

VUE D'ENSEMBLE DE OFFICE OPEN XML



ECMA TC45
TOM NGO (NEXTPAGE), REDACTEUR

1 INTRODUCTION

Le format Office Open XML (OpenXML) est un standard ouvert proposé pour les documents de traitement de texte, les présentations et les feuilles de calcul, qui peut être librement intégré par plusieurs applications sur plusieurs plateformes. Sa publication est saluée par les organisations qui souhaitent mettre en œuvre des applications capables d'utiliser le format, par les entités commerciales et gouvernementales qui proposent de tels logiciels, et par les enseignants ou auteurs qui enseignent ce format. Tous les utilisateurs bénéficient de l'émergence d'un standard XML pour leurs documents, notamment en termes de stabilité, de conservation, d'interopérabilité et d'évolutivité.

Le travail de standardisation OpenXML a été effectué par l'Ecma International via le Technical Committee 45 (Comité Technique 45, TC45), qui comprend des représentants d'Apple, Barclays Capital, BP, la British Library, Essilor, Intel, Microsoft, NextPage, Novell, Statoil, Toshiba et la United-States Library of Congress (1).

Ce livre blanc présente les principaux aspects du standard OpenXML. Il permet de :

- comprendre l'objet d'OpenXML ainsi que la structure de ses spécifications ;
- connaître ses propriétés : son approche en matière de compatibilité en amont, de conservation, d'extension, de schémas personnalisés, de définition de sous-ensembles, de plateformes multiples, d'internationalisation et d'accessibilité ;
- découvrir comment s'articule la structure évoluée d'un fichier OpenXML, et comment naviguer rapidement vers n'importe quelle partie de la spécification pour obtenir des informations détaillées.

2 UTILITÉ DU STANDARD

OpenXML a tout d'abord été conçu pour permettre de représenter fidèlement l'ensemble préexistant des documents de traitement de texte, de présentations et de feuilles de calcul qui sont codés dans les formats binaires mis au point par Microsoft Corporation. Le processus de standardisation consistait à reproduire dans le langage XML les possibilités requises pour représenter les groupes existants, les étendre, fournir une documentation détaillée et assurer l'interopérabilité. Actuellement, plus de 400 millions d'utilisateurs génèrent des documents dans des formats binaires, soit plus de 40 milliards de documents et des milliards de plus créés chaque année.

Les formats binaires initiaux de ces fichiers ont été créés à une époque où l'espace était précieux et où le temps d'analyse affectait sérieusement l'expérience de l'utilisateur. Ils étaient basés sur une sérialisation directe des structures de données en mémoire jusqu'ici utilisées par les applications Microsoft® Office®. Les infrastructures de standards (en particulier XML), les réseaux et le matériel modernes autorisent une nouvelle conception qui favorise l'implémentation par différents fournisseurs sur différentes plateformes et permet une évolution.

Parallèlement à ces avancées technologiques, les marchés se sont diversifiés pour intégrer une nouvelle série d'applications non prévues à l'origine dans le monde simple des programmes d'édition de documents. Ces nouvelles applications incluent les fonctions suivantes :

- génération automatique de documents à partir de données professionnelles ;
- extraction de données professionnelles des documents et transfert de ces données dans des applications d'entreprise ;
- réalisation de tâches restreintes qui opèrent sur un petit sous-ensemble d'un document tout en permettant les modifications ;
- accès par des utilisateurs ayant des besoins spécifiques, tels que les non-voyants ;
- exécution sur une variété de matériels, y compris les appareils mobiles.

Le problème fondamental est sans doute celui de la conservation à long terme. Nous avons appris à créer des volumes d'informations en augmentation exponentielle. Nous avons jusqu'ici codé ces informations à l'aide de représentations numériques si étroitement associée aux programmes qui les a créées qu'après une ou deux décennies, elles sont devenues à terme extrêmement difficiles à lire sans perte significative. La conservation des investissements intellectuels et financiers consentis dans ces documents (existants et nouveaux) est devenue une véritable priorité.

L'émergence de ces quatre tendances (l'adoption extrêmement large des formats binaires, les avancées technologiques, un marché demandant avec force une diversification des applications et les difficultés croissantes de la conservation à long terme) a rendu obligatoire la définition d'un format XML ouvert et la migration de milliards de documents vers ce format avec le moins de perte possible. De plus, standardiser ce format XML ouvert et le gérer dans le temps crée un environnement dans lequel toute organisation peut compter de façon fiable sur la stabilité durable de la spécification, convaincue que l'évolution future profitera des gains et équilibres permis par une procédure de standardisation.

Il existe divers standards et spécifications de documents, notamment HTML, XHTML, PDF et ses sous-ensembles, ODF, DocBook, DITA et RTF. À l'instar des nombreux standards représentant les images bitmap, notamment TIFF/IT, TIFF/EP, JPEG 2000 et PNG, chacun a été créé à des fins spécifiques. OpenXML répond aux besoins d'un standard couvrant les fonctions représentées dans le corpus de documents existant. À notre connaissance, il est le seul format de document prenant en charge toutes les caractéristiques des formats binaires.

OpenXML définit des formats pour des documents de traitement de texte, des présentations et des feuilles de calcul. Chaque type de document est spécifié par le biais d'un langage de balisage principal : WordprocessingML, PresentationML ou SpreadsheetML. L'incorporation de mécanismes permet à un document de ces trois types de contenir des informations dans les autres principaux langages de balisage et dans un certain nombre de langages de balisage secondaires.

La spécification présente à la fois un caractère normatif (définissant OpenXML) et informatif (favorisant une compréhension intuitive par le lecteur). Elle est structurée en diverses parties pour répondre aux besoins de diverses audiences.

Chapitre 1 — Notions fondamentales 165 pages	<ul style="list-style-type: none"> – Définit le vocabulaire, les conventions de notation et les abréviations. – Résume les trois principaux langages de balisage et les langages de balisage secondaires. – Établit les conditions garantissant la conformité et fournit des directives d'interopérabilité. – Décrit les contraintes des conventions OPC (Open Packaging Conventions) applicables à chaque type de document.
Chapitre 2 — Conventions OPC (Open Packaging Conventions) 125 pages	<ul style="list-style-type: none"> – Définit les conventions OPC (Open Packaging Conventions). Chaque fichier OpenXML comprend une série de flux d'octets appelés « parties », combinés dans un conteneur qualifié de « package ». Le format de conditionnement est défini par les conventions OPC. – Décrit une implémentation physique recommandée des conventions OPC qui utilise le format de fichier Zip. – Déclare les schémas XML des OPC comme définitions de schémas XML (XML Schema Definitions, XSD) (2), dans une annexe diffusée uniquement au format électronique. L'annexe inclut également des représentations non normatives des schémas utilisant RELAX NG (ISO/IEC 19757-2) (3).
Chapitre 3 — Document élémentaire 466 pages	<ul style="list-style-type: none"> – Introduit les fonctions de chaque langage de balisage, fournissant un contexte et des éléments d'illustration par le biais d'exemples et de diagrammes. Cette partie est informative (non normative). – Décrit la fonction permettant de stocker des données XML personnalisées dans un package pour permettre l'intégration avec des données d'entreprise.
Chapitre 4 — Référence du langage de balisage 5 756 pages	<ul style="list-style-type: none"> – Définit chaque élément et attribut, la hiérarchie des relations parent/enfant, et des sémantiques supplémentaires le cas échéant. Cette partie est destinée à servir de référence lorsque des informations détaillées sur un élément ou un attribut sont requises. – Définit la fonction de stockage de données XML personnalisées.

