc'INSA. Un internaute peut être un engeignant, un chercheur, un étudiant ou toute autre personne ayant accès à Internet, ceci indépendamment de sa localisation ;

3. Département des Études Doctorales (DED) : cet acteur correspond à « l'entité » supervisant le travail de thèse du doctorant. C'est lui qui émet l'avis de diffusion de la thèse.

2.3 DESCRIPTION DES CAS D'UTILISATION

À chaque paquetage correspond un diagramme décrivant les cas d'utilisation qu'il comprend ainsi que les acteurs mis en jeu.

2.3.1 Paquetage 1 - Composition

Ce paquetage est un regroupement logique de cas d'utilisation permettant au doctorant de disposer de toutes les facilités nécessaires pour la rédaction et la soumission de sa thèse, en sus d'une aide qui peut lui être apportée.

2.3.1.a) DIAGRAMME DES CAS D'UTILISATION

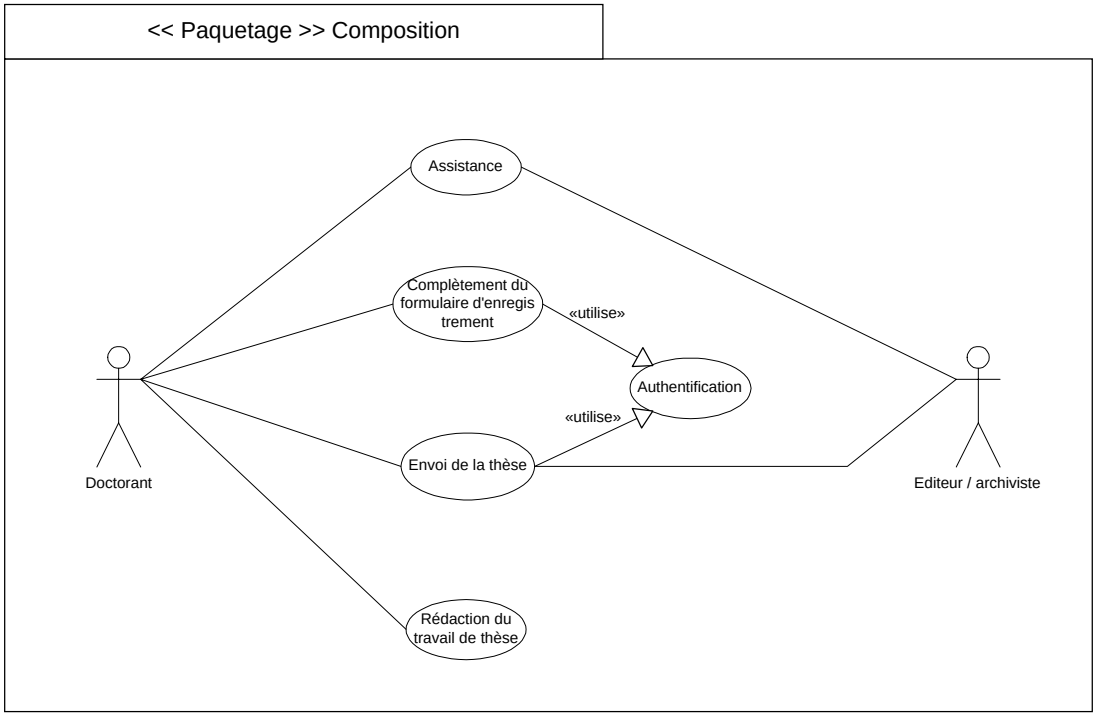

Figure 1 – Paquetage des cas d'utilisation *Composition*

2.3.1.b) *CAS D'UTILISATION 1*- GESTION DES AUTHENTIFICATIONS

But : permettre au doctorant de se connecter à l'extranet *via* une connexion sécurisée.

Acteurs :

- *principal* : doctorant
- *secondaires* : site web

2.3.1.c) CAS D'UTILISATION 2 - RENSEIGNEMENT DU FORMULAIRE D'ENREGISTREMENT

But : permettre au doctorant de remplir en ligne le formulaire d'enregistrement. Notons que jusqu'à présent, ledit formulaire d'enregistrement était sur support papier. Ainsi offre-t-on ici un moyen en ligne de remplir le formulaire d'enregistrement. Le but est aussi d'offrir à l'éditeur/archiviste la possibilité de récupérer des métadonnées facilement, lui faisant gagner un temps de saisie.

Acteurs :
- *principal* : doctorant
- *secondaires* : Extranet

2.3.1.d) CAS D'UTILISATION 3 - ENVOI DU TRAVAIL DE THÈSE

But : permettre au doctorant de soumettre sa thèse en ligne (i.e. sans qu'il se déplace à Doc'INSA). Il dispose pour ce faire d'un moyen par défaut, et d'autres moyens qui permettent d'obvier aux éventuelles défaillances de ce premier.

Acteurs :
- *principal* : doctorant
- *secondaires* : site web

2.3.1.e) CAS D'UTILISATION 4 - AIDE

But : offrir aux doctorants toute l'aide nécessaire leur permettant d'utiliser les outils mis à leur disposition pour la composition de leur thèse. Cette aide sera disponible sur le site Internet de Doc'INSA.

Acteurs :
- *principal* : doctorant
- *secondaires* : site web, éditeur/archiviste

2.3.1.f) CAS D'UTILISATION 4 - RÉDACTION

Nous avons, ici, rajouté un cas d'utilisation *Rédaction*, dans un souci de cohérence en ce qui concerne l'enchaînement des activités lors de la phase de composition d'un travail de thèse. Ce cas d'utilisation ne prend pas de légitimité dans le contexte du système.

2.3.2 Paquetage 2 - Traitement/archivage

Ce paquetage est un regroupement logique de cas d'utilisation permettant le traitement des thèses envoyées par les doctorants ainsi que leur archivage dans un format pivot, en l'occurrence le XML.

2.3.2.a) DIAGRAMME DES CAS D'UTILISATION

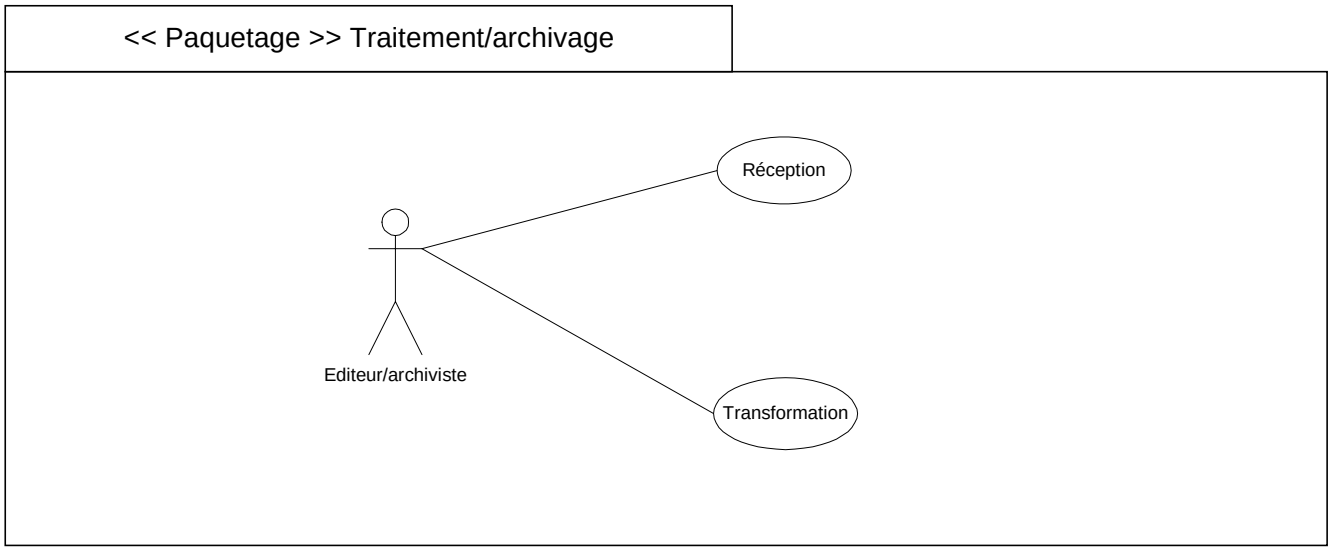

Figure 2 – Paquetage des cas d'utilisation *Traitement/archivage*

2.3.2.b) *CAS D'UTILISATION 1* - RÉCEPTION

But : permettre le stockage temporaire des thèses reçues dans une zone tampon et prévenir l'éditeur/archiviste de leur réception.

Acteurs :

- *principal* : système
- *secondaires* : éditeur/archiviste

2.3.2.c) *CAS D'UTILISATION 2* - TRANSFORMATION

But : convertir un travail de thèse reçu au format Word ou LaTex dans un format pivot à base de XML. Ce cas d'utilisation doit prendre en considération les évolutions futures du système, en ce qu'il devra accepter d'autres types de formats. À la fin de chaque enchaînement un point de reprise est défini, afin qu'il soit possible de reprendre les traitements à partir d'un point de reprise antérieur au dernier. De plus, chaque enchaînement peut être exécuté par lui-même, mais ceci devrait être exceptionnel.

