

# Comprendre ebXML – la norme pour les échanges de données en commerce électronique

## Introduction

### Présentation

UN/CEFACT, OASIS et de nombreux acteurs du commerce électronique ont unis leurs efforts pour concevoir un nouveau standard pour le commerce électronique.

Loin de devoir remplacer EDIFACT, ebXML se positionne dans la complémentarité et dans la continuité. EDIFACT est particulièrement adapté aux échanges de gros volumes avec des partenaires stables, alors que ebXML doit répondre, en plus, à la problématique des petits échanges entre partenaires épisodiques.

De fait, en s'appuyant sur les technologies d'Internet et sur le métalangage XML, ebXML permet de couvrir deux domaines qui n'avait pas pu l'être avec EDIFACT : les échanges électroniques entre PME et / ou entre PME et grandes entreprises ; les échanges de données non textuelles

En utilisant les techniques de modélisation objet, les architectures distribuées et les outils de communication d'Internet, ebXML permet de concevoir des modèles d'échanges plus intégrés et plus réactifs.

Une des ambitions les plus affirmées de ebXML est de permettre aux plus grands nombre d'entreprises de bénéficier des avantages des EEP.

Les potentialités de ebXML couvrent l'ensemble des besoins de communication entre l'entreprise et ses différents partenaires. XML étant particulièrement flexible on peut imaginer aisément que nouvelles utilisations verront le jour dans les années à venir.

## Comprendre XML pour les échanges électroniques professionnels

### Présentation

XML ne se place pas en concurrent de l'EDI conventionnel. Ce dernier possède des caractéristiques qui le rendent particulièrement adapté aux échanges de gros volumes de messages entre partenaires stables. XML se situe plutôt en complément pour de nouvelles applications comme les liaisons avec les places de marché, le transfert de catalogues électroniques, l'utilisation de multimédia dans les messages... et pour une nouvelle tranche d'utilisateurs pour lesquels l'EDI conventionnel reste une technique trop lourde à mettre en œuvre, c'est-à-dire la grande majorité des PME. De plus, les initiatives GCIP et ebXML s'inscrivent dans un contexte mondial et permettent désormais des échanges avec les entreprises du continent nord-américain.

## 1. Pourquoi utiliser XML dans les échanges électroniques professionnels

### Presentation

Les échanges électroniques professionnels (EEP) peuvent utiliser différents langages tels qu'EDIFACT (Electronic Data Interchange for Administration, Commerce and Transport) ou EANCOM , ANSI X12 (American National Standard Institute X12) ou encore XML.

Toutes ces solutions permettent d'échanger des données entre systèmes d'information. Ces échanges supposent que les partenaires disposent :

- d'un minimum de règles d'utilisation et de syntaxe communes ;
- d'un vocabulaire unique ;
- des moyens techniques d'échanges compatibles.

Entre êtres humains, il n'est possible de communiquer que dans la mesure où :

- une même langue, c'est-à-dire une même grammaire et un même vocabulaire sont utilisés ;
- l'usage des sens adaptés à l'échange soient possédés (la parole et l'ouïe pour les échanges verbaux, la vue et la capacité d'écrire pour les échanges scripturaux) ;
- l'on se situe dans un environnement permettant l'échange (c'est-à-dire à portée de voix dans le cas des échanges verbaux et à portée de vue dans les échanges scripturaux).

Il en va de même pour les échanges entre systèmes d'information. Certains pré-requis sont nécessaires à la mise en œuvre de ces échanges.

## 1.1. Les pré-requis pour échanger entre systèmes d'information

### Présentation

Un langage commun est nécessaire, qu'il soit directement utilisable ou accessible par l'intermédiaire d'une opération de traduction.

Un langage est composé de mots (le vocabulaire) et de règles d'assemblage de ces mots pour obtenir des phrases (la grammaire).