- Déclare les schémas XML des langages de balisage comme XSD (2), dans une annexe diffusée uniquement au format électronique. L'annexe les exprime également de façon non normative avec RELAX NG (ISO/IEC 19757-2) (3).

Chapitre 5 — Compatibilité
et extension du balisage
34 pages

- Décrit les fonctions d'extension de documents OpenXML.
- Spécifie des éléments et des attributs assurant l'interopérabilité d'applications avec différentes extensions.
- Exprime des règles d'extension utilisant NVDL (ISO/IEC 19757-4) (4).

Pour simplifier la navigation dans ces documents et leur lecture, les versions électroniques comportent de nombreux liens internes actifs. En particulier, la partie 4 comporte une multitude de liens vers des éléments parents et enfants.

4 PROPRIÉTÉS DU STANDARD

Cette section vous prépare à étudier OpenXML en décrivant certaines de ses propriétés évoluées. Chaque sous-section décrit l'une de ces propriétés et fait référence à des fonctions spécifiques d'OpenXML.

- Le terme « Interopérabilité » décrit comment OpenXML est indépendant des formats, des fonctions et de l'environnement d'exécution propriétaires, laissant une grande latitude aux développeurs.
- « L'internationalisation » fait référence à quelques méthodes représentatives au moyen desquelles OpenXML prend en charge tous les principaux groupes de langues.
- « Simplicité d'adoption par les développeurs », « Compression » et « Modularité » répertorient les diverses approches grâce auxquelles OpenXML évite ou élimine les obstacles à l'implémentation par les diverses parties : courbe d'apprentissage, jeu de fonctions minimal et performances.
- « Migration haute fidélité » indique comment OpenXML répond à l'objectif primordial de préserver les informations, notamment l'intention première du créateur, dans les documents nouveaux et existants.
- « Facilité d'intégration des données » décrit comment OpenXML incorpore des informations d'entreprise dans des schémas personnalisés pour permettre l'intégration et la réutilisation d'informations entre des applications bureautiques et des systèmes d'information.
- « Possibilités d'innovation » décrit comment OpenXML prépare l'avenir en définissant des mécanismes d'extension et en prévoyant l'interopérabilité entre des applications avec différents jeux de fonctions.

Le reste de ce document, notamment cette section, présente les grandes lignes du standard OpenXML. Les références à la spécification ont toutes la forme suivante : §chapitre:section.sous-section ; par exemple, §1:2.5 fait référence au Chapitre 1, Section 2.5 de la spécification. Les références à d'autres titres dans ce document sont effectuées par noms.

4.1 INTEROPÉRABILITÉ

Les développeurs peuvent écrire des applications qui utilisent et génèrent le format OpenXML sur plusieurs plateformes.

De plus, l'interopérabilité d'OpenXML a été obtenue par des contributions, des modifications et des révisions de la spécification par des membres du comité Ecma TC45 (1) provenant de divers horizons et ayant différents intérêts professionnels. La représentation inclut notamment :

- Des constructeurs ou éditeurs (Apple, Intel, Microsoft, NextPage, Novell et Toshiba) exploitant plusieurs systèmes d'exploitation (Linux, Mac OS et Windows) et visant plusieurs utilisations d'OpenXML
- Des entreprises (BP, Barclays Capital, Essilor, Statoil) ayant consenti de lourds investissements dans du contenu existant, notamment des systèmes transactionnels stratégiques
- La British Library et la United States Library of Congress, ayant toutes deux des intérêts directs dans la problématique de la conservation

Pendant les préparatifs, les membres du comité ont soulevé et résolu des centaines de problèmes touchant la stratégie, la clarté, la sémantique et l'éventuelle dépendance de l'environnement. Problèmes et activités marquantes :

- Fonctions assurant l'indépendance de la plateforme pour des mécanismes qui étaient propriétaires dans les formats binaires d'origine
- Conditions de conformité
- Contenu des schémas
- Autres représentations des schémas et mécanismes d'extension utilisant RELAX NG (ISO/IEC 19757-2) et NVDL (ISO/IEC 19757-4) (4)
- Développement d'outils pour analyser et visualiser automatiquement les schémas
- Internationalisation
- Intégrité, exactitude et clarté des descriptions dans la spécification, souvent à la suite d'une tentative d'implémentation de certaines parties de la spécification

Le reste de cette sous-section souligne des points spécifiques pour lesquels OpenXML s'éloigne des formats binaires d'origine par souci d'interopérabilité.

L'une des exigences primordiales de l'interopérabilité est l'indépendance vis-à-vis de tout type de contenu source spécifique.

- OpenXML n'applique aucune restriction quant aux types d'images, de sons ou de vidéos. Par exemple, les images peuvent être au format GIF, PNG, TIFF, PICT, JPEG ou tout autre type d'images (§1:14.2.12).
- Les contrôles incorporés peuvent être de tout type, par exemple Java ou ActiveX (§1:15.2.8).
- Les spécifications de police WordprocessingML peuvent inclure des métriques de police et des informations PANOSE pour simplifier la détermination d'une police de substitution si la police d'origine n'est pas disponible (§3:2.10.5).

En outre, le format OpenXML permet d'éviter la dépendance à l'environnement dans lequel s'exécute l'application qui crée un document.

- L'exemple classique est un contrôle ou une application externe qui génère une image pour une partie de la surface d'affichage. Pour prévoir le cas où le contrôle ou l'application ne serait pas disponible ou ne pourrait pas s'exécuter dans un environnement d'exécution donné, le fichier du document peut contenir une représentation d'image. Ce mécanisme existe également dans les anciens formats binaires.