Acteurs :

- *principal* : éditeur/archiviste
- *secondaires* : système

2.3.2.c - i) DIAGRAMME DE BLOCS DE LA CHAÎNE DE TRAITEMENTS

Le diagramme ci-dessous représente les traitements que la chaîne de traitements devra réaliser pour transformer une thèse en entrée avec ses objets multimédias, puis archiver le résultat de la transformation, et enfin pour pouvoir restituer le travail de thèse pour la consultation par un internaute.

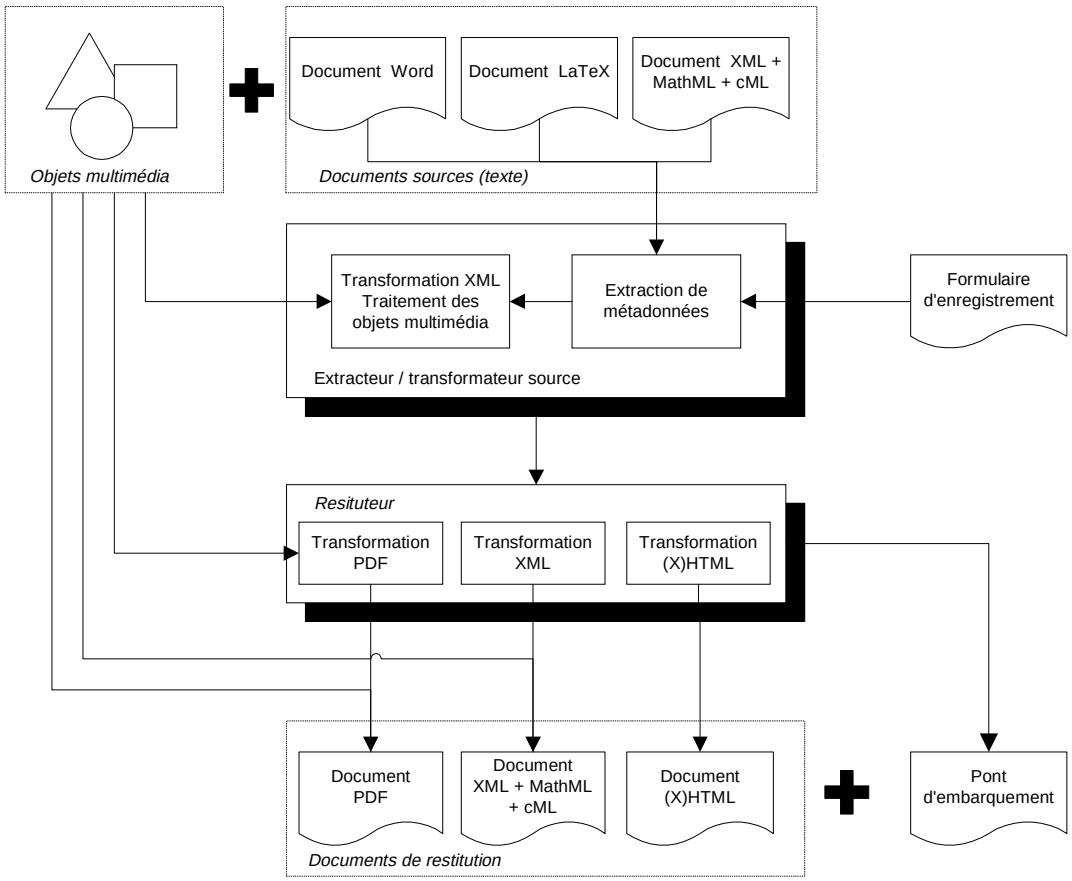

Figure 3 – Diagramme de blocs de la chaîne de traitements

2.3.3 Paquetage 3 - Restitution

Ce paquetage est un regroupement logique de cas d'utilisation permettant à l'internaute de consulter la thèse stockée dans un des formats générés et conservés du côté serveur.

2.3.3.a) DIAGRAMME DES CAS D'UTILISATION

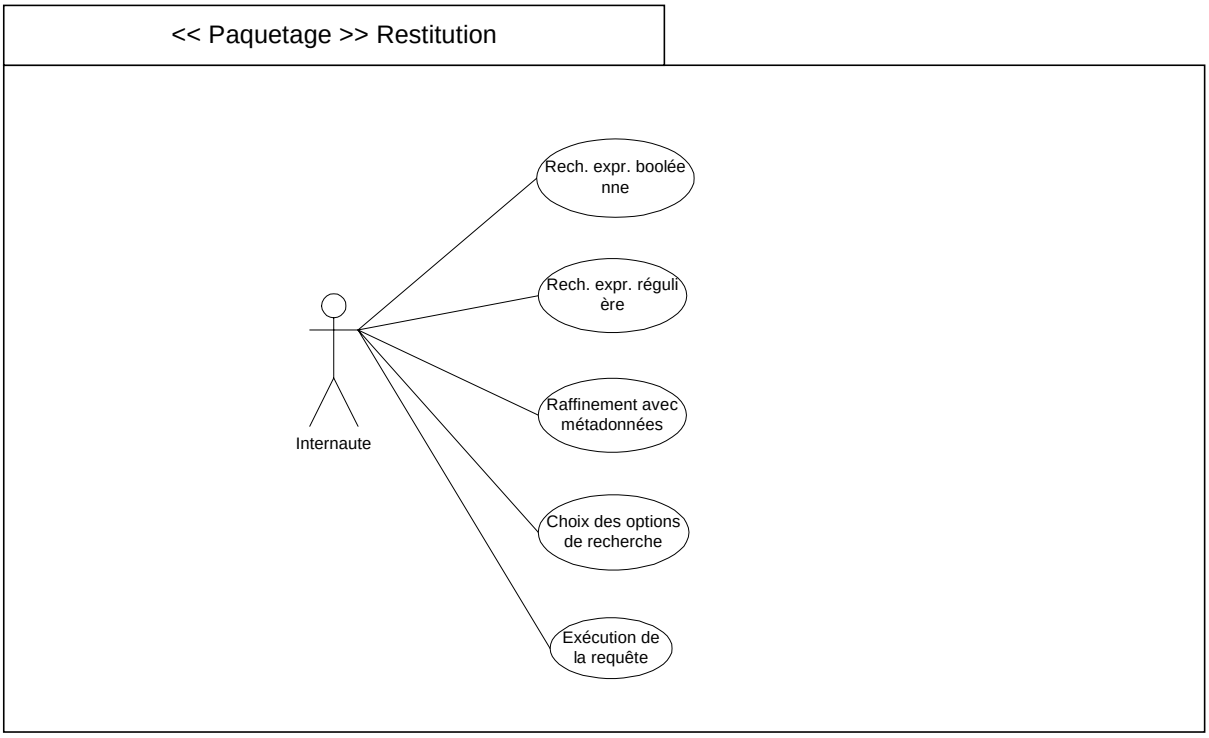

Figure 4 – Paquetage des cas d'utilisation *Restitution*

2.3.3.b) CAS D'UTILISATION 1 - RECHERCHE

But : permettre à l'internaute de rechercher à partir du site de Doc'INSA un travail de thèse correspondant à une requête

Acteurs :

- *principal* : internaute
- *secondaires* : système

2.3.3.c) *CAS D'UTILISATION 2* - CONSULTATION

But : permettre à l'internaute de consulter un travail de thèse dans son entièreté ou une partie de celui-ci suite à une requête qu'il a émise.

Acteurs :
- *principal* : internaute
- *secondaires* : système

2.3.3.d) CAS D'UTILISATION 3 - AIDE

But : offrir aux internautes toute l'aide nécessaire leur permettant d'effectuer des recherches, de consulter un document de thèse ainsi que les contenus multimédias associés. Cette aide fournit également tous les éléments relatifs aux droits d'auteur.

Acteurs :
- *principal* : doctorant
- *secondaires* : site web, éditeur/archiviste

2.3.4 Paquetage 4 - Statistiques

Ce paquetage est un regroupement logique de cas d'utilisation collectant des statistiques et de là permettant de générer des rapports statistiques. Ces statistiques peuvent servir aussi bien pour Doc'INSA.

2.3.4.a) DIAGRAMME DES CAS D'UTILISATION

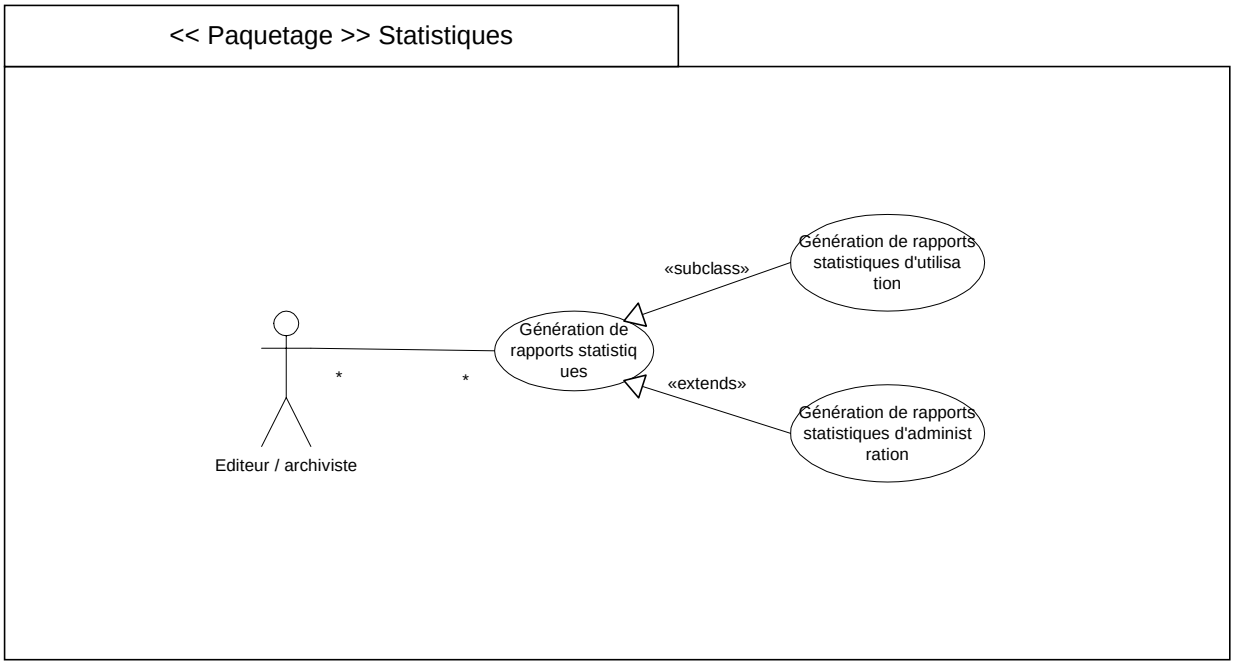

Figure 5 – Paquetage des cas d'utilisation *Statistiques*

2.3.4.b) *CAS D'UTILISATION 1*- CONSULTATION DES STATISTIQUES

But : permettre à Doc'INSA d'avoir des statistiques d'utilisation (i.e. concernant l'utilisation des thèses empruntées ; ex : nombre de thèses empruntées), mais également des statistiques d'administration (ex : nombre de connexion par heure).

Acteurs :

- *principal* : éditeur/archiviste
- *secondaires* : site web

3. ELEMENTS CONSTITUTIFS D'UNE THESE

Les éléments suivants ont été définis comme pouvant faire partie d'une thèse, et pouvant être discernés du contenu normal de la thèse. Les éléments qui doivent obligatoirement figurer dans la thèse sont précédés d'un *tick* (✓), les autres n'étant précédés d'aucun symbole.

3.1 PAGE DE TITRE

✓	Auteur (nom, prénom)	✓	Formation doctorale		Président du Jury
✓	Copyright	✓	Grade	✓	Qualité
✓	Date de soutenance	✓	Laboratoire	✓	Titre de la thèse
✓	Directeur de thèse	✓	Membres du Jury : personnes, fonction	✓	Type de doctorat[1]
✓	Discipline			✓	Université de soutenance
✓	École doctorale	✓	N° d'ordre		

Tableau 2 – Éléments constitutifs de la page de titre

3.2 LIMINAIRES

	Dédicace		Liste des professeurs		Remerciements
	Errata		Liste des tableaux	✓	Résumé anglais
	Liste des écoles doctorales	✓	Mots-clefs français	✓	Résumé français
	Liste des équations	✓	Mots-clefs anglais		Table des matières
	Liste des figures		Préface		

Tableau 3 – Éléments constitutifs des liminaires

3.3 CORPS

	Chapitre		Légende de tableau		Note de fin de chapitre
	Citation d'introduction		Légende d'objet multimédia		Note de fin de partie
	Citation de conclusion		Légende/numéro d'équation		Objet multimédia
	Citation de partie		Lien hypertexte		Paragraphe
	Conclusion		Liste à puce		Partie
	Équation		Liste numérotée		Tableau
	Introduction		Note de bas de page		
	Légende de figure		Note de document		

Tableau 4 – Éléments constitutifs du corps de la thèse

3.4 ANNEXES

	Addenda / postface		Glossaire général		Index des noms propres
	Appendice / annexe		Glossaire technique		Description des objets multimédias attachés
	Bibliographie		Index général		
	Glossaire des noms propres		Index des termes		

Tableau 5 – Éléments constitutifs des annexes

[1] Dans le cas de l'INSA le type de doctorat est la thèse doctorat.

4. RECOMMANDATIONS/STANDARDS UTILISES

4.1 XML

XML est un format de représentation structurée de l'information défini par le W3C[2], nonobstant son appellation « langagièrement » abusive et messéante de *langage*. XML est souvent qualifié de métalangage (i.e. un langage permettant de définir des langages). Plus qu'un format, nous préférons le considérer comme un formalisme, un ensemble de règles ou de conventions qu'il convient de respecter pour structurer un document correctement. Les technologies élaborées autour de ce format permettent notamment de réaliser la dichotomie entre le contenu et la présentation (i.e. la mise en forme).

4.2 XSL

XSL (XSLT, XPath, XSL-FO) est un langage d'expression de feuilles de style défini par le W3C. Il comprend trois parties : un langage pour transformer des documents XML (i.e. XSLT) ; un langage d'expressions utilisé par XSLT pour accéder ou référencer des parties d'un document XML (i.e. XPath) ; et, un vocabulaire XML spécifiant la sémantique de mise en forme (i.e. XSL-FO). C'est grâce à XSL que peuvent être réalisées différentes présentations d'un document XML, lesquelles peuvent être destinées à l'impression (ex. : PDF), à la consultation sur la toile (ex. : HTML) ou à d'autres usages (ex. : VoiceML pour la restitution vocale). En parlant de XSLT, nous incluons de fait XPath qui lui est « indissociable ».

Nous avons utilisé XSLT pour transformer un document XML-UpCast en DocBook, et XSLT et XSL-FO pour représenter la page de titre de la thèse.

4.3 MATHML

MathML est une recommandation définie par le W3C. Elle permet de représenter des formules mathématiques en XML. Ainsi les formules mathématiques écrites avec MS Word ou MathType peuvent être représentées en MathML.

[2] Le W3C a été fondé pour mener la toile à son potentiel maximal en développant les protocoles communs qui favorisent son évolution et assurent son interopérabilité.

4.4 RDF

RDF est une recommandation définie par le W3C. Elle fournit un vocabulaire standard pour représenter les métadonnées en XML (cf. Dublin Core). RDF permet l'interopérabilité entre les applications qui échangent des informations.

4.5 DUBLIN CORE

Dublin Core est un vocabulaire définissant des éléments de métadonnées pour la description de ressources. Il est à l'initiative du DCMI. Utilisé conjointement avec RDF, il est possible d'offrir à ces métadonnées une structure facilitant l'extraction de ces données. Ainsi avons-nous utilisé RDF et Dublin Core pour représenter les métadonnées liées à une thèse et à son auteur.

4.6 DOCBOOK

DocBook est un système d'écriture (i.e. une DTD) de documents structurés en SGML ou en XML dont la maintenance et l'évolution ont été déléguées au consortium OASIS. C'est la DTD que nous avons utilisée pour la représentation des thèses dans un format pivot servant à l'archivage. Nous justifierons le choix de cette DTD dans la Partie 7, *Choix de la DTD*.

5. OUTILS UTILISES

Les outils intégrés à la chaîne de traitements appartiennent en partie au logiciel libre (*open source* en anglais), hormis UpCast, MathType, MS Word et Gemini Solo. Notons aussi que XML Spy n'est pas libre de droits mais qu'il n'intervient pas dans la chaîne de traitements. Aussi peut-il être substitué par un autre éditeur XML. L'astérisque placé en exposant à la droite d'un outil **(*)** signifie que ce dernier est libre de droits.

5.1 XML SPY

XML Spy est un atelier de génie logiciel XML développé par Altova. Nous l'avons utilisé pour la programmation XML (XSL, DocBook, RDF). Très complet, il dispose d'outils d'aide à la programmation, comme un moteur d'expressions XPath. Comme nous l'avons dit plus haut XML Spy n'intervient pas dans la chaîne de traitements et de fait il peut être substitué par un autre environnement comme Xeena d'IBM qui est libre de droits.

5.2 SAXON(*)

Saxon a été développé par Michael KAY[3] de Software AG et est libre de droits. Cet outil est ce que l'on appelle un *processeur XSLT* permettant d'extraire et de restructurer des informations présentées dans le format XML. De plus, Saxon intègre un *parser XML*. Ce dernier vérifie qu'un document XML est bien structuré (i.e. si le code produit respecte le format XML). Par ailleurs, il est qualifié de *validant* car il vérifie si un document XML est conforme au schéma (ex. : DTD, XML Schéma) auquel il est lié. Saxon a été choisi pour sa bonne conformité aux recommandations du W3C.

Nous justifierons son choix dans la partie 8, *Choix des outils de la chaîne de traitements*.

5.3 FOP(*)

FOP est un outil prenant part au le projet *Apache XML* et est libre de droits. À partir d'un document XSL-FO, il permet de produire un document dans d'autres formats (ex. : PDF, HTML, RTF).

Nous justifierons son choix dans la partie 8, *Choix des outils de la chaîne de traitements*.

[3] Michael KAY est l'auteur d'un ouvrage de référence sur XSLT.