- OpenXML introduit un mécanisme plus général qualifié de bloc de contenu de remplacement (Alternate Content Block) (§3:2.18.4), pouvant être utilisé dans diverses situations lorsqu'une application consommatrice n'est pas en mesure d'interpréter le contenu généré par une application productrice. Il est généralement utilisé dans le contexte de l'extension. Ce mécanisme est décrit de façon détaillée dans la sous-section « Possibilités d'innovation ».
- OpenXML évite la dépendance à tout paramètre significatif dans l'environnement du producteur d'un document mais pas nécessairement présent dans l'environnement du consommateur. Par exemple, le paramètre CT_SYSCOLOR est un index dans une table de couleurs dans l'environnement de production. Pour prendre en charge la portabilité vers un environnement de consommation différent, PresentationML permet au producteur de masquer la couleur système qui était utilisée au moment de la création d'un document.

Enfin, et plus fondamental, Office OpenXML est conforme aux standards ouverts W3C tels que le langage XML (5) et les espaces de noms XML (6). Cette spécificité autorise à elle seule un niveau de base d'interopérabilité sur toutes les plateformes et tous les systèmes d'exploitation qui adhèrent à ces standards ouverts.

4.2 INTERNATIONALISATION

OpenXML prend en charge les fonctionnalités d'internationalisation rendues nécessaires par la diversité des langues utilisées, par exemple l'arabe, le chinois (trois variantes), l'hébreu, l'hindi, le japonais, le coréen, le russe ou le turc.

OpenXML, par nature, prend en charge la norme Unicode puisqu'il s'agit d'un format XML. En outre, OpenXML possède un jeu complet de fonctions d'internationalisation qui ont été améliorées au fil des ans. Cette liste est représentative :

Orientation du texte : OpenXML prend en charge les langues de gauche à droite (LTR) et de droite à gauche (RTL). Il prend également en charge les langues bidirectionnelles (« BiDi ») telles que l'arabe, le farsi, l'urdu, l'hébreu et le yiddish, qui vont de droite à gauche, mais peuvent contenir des segments de texte incorporé allant dans le sens inverse. Dans WordprocessingML, la direction du texte peut être contrôlée au niveau du paragraphe (§4:2.3.1.6) et au niveau d'un « run » au sein d'un paragraphe (§4:2.3.2.28). De même, dans le texte DrawingML, elle peut être contrôlée au niveau du corps du texte (§4:5.1.5.1.1), au niveau du paragraphe (§4:5.1.5.2.2) et au sein des listes à puce numérotées (§4.5.1.5.4).

Flux du texte : dans WordprocessingML, la direction du flux du texte peut être contrôlée au niveau d'une section ou d'un tableau (§4:2.3.1.41) ou au niveau d'un paragraphe (§4:2.3.2.28). Aux niveaux de la section et du tableau, le flux du texte peut être contrôlé dans les directions verticale et horizontale. Cela permet à OpenXML de prendre en charge toutes les dispositions de texte potentielles (par exemple les lignes verticales de haut en bas et se succédant de gauche à droite, pour prendre en charge le mongol). Cela affecte la disposition des listes, des tableaux et autres éléments de présentation. DrawingML utilise également les paramètres Kumimoji aux niveaux du paragraphe et du « run » pour dérouler le texte horizontalement et les nombres verticalement (§4:5.1.5.2.3, §4:5.1.5.3.9). Dans WordprocessingML (§4:2.3.1.16) et PresentationML (§4:4.3.1.15), le flux des caractères peut également être spécifié à l'aide de paramètres Kinsoku pour préciser quels caractères peuvent commencer et terminer une ligne de texte.

Représentation des nombres : pour la mise en forme des champs dans WordprocessingML (§4:2.16.4.3), la numérotation des paragraphes/listes dans WordprocessingML (§4:2.9) et la numérotation dans DrawingML (§4:5.1.5.4, §4:5.1.12.61), les quelques dizaines de formats existants peuvent être utilisés, notamment Hiragana, arabe, Abjad, thaïlandais, texte cardinal (par exemple « cent vingt-trois »), chinois, coréen (Chosung ou Ganada), hébreu, hindi, japonais, romain ou vietnamien. Ces fonctions prennent également en charge les valeurs à séparation fractionnaire (par exemple, « 1.00 » et « 1,00 ») et séparateurs de liste arbitraires. L'internationalisation de la mise en forme numérique est particulièrement robuste dans SpreadsheetML, qui prend en charge l'ensemble de ces fonctions dans les formats de cellules (§4:3.8.30) et dans les références à des données externes (§4.3.13.12).

Représentation de la date : dans WordprocessingML (§4:2.18.7) et dans SpreadsheetML (§4:3.18.5), les dates de calendrier peuvent être écrites dans les formats grégoriens (trois variantes), hébreu, Hijri, japonais (ère impériale), coréen (ère Tangun), Saka, taiwanais et thaïlandais.

Formules : la spécification des formules dans SpreadsheetML fournit plusieurs fonctions de conversion associées à l'internationalisation, telles que BAHTTEXT (§4:3.17.7.22), JIS (§4:3.17.7.185) et ASC (§4:3.17.7.11).

Identificateurs de langues : dans WordprocessingML (§4:2.3.2.18) et dans DrawingML (§4:5.1.5.3), chaque paragraphe et chaque « run » peut être balisé avec un identificateur de langue, ce qui permet à une application de sélectionner les outils de contrôle appropriés ainsi que d'autres fonctionnalités spécifiques de la langue. Outre un identificateur pour chaque langue, OpenXML prend en charge l'attribution d'un nom à un jeu de caractères, une famille de police et une valeur PANOSE pour aider l'application à choisir un jeu de caractères de substitution approprié lorsque la prise en charge des paramètres régionaux est absente.

4.3 SIMPLICITÉ D'ADOPTION PAR LES DÉVELOPPEURS

Un développeur expérimenté peut commencer à écrire des applications OpenXML simples en quelques heures après avoir lu la spécification.

Bien que la spécification décrive de nombreuses fonctionnalités, une application Open XML n'a pas besoin de prendre en charge toutes les fonctionnalités de la spécification. La déclaration de conformité (§1:2) impose qu'un consommateur conforme « ne rejette aucun document conforme au type [qu'il attend] » et qu'un producteur conforme « soit capable de produire des documents conformes » (§1:2.5). Elle fournit également des directives d'interopérabilité qui spécifient le rôle des sémantiques d'éléments (§1:2.6).

Une application conforme peut avoir des fonctionnalités extrêmement ciblées. Par exemple, elle pourrait comporter un module de traitement par lots qui se charge simplement de mettre à jour les avis de copyright dans un ensemble de documents de traitement de texte, ou un module de synthèse vocale qui comprend suffisamment une présentation pour en restituer son texte sous forme audio lorsque l'utilisateur navigue d'une diapositive à l'autre. La structure du format de fichier permet à de tels programmes d'être développés avec des connaissances minimales en OpenXML. Plus spécifiquement :

- Le format de fichier est conforme à des standards bien établis, notamment XML et ZIP, pour lesquels il existe des outils efficaces et reconnus.
- Le format de fichier utilise les conventions OPC (Open Packaging Conventions), qui combinent XML et ZIP avec des mécanismes standard pour exprimer des relations à l'intérieur d'un fichier. Pour cette raison, il est souvent possible de naviguer à l'intérieur du contenu d'un fichier sans connaître la sémantique des balises pour les langages de balisage principaux ou secondaires dans OpenXML.
- Des éléments profondément imbriqués dans l'arborescence XML sont accessibles et peuvent être modifiés sans perturber le reste de la structure.