5.4 MS WORD ET VBA

MS Word est le logiciel de traitement de texte le plus utilisé au monde. Il est développé par Microsoft. Dans 95% des cas, les thèses de l'INSA de Lyon sont rédigées avec MS Word. Nous l'avons donc utilisé pour la création d'un modèle de document afin d'aider le doctorant dans son travail de composition. Ce modèle a été amélioré au moyen de macros VBA, langage intégré à MS Word. Nous avons aussi utilisé ce langage pour l'extraction des équations mathématiques et leur conversion en MathML (ceci en conjonction avec MathType), et pour la conversion d'un document MS Word en RTF.

5.5 MATHTYPE

MathType est un éditeur d'équations mathématiques développé par Design Science. Par défaut, MS Word en intègre une version allégée qui est l'*Éditeur d'équations*. Il existe un kit de développement (SDK) MathType permettant d'utiliser les fonctionnalités de MathType *via* des macros VBA. En l'occurrence, MathType permet de convertir en MathML 1.0 des équations écrites avec MS Word ou MathType.

5.6 UPCAST

UpCast est un outil développé par Inifinity-Loop. Il nous a permis de convertir une thèse au format RTF en un document XML selon une DTD propre à UpCast[4]. Notons que UpCast permet d'extraire la mise en forme d'un document RTF et de générer une feuille de style CSS. Il est libre de droits pour un usage personnel. À d'autres fins, une licence d'utilisation est nécessaire et celle-ci est peu coûteuse.

5.7 JAVA(*) ET BORLAND JBUILDER PERSONAL(*)

Java est un langage orienté objet qui est développé par Sun Microsystems. L'exécution d'un programme Java est indépendante de la plate-forme d'exécution. Il a été utilisé afin d'assembler tous les composants de la chaîne de traitements, et d'offrir à l'éditeur/archiviste une interface graphique permettant l'automatisation et le paramétrage de la chaîne de traitements. Nous avons utilisé JBuilder Personal de la société Borland pour la programmation en Java.

[4] Nous appellerons cette DTD *XML-UpCast*.

5.8 CHAÎNE (DE TRAITEMENTS) DE NORMAN WALSH(*)

Norman WALSH de Sun Microsystems fait partie du comité DocBook d'OASIS. Il a créé l'implémentation XML de DocBook et participé à la spécification du langage XSL. En sus, il a développé ce que nous appellerons la *chaîne (de traitements) de Norman WALSH*. Elle comprend un ensemble de feuilles de style XSL paramétrables (environ 650) qui permettent en particulier de convertir un document DocBook en HTML ou en XSL-FO. Enfin, notons que ladite chaîne n'est pas un outil à proprement parler, mais un ensemble de feuilles de styles comme nous l'avons mentionné, et qui est libre de droits.

5.9 GEMINI SOLO

Gemini Solo est un logiciel développé par Inceni Technology. Nous avons utilisé cet outil pour convertir un document PDF en un document HTML/CSS car il préserve la mise en forme. Notons que son utilisation est provisoire et qu'il existe quelques outils libres de droits qui sont disponibles mais que nous n'avons pas pu encore tester.

5.10 NOTES

Nous avons considéré le traitement des formules mathématiques avec MathType. Toutefois, les thèses comprennent également des formules chimiques qu'il est nécessaire de prendre en considération. Le format de représentation préconisé semble être cML, mais nous n'avons pas encore effectué une étude des outils permettant leur traitement (i.e. conversion en cML).

6. CHOIX DE LA DTD

Il existe trois DTD adaptées aux besoins des thèses : ISO 12083, TEI-Lite, et DocBook. Nous avons décidé d'avorter l'étude de la DTD ISO 12083 de part les critiques faites à son égard, mais aussi car sa maintenance n'est guère plus régulière voire abandonnée. De plus, les éléments qu'elle définit sont trop peu nombreux pour représenter une thèse correctement.

Le détail de la comparaison est donné à l'Appendice A.

6.1 COMPARAISON ENTRE LA TEI-LITE ET DOCBOOK

6.1.1 TEI-Lite

L'objectif principal de la TEI est de s'assurer que tous textes créés peuvent être utilisés pour divers types d'applications comme le traitement du langage nature, l'extraction d'informations, l'édition électronique, l'hypertexte, le lexicographie, etc. Comme susmentionné, la TEI émerge d'un besoin de normalisation, et ce dans le balisage des textes en littérature et en sciences humaines, ceci indépendamment de la langue utilisée. De fait, elle est plus adaptée aux sciences sociales.

La TEI utilise une DTD modulaire, modularité au niveau des éléments de la DTD, en fonction des différents domaines d'applications qu'elle vise. Par ailleurs, elle comprend un ensemble d'éléments de base commun à toutes les applications.

Les caractéristiques principales de la TEI-Lite sont :

- elle comprend un ensemble réduit, mais essentiel d'éléments dérivés de la TEI DTD ;
- elle permet de traiter un nombre relativement important de textes de types différents ;
- elle est utilisable avec une grande variété d'outils SGML, certains étant gratuits ;
- elle possède une architecture modulaire et extensible ;
- elle est quantitativement bien documentée, des réserves étant émises le côté qualitatif ;
- elle dispose d'une liste de diffusion.

Notons que certains reproches peuvent être faits à la TEI, en ce sens que certaines balises ne signifient pas en fait ce que leurs libellés prétendent, et le module de la TEI afférant aux humanités est quelque peu lacunaire dans l'encodage des ontologies et des bases de données lexicales, etc.

La TEI DTD comprend de nombreux outils permettant l'édition et la présentation d'un document. Notamment, il existe un module pour Emacs permettant de saisir son document directement sous cet éditeur de façon aisée. D'autres outils comme XSL Stylesheets for TEI XML et TEITools permettent de transformer un document TEI dans des formats comme PDF, HTML, RTF, etc. Il existe également un outil qui s'appelle tei2latex permettant de transformer un document écrit en LaTex vers TEI. Enfin pour palier la complexité de la TEI, il est possible de ne considérer qu'un sous-ensemble pertinent de balises de cette TEI grâce à l'outil Pizza Chef.

6.1.2 DocBook

DocBook est un ensemble de balises permettant de décrire des documents structurés tels que des livres et des articles. DocBook était à l'origine plus particulièrement adapté aux documents scientifiques, notamment les documents au sujet de l'informatique, mais il s'adapte à tous types de documents. Il est aussi utilisé pour l'élaboration de transparents et de sites Web. C'est un système libre de droits et de nombreux développements, en particulier dans le domaine public, ont été conduits pour tirer partie des possibilités de DocBook. Il existe une implémentation SGML et XML de DocBook.

DocBook comprend de nombreux outils. Un module créé par Norman Walsh permet d'éditer un document sous Emacs de façon aisée. Il existe également un éditeur pour DocBook qui s'appelle ThotBook. Mais les travaux les plus importants correspondent aux feuilles de style de Norman Walsh qui permettent la transformation dans de nombreux formats comme HTML, HTML Help, Java Help, manpages, PDF, Postscript, RTF, TeX, LaTeX, XHTML, XSL-FO et des modules permettent de prendre en compte le SVG et MathML 1.0.

6.2 CHOIX DE LA DTD

À partir de la comparaison que nous avons faite entre DocBook et TEI, notre choix s'est porté sur DocBook, ceci pour les raisons suivantes :

- La DTD DocBook semble plus adaptée à l'écriture de documents techniques et elle s'est montrée très adaptée à nos besoins. De plus, elle est très adaptée pour les thèses en informatique / génie logiciel ;

- La complexité de la DTD DocBook est moindre à appréhender de part le nombre d'éléments qu'elle présente, mais aussi car la TEI fait un usage intempestif des attributs et car les noms des éléments ne sont pas toujours clairs ;

- Une version spécifique de DocBook intègre le MathML ;

- Les outils disponibles sur le marché sont de bonne facture et les développements actuels sont très actifs. De plus, la plupart des outils disponibles sont développés par Norman Walsh de Sun Microsystems qui est le président du comité DocBook et qui a participé à la spécification XSL. Notons qu'une grande communauté participe également à l'évolution des outils au travers de Norman Walsh ;

- La documentation est très claire, ce qui ne nous a pas semblé être le cas pour la TEI. La TEI étant tellement complexe qu'il est difficile de trouver une information comparé à DocBook. De plus un livre très bien structuré *DocBook : The Definitive Guide* écrit par Norman Walsh, l'un des mainteneurs de DocBook et de CALS ;

- Il existe deux forums officiels sur DocBook où pléthore d'informations techniques sont recensées, ce qui est d'une grande aide lors de problèmes techniques. Une mailing-list est destinée à la DTD DocBook et l'autre aux outils supportant DocBook, dont la chaîne de Norman Walsh ;

- Un dernier point qui nous a plus est que DocBook est également adapté pour la conception de transparents ce qui peut s'avérer utile pour les projets à venir de Doc'INSA.

7. CHOIX DES OUTILS DE LA CHAINE DE TRAITEMENTS

7.1 CONVERTISSEUR RTF-XML

La conversion d'un document MS Word en un document au format DocBook/MathML se fait au moyen d'un convertisseur. Notons qu'un document MS Word peut être enregistré au format RTF, ce dernier étant un format structuré et lisible. Cette conversion préliminaire MS Word en RTF ne dénature la thèse.

7.1.1 Approches de conversion

La conversion d'un document RTF en un document XML peut se faire selon deux approches.