De petits détails dans les formats de fichiers, dont certains étaient absents dans les formats binaires, prennent en charge des applications avec une fonctionnalité minimale en fournissant des valeurs mises en cache. Par exemple :

- Sans implémenter un module de pagination, une application telle qu'un lecteur pour malvoyants peut assurer la navigation entre les pages en utilisant les derniers sauts de page calculés (§4:2.3.3.13).
- Sans implémentation de formules ni intégration avec une source externe, un tableur peut travailler à partir de calculs mis en cache (§3:3.2.9) et données externes mises en cache (§4:3.14 et §4:3.10.1.76).

Un document conforme minimal est extrêmement simple ; voir la sous-section « Document WordprocessingML minimal ».

4.4 COMPRESSION

Le format de fichier OpenXML prend en charge la création d'applications hautes performances. Dans cette sous-section, nous décrivons certains points de conception permettant de produire un fichier compact, pour accélérer le traitement et l'analyse. Dans la sous-section suivante, nous montrons comment une structure de fichiers modulaire permet à une application d'accomplir de nombreuses tâches en analysant ou en modifiant uniquement un petit sous-ensemble d'un document.

Un fichier OpenXML est par convention stocké dans une archive ZIP à des fins de conditionnement et de compression, en respectant la mise en œuvre recommandée des conventions OPC. Cela peut sans doute paraître surprenant, mais les fichiers OpenXML sont environ 25 % (parfois jusqu'à 75 %) plus petits que leurs homologues binaires. Par exemple, ce livre blanc est 85 % plus volumineux dans le format binaire !

La longueur des identificateurs dans XML constitue une deuxième source simple de compression, notamment lorsqu'une représentation non compressée est requise. Les noms de balises fréquemment utilisées sont courts. Les personnes chargées de l'implémentation sont également encouragées à utiliser des préfixes d'espaces de noms courts ; par exemple, le préfixe conventionnel de l'espace de noms WordprocessingML est « w ».

Un autre niveau de compression est obtenu en évitant les répétitions dans le format du fichier. Une catégorie d'exemples supprime ainsi le stockage redondant d'objets volumineux.

- Dans SpreadsheetML, les chaînes répétées sont stockées dans un tableau de chaînes dans le classeur, puis référencées par un index (§3:3.3).
- Dans SpreadsheetML, une formule qui est remplie vers le bas et latéralement sur plusieurs cellules est stockée en tant que formule « maître » dans la cellule du coin supérieur gauche ; les autres cellules dans la plage de remplissage s'y réfèrent par un index de regroupement (§3:3.2.9.2).
- Dans DrawingML, les noms de forme (§4:5.1.12.56), les géométries de texte (§4:5.1.12.76) et autres valeurs prédéfinies (plusieurs dans §3:5.8, §3:5.9 et §4:5.1.12) sont représentés par noms ou numéros, et non désignés explicitement. Dans ces cas, les significations des noms et des numéros résident dans la spécification, pas dans le fichier. Ici, la représentation choisie est le résultat d'un compromis explicite retenu lors du processus de normalisation. Elle est compacte et permet la modification au niveau d'abstraction approprié : par exemple, un rectangle peut être changé en ovale par simple modification d'un attribut (§4:5.1.11.18).

Dans une autre catégorie d'exemples, une hiérarchie sert à fournir une sémantique d'héritage. Cela permet une amélioration des performances par une réduction de la taille des fichiers.

- Dans WordprocessingML, les styles sont hiérarchiques (§3:2.8.9).
- Dans DrawingML, les formes sont hiérarchiquement regroupées (§4:5.1.2.1.20).
- Dans PresentationML, une hiérarchie par défaut établit des relations entre des matrices de diapositives, des dispositions de diapositives et des diapositives (§3:4.2).

D'autres aspects d'OpenXML sont également conçus pour permettre une implémentation efficace. Par exemple, dans SpreadsheetML, le tableau des cellules stocke seulement des cellules non vides et est capable de représenter des cellules fusionnées en tant qu'unité. L'économie permise par cette technique s'avère significative pour l'analyse des feuilles de calcul.

4.5 MODULARITÉ

Une application peut accomplir de nombreuses tâches en analysant ou en modifiant un petit sous-ensemble du document.

Trois fonctions du format OpenXML coopèrent pour fournir cette modularité.

- Un document n'est pas monolithique ; il est composé de plusieurs parties.
- Les relations entre les parties sont elles-mêmes stockées dans les parties.
- Le format d'archive ZIP qui est généralement utilisé pour prendre en charge les documents OpenXML permet un accès aléatoire à chaque partie.

Par exemple :

- Une application peut déplacer sans difficulté une diapositive d'une présentation à une autre, avec des ressources telles que des images et des dispositions, entièrement sans analyse du contenu des diapositives (§3:13.3.8). Cet exemple utilise des données qualifiées de relations explicites pour identifier la diapositive et ses ressources. Des relations explicites sont définies par les conventions OPC et peuvent être analysées sans connaissance de la sémantique des balises PresentationML (§1:9.2, §2:8.3).
- Une application peut retirer tous les commentaires d'un document WordprocessingML sans analyser son contenu (§1:11.3.2). Cet exemple utilise des données qualifiées de relations implicites pour trouver les commentaires. Les relations implicites sont spécifiques d'OpenXML et nécessitent par conséquent une connaissance du langage de balisage concerné (§1:9.2).

4.6 MIGRATION HAUTE FIDÉLITÉ

OpenXML est conçu pour prendre en charge toutes les fonctions des formats binaires Microsoft Office 97-2003.

Il faut souligner les difficultés qu'implique cet objectif et l'efficacité singulière d'OpenXML à cet égard. Certains formats, tels que PDF, sont conçus pour livrer une représentation visuelle d'un document final à l'utilisateur. Par opposition, OpenXML a pour but de permettre une édition ou une manipulation ultérieures au même niveau d'abstraction que son créateur d'origine ; par exemple, la réduction d'un graphique vectoriel en un bitmap ne répondrait pas à cet objectif, comme la réduction d'une hiérarchie de styles en styles indépendants. En outre, un document peut contenir une sémantique de calcul que le créateur d'origine entend préserver, par exemple une logique de formule qui dépend de résultats de calculs intermédiaires, notamment des codes d'erreur ou des règles d'animation produisant un comportement dynamique.

Ces références à la spécification illustrent la capacité d'OpenXML de représenter de subtils aspects des formats binaires.