7.1.1.a) PREMIÈRE APPROCHE

La première consiste à convertir un document RTF directement en DocBook. Des outils adaptés doivent être utilisés. Il en existe peu et nous en décrivons brièvement deux d'entre eux :

7.1.1.a - i) LOGICTRAN

Cet outil convertit un fichier RTF 1.0 en un document XML DocBook, (X)HTML ou selon une DTD personnalisée. Assez restrictif dans ses options de configuration, il a l'inconvénient de supporter une ancienne version de RTF (à savoir la version 1.0) alors que la dernière version à ce jour est la 1.6 qui est en conformité avec MS Word 2000. De plus, la dernière mise à jour date de 1999, ce qui n'est pas satisfaisant.

7.1.1.a - ii) MAJIX

Cet outil convertit un document RTF en un document XML, la DTD par défaut étant sDocBook (Simplified DocBook). Nous l'avons trouvé peu convivial et pas très intuitif. De plus la conversion ne nous a pas paru satisfaisante.

7.1.1.b) SECONDE APPROCHE

La seconde approche consiste à convertir un document RTF en un document XML temporaire, la structure de ce dernier document suivant celle du document RTF. Le document temporaire doit ensuite être transformé en DocBook au moyen d'une feuille de style XSLT utilisée par un processeur XSLT. Nous avons retenu cette approche et nous décrivons ci-dessous les outils disponibles.

7.1.1.b - i) UPCAST

Développé par Infinity-Loop, cet outil supporte RTF 1.6, la dernière en date. Il permet d'opérer la dichotomie entre le contenu et la mise en forme, cette dernière pouvant être préservée dans un fichier séparé au format CSS. De plus, il utilise le format de représentation CALS pour les tableaux, lequel format est également utilisé par DocBook. Les images sont extraites et peuvent être enregistrées dans différents formats. Un inconvénient est qu'il ne prend pas en compte les dessins MS Word. Il est libre de droits dans un cadre personnel, mais il n'est pas très coûteux si le choix de la licence est préférable. Sa documentation est bonne et la fréquence de ses mises à jour est régulière.

7.1.1.b - ii) MS WORD 2000

MS Word permet d'enregistrer un document Word au format XHTML. Un gros problème est que le code généré est complexe et que les informations de mise en forme sont intégrées au contenu.

7.1.1.b - iii) DOCUMENT CONVERSION MANAGER

Cet outil permet de convertir un document MS Word 97/2000 en XML, et inversement. Son coût de 25 000 € permet de justifier un manque d'engouement de notre part pour le choix de cet outil.

7.1.1.b - iv) EXPORTXML

Plug-in s'intégrant à MS Word, il ne bénéficie pas d'une totale autonomie dans le processus de conversion, une interaction humaine étant nécessaire. De plus, il n'est pas possible de séparer les informations de mise en forme du contenu.

7.1.1.b - v) OPENOFFICE

OpenOffice est une suite bureautique libre de droits. Le format d'enregistrement des documents écrits avec le traitement de texte qu'il intègre est conforme au format XML. La DTD utilisée est propre à OpenOffice. Les documents écrits au format MS Word peuvent être importés sous OpenOffice facilement et de fait être enregistrés au format XML ; il en est de même pour les documents RTF. Toutefois, nous avons relevé des différences notables entre le document source MS Word/RTF et le document généré sous OpenOffice à la suite de l'importation, le document source pouvant être dénaturé de façon non acceptable.

7.1.2 Solution retenue

Nous avons retenu la solution proposée par Infinity-Loop, UpCast. Grâce à cet outil, il nous est possible d'obtenir à partir d'un document RTF un document XML suivant une DTD spécifique à UpCast et que nous nommons DTD XML-UpCast.

Nous l'avons retenu car :

- il est libre de droits pour utilisation personnelle ;
- il supporte RTF 1.6 ;
- il supporte le format CALS de représentation des tableaux, format également utilisé par DocBook ;
- la plupart des éléments RTF sont représentés dans le nouveau fichier XML-UpCast ;
- les styles définis sous MS Word sont conservés en XML-UpCast;
- les informations de mise en forme peuvent être sauvegardées dans un fichier CSS séparé, ce qui permet de restituer un document de thèse dans un format personnalisable par le doctorant ;
- ses évolutions sont régulières ;

- il est bien documenté ;
- il dispose d'une interface graphique et d'une ligne de commande ;
- son utilisation est simple et conviviale.

7.2 PROCESSEUR XSLT

Concernant le choix du processeur XSLT permettant de convertir un document XML-UpCast en un document DocBook/MathML, nous n'avons pas ressenti l'utilité d'effectuer une comparaison poussée. En effet, le choix du processeur XSLT est moins important que le choix du processeur XSL-FO, étant donné que son rôle est simplement de transformer une structure source en une autre structure. Aucun risque de dénaturation d'une thèse n'est encouru avec les processeurs XSLT actuels. Toutefois, une comparaison succincte nous a permis de tester la rapidité des différents processeurs XSLT ainsi que leur conformité à la recommandation XPath. Les trois processeurs testés étaient : MS XML 3.0 et 4.0, Saxon 6.2 et 7.0, XalanJ 2.0 de Apache.

Les tests effectués ont montré que MS XML 3.0 et 4.0 étaient les plus rapides. Toutefois, seul Saxon 7.0 se conforme quasi totalement à la recommandation XPath 1.0 en couvrant une partie de la version 2.0. De plus, Saxon est un processeur très utilisé et ses évolutions sont régulières. Son auteur est bien connu dans le domaine et s'appelle Michael Kay ; il est l'auteur d'un livre de référence sur XSLT. Enfin, Saxon possède une API Java.

Nous avons donc choisi Saxon 7.0, mais nous continuons à utiliser les trois processeurs.

7.3 PROCESSEUR XSL-FO

La conversion d'un document XSL-FO – ce document étant obtenu à partir de la transformation DocBook/MathML en utilisant un processeur XSLT – en un document au format PDF se fait au moyen d'un processeur XSL-FO. Il existe différents processeurs XSL-FO que nous décrivons et comparons ci-dessous. Cette comparaison nous a permis de choisir le processeur le mieux adapté à nos besoins. Elle a été réalisée par la SNCF et nous nous servons ici de leur étude pour effectuer notre choix.

7.3.1 Comparatif des processeurs XSL-FO

La recommandation XSL-FO en est, en octobre 2001, à sa version 1.0, qui est d'ailleurs la seule à avoir été diffusée. Elle se décline en trois versions :

- niveau basique : objets et propriétés pour pagination de base ;
- niveau étendu : la spécification entière hormis le module de sténographie ;
- niveau complet : la spécification dans son intégralité.

Les quatre processeurs XSL-FO étudiés respectent le niveau basique de la spécification, à quelques différences près, et couvrent une partie du niveau étendu.

7.3.1.a) FOP

FOP est un processeur prenant part au le projet *Apache XML* et est libre de droits. Il est maintenu par le consortium Apache. Il est très utilisé, ceci étant dû notamment à la notoriété d'Apache qui a son aile de nombreux projets très populaires et qui a le soutien de grandes enseignes, aux mises à jour quotidiennes et aux mailing-lists qui fournissent pléthore d'informations techniques en cas de problèmes.

Ce processeur s'exécute *via* la ligne de commande et possède une API Java.

7.3.1.b) PASSIVETEX

PassiveTeX est un processeur développé en TeX. Peu documenté, il bénéficie de peu de support et les évolutions sont beaucoup moins fréquentes qu'avec FOP. De fait, la résolution des problèmes est très peu aisée. Toutefois il est supporté par la chaîne de traitements de Norman Walsh et semble être stable. Notons que sa configuration requiert des compétences techniques plus poussées qu'avec FOP.

Ce processeur s'exécute *via* la ligne de commande.

7.3.1.c) XEP

XEP est un processeur développé par RenderX. Complet et bénéficiant d'un très bon support, il est payant et coûte 5000 $.

Ce processeur s'exécute *via* la ligne de commande.

7.3.1.d) XSL PROCESSOR

XSL Processor est un processeur développé par Antenna House. Très complet et bénéficiant d'une interface graphique, il est disponible pour un coût de 3000$. Toutefois, sa configuration reste peu intuitive malgré l'interface graphique. Un pré-requis est qu'il faut disposer de Acrobat Distiller ou Acrobat PDF Writer pour pouvoir effectuer la sortie en PDF, ou bien prendre l'option PDF en sortie (sans PDF Writer). Ce dernier point n'est certes pas un problème car Doc'INSA possède déjà ces outils.

7.3.2 Bilan

7.3.2.a) TABLEAU COMPARATIF

Pour les projets commerciaux, XEP et Antenna marchent très bien, et sont très stables. FOP est un peu plus limité au point de vue fonctionnalités.

Processeur XSL-Fo	Avantages	Inconvénients
FOP (Apache)	- Le plus connu - Libre de droits - Évolutions quotidiennes - Mailing lists très riches - Fait partie du projet Apache - Utilisation simple - Nombreux formats de sortie supportés (PDF, RTF, ps…)	- Les plus mauvais résultats - Nombreux bogues
XEP (RenderX)	- Très bons résultats - Permet de produire des documents complexes (colonnes, tableaux, images, etc.) - Le plus complet, polyvalent, puisque Antenna est Windows Only	- Payant - Très mauvaise interface utilisateur - Installation difficile (pas selon RenderX)
PassiveTeX	- Libre de droits - Bons résultats	- Implémenté en TeX - Peu de support - Évolutions sporadiques
Antenna House XSL Formatter	- Les meilleurs résultats - Installation très simple - Interface graphique conviviale - Utilisation intuitive - Très bonne stabilité	- Payant - Pas de sortie PDF mais un graphique Windows

Figure 6 – Tableau comparatif des processeurs XSL-FO

7.3.2.b) TABLEAU COMPARATIF DÉTAILLÉ

L'Appendice B fournit le détail de l'étude.

7.3.3 Solution retenue

L'étude qui vient d'être faite nous permet de conclure sur le choix de FOP du consortium Apache. Sa gratuité, son support et ses évolutions quotidiennes sont des atouts qui lui sont favorables. Notons toutefois que c'est le processeur le moins fonctionnel de tous, mais bons espoirs sont gardés quant à ses évolutions. De plus, sa conformité actuelle suffit plus ou moins pour les besoins de représentations des thèses de l'INSA, les tests actuels l'ayant prouvé. Des tests plus poussés avec des documents plus complexes seront réalisés prochaînement.

8. PRESENTATION DE LA CHAINE DE TRAITEMENTS

La chaîne de traitements se décompose en deux sous-chaînes que nous présentons ci-après.

8.1 SOUS-CHAÎNE « TRAITEMENT DU CONTENU »

Cette sous-chaîne (Figure 6) comprend trois étapes :

- *étape 1* : le document de thèse original au format MS Word est traité par un ensemble de macros VBA qui extraient les équations mathématiques écrites avec MS Word ou MathType. Ces équations se présentent – c'est une présentation de haut niveau – au doctorant sous la forme d'images. Les équations extraites (sous forme d'images) sont regroupées dans un fichier, puis transformées en MathML grâce au SDK de MathType qui fournit des modules de conversion. Afin de pouvoir reconstruire le document final, chaque équation MathML est pourvue d'un identifiant. De fait, les équations du document MS Word sont remplacées par ces identifiants. Ce document est ensuite converti au format RTF grâce à une macro VBA. UpCast convertit ce document RTF en un document XML-UpCast. Il extrait et sauvegarde aussi les objets multimédias (i.e. images, graphiques, etc.) dans des fichiers distincts. Notons qu'UpCast convertit les tableaux RTF en tableaux CALS, format de tableaux utilisé par DocBook. Un fichier CSS contenant la mise en forme du document MS Word peut également être produit, mais nous ne nous sommes pas encore intéressé à ce point.

- *étape 2* : le document XML-UpCast est converti en DocBook grâce à une feuille de style XSLT que nous avons conçue et qui crée une correspondance – parfois difficile – entre les éléments du fichier XML-UpCast et ceux relatifs à DocBook. Les références vers les objets multimédias sont préservées dans le document DocBook et les équations MathML y sont intégrées ;

- *étape 3* : la chaîne de Norman WALSH associée à un document DocBook permet au processeur XSLT, Saxon, de générer un document XSL-FO. Ce dernier est transmis au processeur XSL-FO, FOP, qui génère un document PDF. La chaîne de Norman WALSH a été modifiée afin de prendre en compte les spécificités de présentation des thèses de l'INSA. Le document PDF intègre les objets multimédias extraits par UpCast. Enfin, Gemini Solo nous a permis de transformer le document PDF en un document HTML/CSS.

8.2 SOUS-CHAÎNE « TRAITEMENT DES MÉTADONNÉES »

Dans la sous-chaîne *Traitement du contenu*, précédemment décrite, les documents générés ne comprennent pas de page de titre. Au moment de l'écriture de ce document, c'est une sous-chaîne (Figure 7) dédiée au traitement des métadonnées qui a la charge de générer cette page.

Les métadonnées sont décrites au moyen du vocabulaire Dublin Core et représentées dans le format RDF. Une page de titre écrite en XSL-FO récupère les métadonnées à partir du fichier RDF précité. Le fichier de configuration permet à l'éditeur/archiviste de paramétrer la présentation de certains objets sur cette page. Il se peut que des objets multimédias y figurent, comme le logo de l'école doctorale. FOP produit à partir de tous ces fichiers un document PDF. Son équivalent HTML/CSS est ensuite généré par Gemini Solo.

Par la suite, ces deux sous-chaînes fusionneront ; les métadonnées seront directement incluses au format RDF dans le document DocBook. Cette « ségrégation » entre le contenu et les métadonnées a permis une plus grande facilité et rapidité de développement. Ainsi avons-nous pu obvier aux difficultés de configuration de la chaîne de Norman WALSH, d'autant que les modifications à y apporter sont peu ou prou nombreuses et délicates.

Notons que les fichiers permettant la production de la page de titre ont été pensés avec le souci de la modularité et la facilité de configuration. Toutefois, des efforts restent encore à réaliser afin de séparer autant que possible le code XSL-FO des métadonnées.

8.3 NOTES

Le document DocBook n'inclut pas certaines informations comprises à l'origine dans le document MS Word. Il s'agit notamment de la table des matières, de la liste des figures, de la liste des tableaux, de la liste des équations, de la mention de copyright, des hauts et bas de page, de l'index et de la page de titre. La raison est que ces éléments peuvent être générés automatiquement grâce à la chaîne de traitements de Norman Walsh, hormis la page de titre qui est générée au moyen de la sous-chaîne *Traitement des métadonnées*. Par ailleurs, la numérotation des légendes est gérée automatiquement par la chaîne de Norman Walsh, ce qui nous garantit que les objets seront numérotés correctement (voir toutefois la partie 9, *Problèmes rencontrés*).

8.4 DIAGRAMME DE LA CHAÎNE DE TRAIEMENTS

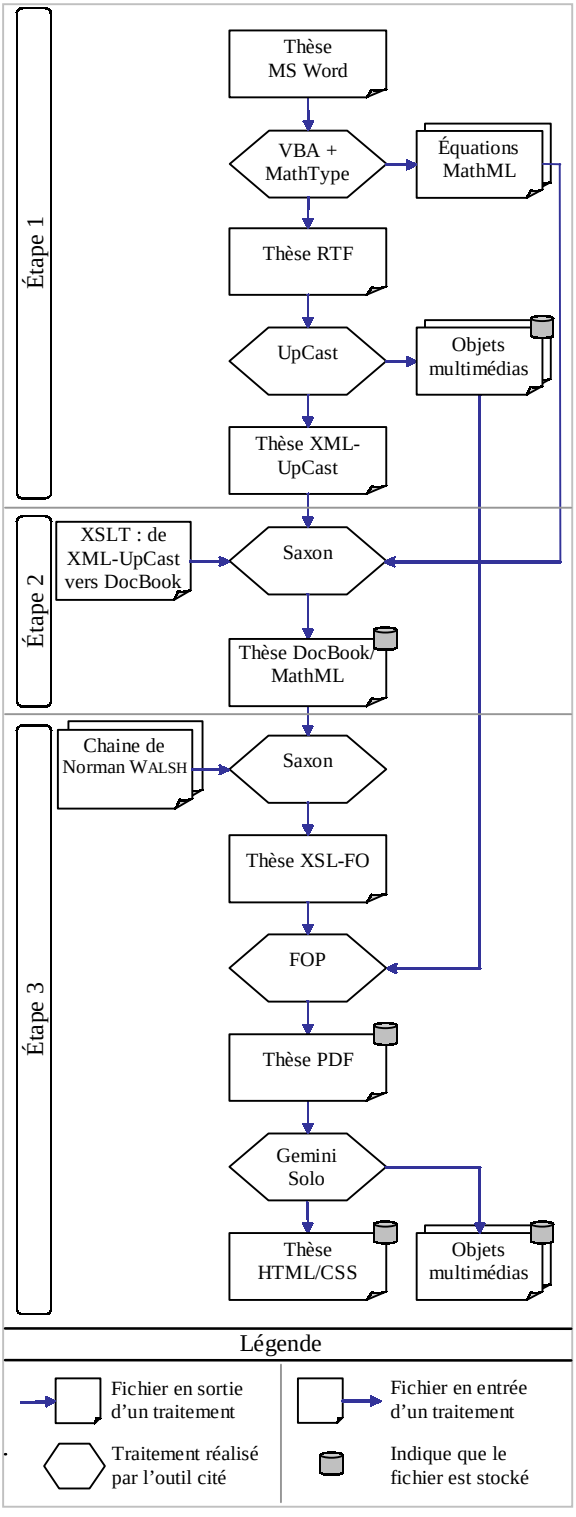

Figure 7 – Sous-chaîne *Traitement du contenu*

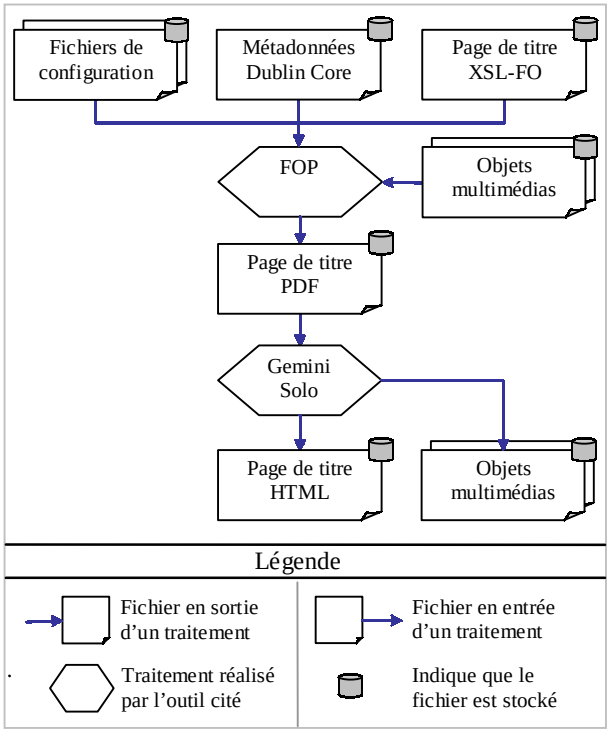

Figure 8 – Sous-chaîne *Traitement des métadonnées*

9. PROBLEMES RENCONTRES

Nous considérons ici trois types de problèmes ; ceux concernant le transfert des thèses, ceux concernant l'apprentissage des outils et les langages, et ceux touchant à la chaîne de traitements. Pour le dernier type, les problèmes ont plus une teneur technique.