- La description SpreadsheetML inclut une spécification de formules étendue (§4:3.17.7).
- La spécification WordprocessingML documente les règles par lesquelles des propriétés de paragraphes, de caractères, de numérotation et de tableaux sont composées avec un formatage direct (§3:2.8, notamment §3:2.8.10).
- La spécification PresentationML documente les fonctions d'animation (§3:4.4).

OpenXML permet de se conformer à plusieurs implémentations sans devoir appliquer une parfaite correspondance dans chaque détail sous-jacent. Cela est particulièrement important lorsque des calculs numériques sont impliqués, par exemple la disposition, le rendu d'effets et l'évaluation des formules. Imposer un niveau de cohérence impraticable créerait un obstacle inutile pour les développeurs dans la mise en œuvre de la conformité souhaitée. Ces remarques mettent en lumière certaines décisions prises par le comité à cet égard.

- OpenXML définit des effets tels que les aspects de surface (§5.1.12.50) sans contraindre un développeur à respecter une correspondance pixel par pixel pour ces effets.
- OpenXML définit des paramètres tels que les marges des pages (§4:2.6.11), la police (§4:2.8) et la justification (§4:2.3.1.13). Il permet aux développeurs d'implémenter des algorithmes de flux différents pour autant qu'ils respectent ces paramètres.
- La spécification des formules SpreadsheetML (§4:3.17.7) ne tente pas de supprimer les variantes de calcul à virgule flottante car, en général, cela obligerait les applications conformes à mettre en œuvre une émulation lente au lieu de s'appuyer sur un matériel natif. Elle spécifie plutôt le degré minimal de précision des calculs numériques (§4:3.17.5).
- La spécification des formules SpreadsheetML permet de prendre certaines décisions conditionnelles au moment de l'implémentation afin de favoriser de futures innovations. Par exemple, elle ne limite pas le nombre maximal d'itérations d'un calcul comme NORMINV (§4:3.17.7.227). La fonction NORMINV effectue l'inverse de la distribution normale en effectuant une recherche itérative.

De nombreuses anciennes fonctions, telles que VML (§3:6), sont incluses principalement à des fins de compatibilité en amont. L'utilisation de nouveaux standards déjà présents dans OpenXML, tels que DrawingML (§3:5), est encouragée lors de l'écriture de nouveaux documents.

4.7 FACILITÉ D'INTÉGRATION DES DONNÉES

OpenXML permet aux organisations d'intégrer des applications bureautiques dans les systèmes d'information qui gèrent les processus d'entreprise en permettant l'utilisation de schémas personnalisés dans des documents OpenXML. En adoptant cette approche, les objectifs d'une organisation sont de réutiliser et d'automatiser le traitement des données qui seraient autrement enfouies de façon opaque dans des documents illisibles et non modifiables par les applications bureautiques.

Les applications incluent les fonctions suivantes :

- *Recherche* : un utilisateur final peut rechercher parmi un ensemble de feuilles de calcul les entreprises présentant des marges bénéficiaires supérieures à 20 %.
- *Balilage des métadonnées* : une entreprise peut baliser des présentations ayant été approuvées dans une perspective réglementaire.
- *Assemblage de documents* : un groupe de proposition peut simplifier la génération d'initiatives en automatisant la préparation des données sous-jacentes.
- *Réutilisation des données* : un directeur des ventes peut générer un rapport de tous les contrats commerciaux conclus dans une plage de dates donnée, répertoriant les clients, la taille des contrats, ainsi que les termes et conditions éventuellement modifiés.
- *Applications métier* : les professionnels d'un secteur spécialisé peuvent préparer dans un environnement de création familier des travaux qui sont ensuite automatiquement injectés dans les systèmes de l'entreprise.

La réalisation de ces objectifs nécessite la définition de la structure et du type de données qu'une catégorie de documents peut contenir, et la garantie que les informations peuvent être librement révélées lorsqu'elles apparaissent naturellement dans le flux de chaque document. Prenez l'exemple simple d'un CV. Une personne pourrait définir une structure de données incluant des champs nommés nom, numéro de téléphone, adresse, objectifs de carrière et qualifications. Une autre personne pourrait prévoir que ces champs apparaissent lorsqu'un auteur les place dans un document. Dans un contexte d'entreprise différent, celui par exemple d'un groupe financier ou d'un centre médical, la structure et les champs de données pourraient être différents.

OpenXML permet qu'un tel processus s'exécute de manière normalisée.

D'abord, la structure des données d'entreprise est exprimée au moyen d'un schéma XML personnalisé. Cela permet à une organisation d'exprimer les données avec des balises significatives dans une perspective professionnelle. Une organisation peut créer ses propres schémas, ou utiliser des schémas industriels standardisés tels que XBRL pour des rapports financiers (7) et HL7 pour des informations médicales (8). Des schémas sont créés dans le secteur public, dans des entreprises, et en tant que standards industriels, à diverses fins, allant des certificats de naissance aux informations relatives aux assurances. Tout schéma personnalisé peut être utilisé à condition d'être exprimé sous forme XSD (2).

Ensuite, les données personnalisées sont incorporées dans un document OpenXML dans une partie XML personnalisée (§3.7.3) et peuvent être décrites au moyen d'une partie de propriétés de données XML personnalisée (§4:7.5). En séparant ces données personnalisées de la présentation, OpenXML permet l'intégration de données propres, tout en autorisant une présentation et une manipulation par l'utilisateur final dans une large variété de contextes, notamment des documents, des formulaires, des diapositives et des feuilles de calcul. L'interopérabilité peut donc être obtenue à un niveau plus fondamental et avec une grande précision sémantique.

4.8 POSSIBILITÉS D'INNOVATION

OpenXML est conçu pour encourager les développeurs à créer de nouvelles applications qui n'étaient pas envisagées lorsque les formats binaires ont été définis, ou bien même lorsque OpenXML a été défini initialement.

Commençons par décrire les mécanismes d'extension qui permettent ensemble l'interopérabilité entre des applications dotées de jeux de fonctionnalités différents. Prenez une application *de niveau supérieur* (qui contient une nouvelle fonction non documentée dans OpenXML) et une application *de niveau inférieur* (qui ne comprend pas cette fonction). Les trois principaux objectifs de l'extension sont les suivants :

- *Fidélité visuelle* : possibilité pour l'application de niveau inférieur d'afficher ce que l'application de niveau supérieur afficherait. Cela impose qu'un fichier stocke plusieurs représentations des mêmes données.
- *Possibilité de modification* : possibilité de modifier une ou plusieurs des représentations.
- *Confidentialité* : possibilité de garantir que les anciennes versions d'une représentation ne subsistent pas après la modification d'une autre représentation, laissant ainsi contre toute attente des informations qu'un utilisateur croyait avoir supprimées ou modifiées. Une application peut atteindre cet objectif en éliminant ou en synchronisant des représentations.

Un développeur souhaitant étendre le jeu de fonctions OpenXML a deux options :

- Blocs de contenu de remplacement : un bloc de contenu de remplacement (§3:2.18.4 et §5:9.2) stocke plusieurs représentations du même contenu, chacun dans un bloc spécifique. Une application de niveau inférieur lit un bloc spécifique qu'elle est capable de lire. Lors d'une modification, elle écrit autant de blocs spécifiques qu'elle est capable d'écrire.
- Listes d'extensions : une liste d'extensions (§3:2.6) stocke des données XML personnalisées arbitraires sans représentation visuelle.

Les développeurs ont la possibilité d'innover en dehors de ces mécanismes d'extension.

- *Autres paradigmes d'interaction.* OpenXML va au-delà de la syntaxe d'un document, mais en deçà du comportement d'une application. Tel que décrit dans la déclaration de conformité, la spécification se concentre sur la sémantique (§1:2.2, §1.2.3). Par conséquent, une application conforme peut communiquer avec un utilisateur par divers moyens, ou ne pas communiquer du tout avec un utilisateur final, pour autant qu'elle respecte la sémantique spécifiée.
- *Environnements informatiques de Novel.* La déclaration de conformité admet des applications de faible capacité, afin qu'elles puissent s'exécuter sur de petites unités, ainsi que des applications ne mettant en œuvre qu'un sous-ensemble d'OpenXML (§1:2.6). Le mécanisme Caractéristiques supplémentaires permet à une application productrice de communiquer ses limites de capacité (§3:8.1).

Comme l'indiquait la sous-section précédente, certaines opportunités d'innovation les plus intéressantes n'impliquent pas le rendu de documents pour une interaction directe avec l'utilisateur. Elles impliquent plutôt un traitement de machine à machine en utilisant des formats de messages XML, telles que les services Web XML (9). Bien que de telles applications n'aient pas de comportement visible par l'utilisateur autre que leurs opérations sur les données contenues dans les documents OpenXML, elles sont soumises aux impératifs de conformité des documents (§1:2.4) et de conformité des applications (§1:2.5), qui sont purement syntaxiques, et aux directives d'interopérabilité (§1:2.6), qui incorporent de la sémantique.

Bien qu'il soit impossible d'énumérer tous les cas d'utilisation possibles du traitement XML personnalisé, il est possible d'imaginer des services centrés sur XML pouvant traiter des documents OpenXML pour une extraction et une insertion automatiques de données personnalisées, des services de sécurité personnalisés tels que la signature numérique XML (10) ou le chiffrement XML (11), voire des transformations XSLT arbitraires (12) assurant la conversion entre formats XML. OpenXML ne prévoit aucune interdiction ni restriction sur un tel traitement.

5 STRUCTURE D'UN DOCUMENT OFFICE OPEN XML

L'un des objectifs principaux de ce livre blanc est de permettre au lecteur de suivre la structure évoluée de tout fichier OpenXML. Pour atteindre cet objectif, nous fournissons un niveau modéré de détail concernant les conventions OPC (Open Packaging Conventions), et des informations moins détaillées sur les langages de balisage individuels.

5.1 CONVENTIONS OPC (OPEN PACKAGING CONVENTIONS)

Les conventions OPC (Open Packaging Conventions) permettent de stocker plusieurs types de contenus (par exemple, du code XML, des images et des métadonnées) dans un conteneur, par exemple une archive ZIP, pour représenter entièrement un document. Ces conventions décrivent un modèle logique pour la représentation d'un confinement et de relations.

L'implémentation recommandée des OPC utilise le format d'archive Zip. Il est possible d'inspecter la structure de tout fichier OpenXML au moyen d'un afficheur ZIP. Il convient d'inspecter le contenu d'un petit fichier OpenXML de cette manière lors de la lecture de cette description. Sous le système d'exploitation Windows, il faut seulement ajouter une extension « .zip » au nom de fichier et de double-cliquer dessus.

Logiquement, un document OpenXML est un *package* OPC (§5:8). Un package est un ensemble plat de *parties* (§5:8.1). Chaque partie possède un *nom de partie* qui ne respecte pas la casse et qui consiste en une séquence de noms de segments délimités par des barres obliques, comme dans « /pres/slides/slide 1.xml » (§5:8.1.1). Chaque partie a également un *type de contenu* (§5:8.1.2). Physiquement, l'archive ZIP est un package, chaque élément ZIP dans l'archive est une partie, et les noms de chemins dans l'archive ZIP correspondent directement aux noms des parties.

Dans l'implémentation ZIP, « /[Content_Types].xml » permet à un consommateur de déterminer le type de contenu de chaque partie du package (§2:9.2.6). La syntaxe et la définition des types de supports sont conformes à la section 3.7 du RFC 2616 (13).

Les packages et les parties peuvent contenir des *relations explicites* (§1:9.2) à d'autres parties dans le package, ainsi qu'à des ressources externes. Chaque relation explicite a un ID de relation, ce qui permet au contenu d'une partie de s'y référer ; et un type, qui permet à une application de décider comment la traiter. Les types de relations sont nommés à l'aide d'URI, permettant à des parties sans coordination entre elles de créer en toute sécurité de nouveaux types sans conflit.

L'ensemble de relations explicites pour un package ou une partie source donné est stocké dans une *partie de relations*. La partie de relations du package dans son ensemble se nomme « /_rels/.rels » ; la partie de relations pour une partie nommée « /a/b/c.xml » est appelée « /a/b/_rels/c.xml.rels ». Les parties de relations (et, dans l'implémentation ZIP, la partie de type de contenu) sont les seules parties nommées spécifiquement dans un package. Pour ouvrir un package, une application doit analyser la partie des relations du package et suivre les relations du type approprié.

Toutes les autres parties d'un document OpenXML contiennent du contenu OpenXML, du contenu XML personnalisé, ou du contenu de type arbitraire tels que des objets multimédias. La possibilité d'une partie de comporter du contenu XML personnalisé constitue un mécanisme particulièrement puissant pour l'incorporation de données d'entreprise et de métadonnées.

5.2 WORDPROCESSINGML

Un document WordprocessingML est composé d'une collection de « storys » (§3:2.1). Les différents storys sont les suivants : le document principal (§3:2.2), le document glossaire (§3:2.13), un sous-document (§3:2.18.2), un en-tête (§3:2.11.1), un pied de page (§3:2.11.2), un commentaire (§3:2.14.5), un cadre, une zone de texte (§3:2.18.1), une note de bas de page (§3:2.12.1) ou une note de fin (§3:2.12.2).