9.1 TRANSFERT DES THÈSES

- Lors de l'élaboration du cahier des charges, le problème du moyen ou de l'outil permettant la soumission électronique du travail de thèse vers Doc'INSA nous a amené à conduire une étude des différentes solutions possibles. L'applet Java semble être la solution, bien que certaines réserves soient encore émises à ce sujet. Le problème de l'applet est qu'elle nécessite une machine virtuelle que le doctorant doit installer sur sa station si ce n'est pas déjà fait. Il se peut que le doctorant se heurte à des problèmes d'installation de la machine virtuelle dus à des restrictions de sécurité. Une autre solution viable serait d'utiliser Flash de Macromedia associé à son langage de script ActionScript. Il faut cette fois-ci télécharger un *plug-in* afin de pouvoir exécuter un programme Flash, ce *plug-in* étant beaucoup moins lourd que la machine virtuelle Java. De plus, les possibilités d'interface graphique sont beaucoup plus nombreuses et aisées.

9.2 APPRENTISSAGE DES LANGAGES

- Ce projet faisant intervenir un grand nombre de langages que nous ne connaissions pas ou peu au début de ce projet, il a parfois été difficile de jongler avec eux. Notamment, nous avons eu des difficultés à assimiler le fonctionnement de XSLT, langage déclaratif basé sur la récursivité. De plus, certaines notions fondamentales étaient exposées de façons différentes et parfois contradictoires selon les ouvrages utilisés. L'apprentissage de tous ces langages, recommandations, standards et outils est une tâche longue et fastidieuse que nous avons souvent sous-estimée au niveau du temps, nous faisant par conséquent prendre du retard dans l'avancement du projet.

9.3 CHAÎNE DE TRAITEMENTS

9.3.1 Problèmes liés à la plate-forme d'exécution

- Il est nécessaire d'exécuter la chaîne de traitements sous un environnement NT (Windows NT/2000/XP) pour une prise en charge complète et aisée de Unicode. En effet, nous avons rencontré des problèmes liés à Windows 98 qui ne supporte pas Unicode directement.

9.3.2 Problèmes liés à MS Word

- Les légendes des objets (tableaux, images, etc.) ne sont pas associées logiquement auxdits objets. Ceci pose problème attendu que DocBook permet d'attacher une légende à un objet. De fait, une légende devra toujours se trouver précisément au même endroit dans le document MS Word pour qu'elle puisse être attachée sans ambiguïté à son objet dans le document DocBook.

- Les légendes doivent obligatoirement figurer dans le corps du texte. Elles ne peuvent en aucun cas figurer dans une zone de texte, une cellule d'un tableau ou tout autre conteneur. Ceci peut avoir un impact sur les traitements effectués par la chaîne. Voir le point qui suit.

- Le doctorant doit être vigilant sur la numérotation des légendes et les renvois manuels vers des légendes numérotées. En effet, sous MS Word, la numérotation (i.e. le séquencement) peut ne pas être logique mais rester viable tandis que la chaîne de traitements de Norman Walsh renumérote toutes les légendes. De fait, les référencements manuels vers des légendes numérotés peuvent ne plus être viables. Une solution serait de faire confiance au doctorant, et une autre serait de garder dans le document DocBook les légendes telles qu'elles sont numérotées dans le document source, la chaîne de Norman Walsh étant de fait déchargée de cette tâche de renumérotation.

- Les objets flottants, c'est-à-dire ceux n'ayant pas de cadre (par exemple, les zones de texte), posent également le problème des légendes. En effet, une légende peut apparaître au-dessous d'un tel objet, mais dans la structure du document, il peut en être autrement, l'objet pouvant se trouver une demi page au-dessus de sa légende. La solution est de créer un objet Microsoft Word. Voir ci-dessous.

- Il est important de structurer sa thèse correctement, en ce sens qu'un titre de niveau 2 doit être compris dans un titre de niveau 1 uniquement, un titre de niveau 3 dans un titre de niveau 2 uniquement, etc. Ceci est nécessaire car la DTD DocBook est stricte sur l'imbrication des titres.

- Une thèse peut comprendre des renvois à des numéros de page. Ici, le problème est complexe car il faut que le document de restitution soit à l'identique du document source pour que le référencement soit toujours viable et fiable. Une solution serait de référencer un paragraphe, un titre, ou plus généralement une information de contenu

- Au départ, il avait été convenu d'extraire les métadonnées (par exemple, titre de la thèse, nom du doctorant, membres du jury, école doctorale…) à partir du document MS Word, ces métadonnées apparaissant sur la page de titre. Cette solution a été prototypée, et nous en sommes venus à la conclusion qu'elle était risquée. En effet, pour que les métadonnées puissent être reconnues en tant que telles, il faut que le doctorant choisisse les bons styles, le modèle de document l'assistant toutefois dans cette tâche. Un problème spécifique est celui des mots-clefs. Dans les thèses traditionnelles, les mots-clefs sont séparés par une espace, suivie d'un tiret, suivie d'une seconde espace. Il suffit que ceci ne soit pas respecté pour qu'il y ait une ambiguïté d'interprétation (mot composé ou mot-clef ?). En outre, il se peut que le doctorant veuille introduire un sous-titre ou une autre métadonnée sans lui attacher de style (car il n'y a pas pensé ou aucun style n'est approprié). Aussi cette métadonnée ne sera pas extraite, et de l'information sera perdue. Finalement, la solution est de rassembler toutes les métadonnées à renseigner dans un formulaire que le doctorant devra compléter, cette procédure introduisant une redondance des informations tapées.

- MS Word ne comprend pas de fonctions permettant de structurer un glossaire ou une bibliographie. De fait, ces éléments seront considérés comme du texte brut sans sémantisme particulier associé.

9.3.3 Problèmes liés à UpCast

- UpCast ne prend pas en compte les dessins MS Word. Une solution est de créer un objet (plus précisément, un objet conteneur) *Image Microsoft Word* (Menu Insertion/Objet…/Image Microsoft Word) et de dessiner dans le conteneur ainsi créé.

- Des premiers tests avaient montré que UpCast supportait Unicode. Des tests récents ont montré que certains caractères particuliers n'étaient pas considérés. UpCast est ici mis en cause. Une étude plus poussée permettra de déceler l'origine réelle du problème et de trouver une solution adaptée.

9.3.4 Problèmes liés à la chaîne de traitements de Norman Walsh

- La chaîne de traitements de Norman Walsh permet de convertir un document DocBook en PDF et en HTML notamment, ceci selon une présentation déjà prédéfinie. Pour une présentation personnalisée quelque peu éloignée de celle par défaut, il est nécessaire de se plonger dans le code de la chaîne de traitements qui comprend en tout environ 650 fichiers. De plus, le code n'est pas commenté, mais il est bien structuré ! Cette entreprise est difficile et très longue. Il faut compter en moyenne deux heures pour configurer un élément de présentation complexe.

- Les tableaux CALS sont utilisés par UpCast et la chaîne de Norman Walsh. Toutefois, nous ne sommes pas arrivé à restituer la présentation initiale d'un tableau, un tableau CALS contenant des informations de mise en page (i.e. taille des celles, orientation du texte, etc.) mais qui n'apparaissent pas lors de la restitution. La chaîne de traitements de Norman Walsh est mise en cause, mais il serait plus légitime d'incriminer FOP. Des tests plus poussés nous conduirons à déterminer la cause.

9.3.5 Problèmes liés à DocBook

- Bien que n'ayons pas « investiguer » la chose, il semblerait que DocBook ne prenne pas en compte les pages en paysage et les sections à colonnes, le premier point étant plus problématique.

CONCLUSION

Résultats obtenus

La chaîne de traitements produit à l'heure actuelle des documents DocBook/MathML à partir d'un document MS Word, ceci en conformité avec le modèle de document élaboré. Elle produit aussi à partir du document DocBook généré des documents PDF et HTML. Restent toutefois certains problèmes liés aux limitations d'UpCast (ex. : gestion des objets sans cadre, dits *flottants*). D'autres éléments n'ont pas été pris en compte, comme les pages à colonnes et les pages en paysage (ou *à l'italienne*). Enfin, d'autres problèmes sont liés à MS Word, comme le fait qu'un tableau ou un objet ne soient pas associés par un lien logique avec leur légende. Ceci peut toutefois être résolu avec un peu de rigueur de la part des doctorant.

De plus, les modules de la chaîne de traitements sont en train d'être assemblés afin que les traitements puissent être contrôlés *via* une IHM.

D'autres développements restent encore à réaliser, notamment pour que le doctorant puisse enregistrer électroniquement sa thèse auprès de Doc'INSA, et de fait renseigner les métadonnées utiles pour la création de la page de titre.

Perspectives d'évolution de la chaîne

Au moment de la rédaction de ce rapport la phase de prototypage a atteint les 70% des développements nécessaires, la sous-phase *Traitement et archivage* étant pratiquement aboutie.

À court terme, l'on peut s'attendre à ce que les thèses commencent à être transformées en DocBook/MathML, mais ceci ne pourra se faire sans la participation des doctorants qui sont le premier maillon de la chaîne. Aussi une formation *ad hoc* devra-t-elle leur être dispensée.

Par la suite, il sera possible d'envisager que les doctorants rédigent leur thèse directement en DocBook/MathML avec un éditeur adapté comme Emacs associé au module DocBook écrit par Norman WALSH.

Par ailleurs, la chaîne de traitements devra prendre en compte les thèses au format L^AT_EX. Afin de réduire les développements, il serait envisageable de convertir un document L^AT_EX en RTF.

Dans quelques années, il est quasi certain que les navigateurs actuels pourront afficher correctement un document XML associé à une feuille de style XSL. De fait, il sera possible de visualiser sans conversion préliminaire un document DocBook directement dans son navigateur.

Notons que la chaîne de traitement développée ne prend pas en compte les mises en forme personnalisées des doctorants, lesquelles peuvent être récupérées par UpCast dans un fichier CSS. Il serait intéressant de voir le résultat obtenu.

D'autres choses n'ont pas été considérées comme la structuration d'un document de thèse en plusieurs fichiers, les références à des numéros de page (chose complexe), etc. et qu'il va falloir prendre en compte à l'avenir.

Enfin, l'intégration d'une base de données, telle qu'Oracle 9i qui gère le XML, serait souhaitable - solution que nous avons étudiée lors de ce PFE.

Bilan personnel

Ce projet nous a beaucoup apporté car il nous a permis de mettre en pratique des méthodes ainsi que d'autres acquis souvent restés au stade de la théorie. En outre, il nous a permis de réaliser l'importance des premières phases d'un projet qui établissent les fondations pour son bon déroulement.

L'intérêt de ce projet a été de travailler avec les technologies de la nébuleuse XML et d'en apprécier leur puissance. Toutefois, cet intérêt n'a pas été sans susciter de nombreuses difficultés notamment dues au grand nombre de technologies et outils utilisés.

Le projet étant terminé, nos perspectives sont maintenant d'aboutir à un prototype opérationnel couvrant principalement les étapes de composition, et de traitement et d'archivage, même s'il n'est pas entièrement fonctionnel. Enfin, il convient également de terminer la documentation pour faciliter la maintenance et l'évolution du système.

APPENDICE A – COMPARAISON ENTRE TEI-LITE ET DOCBOOK

Ce tableau compare la DTD TEI Lite et la DTD DocBook en fonction de critères. Un critère est vérifié s'il est marqué d'un *tick* (✓). Dans le cas où la réponse n'a pu être trouvée, nous avons laissé la case vide.

Éléments constitutifs d'une thèse

Critères	TEI-Lite	DocBook
Liminaires		
• Copyright	✓	✓
• Dédicace	✓	✓
• Épigraphe	✓	✓
• Errata		✓
• Mots-clefs français	✓	✓
• Mots-clefs anglais	✓	✓
• Préface	✓	✓
• Remerciements	✓	✓
• Résumé anglais	✓	✓
• Résumé français	✓	✓
• Table		
- Des matières	✓	✓
- Des équations	✓ grâce aux IDs	✓ grâce aux IDs
- Des figures	✓ grâce aux IDs	✓ grâce aux IDs
- Des tableaux	✓ grâce aux IDs	✓ grâce aux IDs
Corps de texte		
• Citation		
- Simple	✓	✓
- Division		✓
• Commentaires	✓	✓
• Divison	✓	✓
• Lien hypertexte		
- Intradocument (renvois)	✓	✓
- Inter-document	✓	
- URL		✓
• Listes		
- À puces	✓	✓
- Ordonnées	✓	✓
- Simples	✓	✓
- Segmentées		✓
- Indentées	✓	✓
- Sous-liste	✓	✓
• Notes		
- De bas de page	✓	✓
- De fin de document	✓	✓
- De fin de division	✓	

Critères	TEI-Lite	DocBook
• Objet multimédia		
- Image	✓	✓
- Légende d'image	✓	✓
- Texte alternatif d'image	✓	✓
- Graphique	✓	✓
- Légende de graphique	✓	✓
- Support SVG	✓ grâce aux espaces de noms	✓
- Vidéo		✓
- Légende de vidéo		✓
- Texte alternatif de vidéo		✓
- Audio		✓
- Légende d'audio		✓
- Texte alternatif d'audio		✓ grâce aux espaces de noms
- Équation		✓ DocBook MathML
- Support MathML 2.0	✓ grâce aux espaces de noms	✓
- Légende d'équation	~ on peut croire que oui	✓
- Texte alternatif d'équation		✓
- Autres objets binaires	✓	✓
• Paragraphe	✓	✓
• Signet		
- Générique	✓	✓
- Bibliographique	✓	✓
- Glossaire	✓	✓
• Tableau	✓	✓
- Tableaux CALS	✓	✓ CALS
- Légende de tableau	✓	✓ CALS
- Tableaux imbriqués	✓ CALS	✓ CALS
- Fusion de cellules	✓ CALS	✓ CALS
- Entêtes de tableaux	✓ CALS	✓ CALS
- Cellules conteneurs	✓ CALS	✓ CALS
Annexes		
• Addenda / postface	✓	✓
• Appendice / annexe	✓	✓
• Bibliographie	✓	✓
• Glossaire	✓	✓
• Index	✓	✓

Tableau 6 – Comparaison détaillée entre la TEI Lite et DocBook (éléments constitutifs d'une thèse)

Autres critères

Critères	TEI Lite	DocBook
Outils		
Validateur de DTD	✓	✓
Transformation vers HTML	✓	✓ FOP + Norman Walsh
Transformation vers PDF	✓	✓ FOP + Norman Walsh
Transformation de RTF	✓	✓ partiel : FOP + N. Walsh
Éditeur DocBook	✓ Emacs	✓ Emacs
Caractères / texte		
Texte brut CDATA	✓	✓
Mise en évidence (italique, …)	✓	✓
Code de langage	?	✓
Saut de page	✓	✓
Exposant		✓
Indice		✓
Support UNICODE	✓	✓
Langue pour le document	✓	✓
Langue pour un élément	✓	✓
Texte brut	✓	✓

Tableau 7 – Comparaison détaillée entre la TEI Lite et DocBook (autres critères)

APPENDICE B – COMPARAISON DES PROCESSEURS XSL-FO

Ce tableau est tiré d'une étude réalisée par la SNCF. Il permet d'évaluer les principaux processeurs XSL-FO

		Général			Performances techniques									Côté utilisateur							Performances			Conformité	
1=mauvais 5=excellent	Critères	Environnement	Libre de droits / Payant	Fréquence des MAJ (1-5)	Rapport des erreurs (1-5)	Gestion des objets complexes (1-5)	Gestion des jeux de caractères (1-5)	Référencement aux DTD (1-5)	Gestion des bordures de tableau (1-5)	Génération TDM (1-5)	Génération Index (1-5)	Gestion des ruptures de page (1-5)	Respect des marges (1-5)	Simplicité d'utilisation (1-5)	Aide en ligne (1-5)	Manuel utilisateur, didacticiel (o/n)	Interface graphique (o/n)	Réponses aux problèmes rencontrés (1-5)	Facilité de configuration (1-5)	Personnalisation (1-5)	Traitement des fichiers de >50 pages (1-5)	Efficacité, résultats obtenus (1-5)	Rapidité (1-5)	Liste des objets et propriétés de la spécification supportées (o/n)	Correspondance à la norme
Processeurs sélectionnés																									
FOP, Apache		Java 2	Libre de droits	4	3	4	4	3	2	4	4	1	2	3	2	o	n	3	1	3	2	2	2	o	2
XSL Processor, Antenna House		Windows	Payant	2	4	4	4	4	3	4	4	2	3	4	1	o	o	4	3	4	3	4	5	o	2
XEP, RenderX		Java 2	Payant	2	3	5	4	4	4	5	4	3	3	3	1	o	n	4	1	3	4	4		o	3
PassiveTeX		TeX	Libre de droits	1	*non testé*									*non testé*							*non testé*			*non testé*	
Autres processeurs																									
UFO Formatter, Unicorn		Windows	Libre de droits														n								
FOA		Java 2	Libre de droits													o	o							n	
REXP		Java 2	Libre de droits														n					4			

Figure 9 – Tableau comparatif des principaux processeurs XSL-FO

Dernière page