Le seul story obligatoire est le document principal. C'est la cible de la relation du package dont le type est :

<http://schemas.openxmlformats.org/officeDocument/2006/relationships/officeDocument>

Un chemin typique de la racine à la feuille dans l'arborescence XML comprendrait ces éléments XML (§3:2.2) :

- `document` — l'élément racine du document principal (§3:2.3).
- `body` — corps (§3:2.7.1). Peut contenir plusieurs paragraphes. Peut également contenir des propriétés de section spécifiées dans un élément `sectPr`.
- `p` — paragraphe (§3:2.4.1). Peut contenir un ou plusieurs « runs » (séries de caractères ayant les mêmes propriétés). Peut également contenir des propriétés de paragraphe spécifiées dans un élément `pPr`, qui à son tour peut contenir des propriétés de « run » par défaut (également qualifiées de propriétés de caractères) spécifiées dans un élément `rPr` (§3:2.4.4).
- `r` — run (§3:2.4.2). Peut contenir plusieurs types de contenus de run, principalement des plages de texte. Peut également contenir des propriétés de run (`rPr`). Le run est un concept fondamental dans OpenXML. Un run est un ensemble contigu de texte ayant des propriétés identiques ; un run ne contient pas de balisage de texte supplémentaire. Par exemple, si une phrase contient les mots « voici **trois** runs », elle serait représentée par au moins trois runs : « voici », « **trois** » et « runs ». À cet égard, OpenXML diffère considérablement des formats qui permettent l'imbrication arbitraire de propriétés, tels que HTML.
- `t` — plage de texte (§3:2.4.3.1). Contient une quantité arbitraire de texte sans mise en forme, sauts de ligne, tableaux, graphiques ou autres éléments non-texte. La mise en forme du texte est héritée des propriétés de run et des propriétés de paragraphe. Cet élément utilise souvent l'attribut `xm1:space="preserve"`.

Dans cette sous-section, nous avons traité la mise en forme directe du texte en spécifiant des propriétés de paragraphe et de run. La mise en forme directe se situe à la fin d'un ordre d'application qui inclut également des styles de caractères, de paragraphes, de numérotation et de tableaux, ainsi que des valeurs par défaut de documents (§3:2.8.10). Ces styles sont eux-mêmes organisés en hiérarchies d'héritage (§3:2.8.9).

La sous-section « Document WordprocessingML minimal » ci-dessous présente intégralement un document WordprocessingML.

5.3 PRESENTATIONML

Un document PresentationML est décrit par une partie de présentation. La partie de présentation est la cible de la relation de package dont le type est :

<http://schemas.openxmlformats.org/officeDocument/2006/relationships/officeDocument>

La présentation se réfère à ces structures principales (§3:4.2), que nous répertorions de haut en bas dans la hiérarchie par défaut :

- matrices de diapositives, matrices de notes et matrices de document (§3:4.2.2), toutes héritant de propriétés de la présentation ;
- dispositions de diapositive (§3:4.2.5), qui héritent de propriétés de la matrice de diapositive ; et
- diapositives (§3:4.2.3) et pages de note (§3:4.2.4), qui héritent respectivement des propriétés de dispositions de diapositives et de matrices de notes.

Chaque matrice, disposition et diapositive est stockée dans sa propre partie. Le nom de chaque partie est spécifié dans la partie de relation pour la partie de présentation. Chacune des six parties autres que la présentation est structurée essentiellement de la même manière. Un chemin typique de la racine à la feuille dans l'arborescence XML comprendrait les éléments XML (§3:2.2) suivants :

- `sld`, `sldLayout`, `sldMaster`, `notes`, `notesMaster`, ou `handoutMaster` — l'élément racine.
- `csld` — diapositive (§4:4.4.1.15). Peut contenir des éléments DrawingML (tels que décrits dans les deux paragraphes suivants) ainsi que d'autres éléments structuraux (tels que décrits ci-dessous).
- `spTree` — arborescence de forme (§4:4.4.1.42). Peut contenir des propriétés de forme de groupe dans un élément `grpSpPr` (§4:4.4.1.20), ainsi que des propriétés de forme de groupe non visuelles dans un élément `nvGrpSpPr` (§4:4.4.1.28). Ce nœud et ses descendants sont tous des éléments DrawingML. Nous répertorions ici certains éléments DrawingML en raison de leur rôle primordial dans PresentationML.
- `sp` — forme (§4:4.4.1.40). Peut contenir des propriétés de forme dans un élément `spPr` (§4:4.4.1.41) ainsi que des propriétés de forme non visuels dans un élément `nvSpPr` (§4:4.4.1.31).

En complément du contenu de forme DrawingML, un `csld` peut contenir des éléments structuraux, selon l'élément racine dans lequel il réside, tel qu'indiqué dans ce tableau :

	Diapositive	Disposition de diapositive	Matrice de diapositive	Matrice de document	Matrice de notes	Page de notes
Données communes	X	X	X	X	X	X
Transition	X	X	X			
Synchronisation	X	X	X			
En-têtes et pieds de page		X	X	X	X	
Nom correspondant		X				
Type de disposition		X				
Préserver		X	X			
Liste de dispositions			X			
Style de texte			X			

Les propriétés spécifiées par des objets plus bas dans la hiérarchie par défaut (matrice de diapositive, disposition de diapositive, diapositive) remplacent les propriétés correspondantes spécifiées par des objets plus haut dans la hiérarchie. Par exemple, si une transition n'est pas spécifiée pour une diapositive, elle est alors retirée de la disposition de diapositive ; si elle n'y est pas spécifiée, elle est retirée de la matrice de diapositive.

5.4 SPREADSHEETML

Un document SpreadsheetML est décrit au niveau supérieur par une partie de classeur. La partie de classeur est la cible de la relation de package dont le type est :

<http://schemas.openxmlformats.org/officeDocument/2006/relationships/officeDocument>

La partie de classeur stocke des informations sur le classeur et sa structure, comme la version du fichier, l'application de création et le mot de passe à modifier. Logiquement, le classeur contient une ou plusieurs feuilles (§3:3.2) ; physiquement, chaque feuille est stockée dans sa propre partie et elle est référencée de la manière habituelle à partir de la partie classeur. Chaque feuille peut être une feuille de calcul, une feuille de graphique ou une feuille de dialogue. Nous décrivons uniquement la feuille de calcul, qui est le type le plus courant. Dans un objet de feuille de calcul, un chemin classique de la racine à la feuille dans l'arborescence XML comprendrait les éléments XML suivants :

- `worksheet` — l'élément racine dans une feuille de calcul (§3:3.2).
- `sheet Data` — le tableau de cellules, qui représente chaque cellule non vide dans la feuille de calcul (§3:3.2.4).
- `row` — une ligne de cellules dans le tableau de cellules (§3:2.8).
- `c` — une cellule (§3:3.2.9). L'attribut `r` indique l'emplacement de la cellule à l'aide de coordonnées de type A1. La cellule peut également comporter un identificateur de style (attribut `s`) et un type de données (attribut `t`).
- `v` et `f` — la valeur (§3:3.2.9.1) et la formule facultative (§3:3.2.9.2) de la cellule. Si une cellule a une formule, la valeur est le résultat du calcul le plus récent.

Les chaînes et les formules sont stockées dans des tableaux partagés (§3:3.3 et §3:3.2.9.2.1) pour éviter un stockage redondant et accélérer les chargements et les enregistrements.

5.5 LANGAGES DE BALISAGE SECONDAIRES

Plusieurs langages de balisage secondaires peuvent être utilisés pour décrire le contenu d'un document OpenXML.

- DrawingML (§3:5) — utilisé pour représenter des formes et d'autres objets rendus graphiquement dans un document.
- VML (§3:6) — format de graphique vectoriel qui est inclus à des fins de compatibilité en amont et qui sera remplacé à terme par DrawingML.
- ML partagés : Math (§3:7.1), Métadonnées (§3:7.2), XML personnalisé (§3:7.3) et Bibliographie (§3:7.4).

5.6 DOCUMENT WORDPROCESSINGML MINIMAL

Cette sous-section contient un document WordprocessingML minimal composé de trois parties.

La partie content-type « /[Content_Types].xml » décrit les types de contenus des deux autres parties requises.

```
<Types xmlns="http://schemas.openxmlformats.org/package/2006/content-types">
  <Default Extension="rels"
    ContentType="application/vnd.openxmlformats-package.relationships+xml"/>
  <Default Extension="xml"
    ContentType="application/vnd.openxmlformats-officedocument.wordprocessingml.document.main+xml"/>
</Types>
```

La partie relation de package « /_rels/.rels » décrit la relation entre le package et la partie du document principal.

```
<Relationships xmlns="http://schemas.openxmlformats.org/package/2006/relationships">
  <Relationship Id="rId1"
    Type="http://schemas.openxmlformats.org/officeDocument/2006/relationships/officeDocument"
    Target="document.xml"/>
</Relationships>
```

La partie document, dans ce cas « /document.xml », contient le contenu du document.

```
<w:document xmlns:w="http://schemas.openxmlformats.org/wordprocessingml/2006/main">
  <w:body>
    <w:p>
      <w:r>
        <w:t>Hello, world.</w:t>
      </w:r>
    </w:p>
  </w:body>
</w:document>
```

La spécification fournit les documents minimaux et des détails supplémentaires pour WordprocessingML (§1:11.2), PresentationML (§1:13.2) et SpreadsheetML (§1:12.2).

6 RÉSUMÉ

OpenXML est le produit d'importants efforts consentis par des représentants de nombreux secteurs et établissements publics de divers horizons et intérêts. Il couvre l'ensemble des fonctions utilisées dans le corpus de documents existant, ainsi que les besoins d'internationalisation inhérents dans l'ensemble des principaux groupes linguistiques dans le monde entier. À la suite du travail de standardisation effectué par le comité Ecma TC45 (1) et des commentaires apportés par le public, OpenXML apporte un niveau élevé d'interopérabilité et d'indépendance vis-à-vis des plateformes. Sa documentation est désormais complète (par des documents de référence exhaustifs) et accessible (par des descriptions non normatives). Il comprend toutes les informations nécessaires pour que les programmes de technologie d'assistance permettent un traitement adéquat des documents. Les implémentations OpenXML peuvent être très réduites et fournir une fonctionnalité très ciblée, où elles peuvent englober le jeu complet des fonctionnalités. Des mécanismes d'extension intégrés dans le format garantissent les possibilités d'innovation.

La normalisation de la spécification du format et sa mise à jour dans le temps garantissent que de nombreux intervenants peuvent s'y fier, sachant que toute évolution ultérieure bénéficiera des contrôles et de l'équilibre qu'autorise un processus normalisé et ouvert. Il existe un réel besoin d'un standard de documents ouvert capable de préserver les milliards de documents ayant été créés dans les formats binaires préexistants, ainsi que les milliards qui continuent à être créés chaque année. Les récentes avancées technologiques en matériel, en réseau et en infrastructures logicielles standardisées rendent ces objectifs tout à fait réalisables. La grande diversification de la demande du marché, notamment les importants investissements consentis dans des systèmes d'entreprise stratégiques, rendent cette évolution essentielle.

7 CITATIONS

1. **Ecma International** TC45 - Office Open XML Formats. *Ecma International*. [En ligne] <http://www.ecma-international.org/memento/TC45.htm>.

2. **W3C**. XML Schema. *World Wide Web Consortium*. [En ligne] <http://www.w3.org/XML/Schema>.

3. **ISO**. ISO/IEC 19757-2:2003. *Organisation internationale de normalisation*. [En ligne] <http://www.iso.org/iso/en/CatalogueDetailPage.CatalogueDetail?CSNUMBER=37605&ICS1=35&ICS2=240&ICS3=30>.

4. **ISO**. ISO/IEC 19757-4:2006. *Organisation internationale de normalisation*. [En ligne] <http://www.iso.org/iso/en/CatalogueDetailPage.CatalogueDetail?CSNUMBER=38615&ICS1=35&ICS2=240&ICS3=30>.

5. **W3C**. Extensible Markup Language (XML) 1.0 (Fourth Edition). *World Wide Web Consortium*. [En ligne] 2006. <http://www.w3.org/TR/2006/REC-xml-20060816/>.

6. **W3C**. Namespaces in XML 1.0 (Second Edition). *World Wide Web Consortium*. [En ligne] 2006. <http://www.w3.org/TR/2006/REC-xml-names-20060816/>.

7. **XBRL International**. XBRL Specifications. *Extensible Business Reporting Language*. [En ligne] <http://www.xbrl.org/Specifications/>.

8. **Health Level Seven**. HL7 ANSI-Approved Standards. *Health Level Seven*. [En ligne] http://www.hl7.org/about/directories.cfm?framepage=/documentcenter/public/faq/ansi_approved.htm.

9. **W3C**. W3C Web Services Architecture. *World Wide Web Consortium*. [En ligne] 2002. <http://www.w3.org/2002/ws/>.

10. **W3C**. W3C XML Signature. *World Wide Web Consortium*. [En ligne] <http://www.w3.org/Signature/>.

11. **W3C**. W3C XML Encryption. *World Wide Web Consortium*. [En ligne] 2001.
<http://www.w3.org/Encryption/2001/>.
12. **W3C**. XSL and XSLT. *World Wide Web Consortium*. [En ligne]
<http://www.w3.org/Style/XSL/>.
13. **W3C**. Hypertext Transfer Protocol - HTTP/1.1. *World Wide Web Consortium*. [En ligne]
<http://www.w3.org/Protocols/rfc2616/rfc2616.html>.