Ressemblances des échanges de données avec les

## communications humaines

Dans le domaine des échanges de données entre systèmes d'information :

- les mots correspondent à des concepts de gestion plus ou moins globaux, par exemple :

- une date de livraison, un code article, une devise de paiement constituent des concepts simples ;

- une ligne article dans une commande, un en-tête de facture constituent des concepts complexes ;

- les phrases correspondent à des messages qui assemblent des mots dans des structures logiques proches de celles des documents papier que nous avons l'habitude d'utiliser. Par exemple :

- une commande sera composée d'un en-tête de commande, de lignes de commande et d'un pied de commande ;

- chacune de ces entités est composée de données plus petites comme pour l'en-tête de commande d'un numéro de commande, d'une date de commande, de l'adresse de l'émetteur de la commande et de celle du destinataire...

Les phrases échangées entre êtres humains s'enchaînent entre elles pour former une conversation. Dans le domaine des échanges électroniques professionnels, un échange complet de messages constitue un processus d'affaires.

Dans le cas de notre commande, ce processus est un processus d'approvisionnement en biens ou en services qui s'achève avec leur paiement.

## Différences des échanges de données par rapport aux communications humaines

S'il existe de nombreuses similitudes entre les concepts mis en œuvre dans les échanges humains et ceux des échanges électroniques, des différences

peuvent néanmoins être relevées.

Les systèmes d'information travaillent généralement avec des données codées : code article, code client ou fournisseur... Afin de communiquer, il est nécessaire que les deux systèmes utilisent des référentiels de communication similaires ou soient capables, par des moyens informatiques, de transcoder les codes d'un référentiel à un autre, c'est-à-dire de mettre en œuvre une table de correspondance permettant de changer une donnée par une autre donnée ayant la même valeur sémantique, mais une représentation différente.

Les systèmes d'information stockent et gèrent leurs données de gestion dans des bases de données qui peuvent avoir une structure différente de celle des messages. Il est donc nécessaire de disposer d'outils permettant de modifier la structure des données en émission comme en réception. Ainsi, ces données seront transformées de manière à être, en émission, dans la forme des messages et, en réception, dans la forme de la structure interne.

Les échanges électroniques professionnels sont généralement réalisés entre systèmes distants. Il est donc nécessaire donc d'utiliser des moyens de transmission des données entre les sites. Ces moyens intègrent des fonctions d'enveloppement, de routage, de sécurité, etc. qui permettent de garantir le bon acheminement des messages d'un émetteur identifié vers un destinataire connu.

Les éléments de sécurité à mettre en œuvre dépendent des risques que les partenaires souhaitent couvrir. Ils permettent d'assurer la confidentialité, l'intégrité et la non-répudiation des messages.

## 1.2. Présentation succincte d'EDIFACT

EDIFACT est un langage de communication entre systèmes d'information composé de règles de syntaxe et d'un vocabulaire.

La structure d'EDIFACT est décrite dans la figure suivante.

### Figure 1 - Les éléments du langage EDIFACT

La grammaire est définie par des règles ISO permettant la gestion de dictionnaires de données élémentaires ou composites, de segments de données ainsi que leur assemblage en message.

La syntaxe et le vocabulaire d'EDIFACT sont suffisamment stricts pour que des éditeurs de logiciels aient pu concevoir des produits permettant la génération et / ou le contrôle des messages échangés ainsi que leur transformation dans la structure des fichiers de l'entreprise (ou inversement). Ces logiciels sont appelés traducteurs EDIFACT.

Le transfert des messages vers les partenaires est généralement réalisé au moyen de réseaux à valeur ajoutée (RVA) qui assurent le routage des messages vers les différents partenaires de l'entreprise.

EDIFACT est un langage structuré qui convient parfaitement à des échanges réguliers et nombreux entre partenaires stables. EDIFACT n'est pas adapté à des échanges épisodiques entre partenaires irréguliers.

Par rapport à EDIFACT, l'utilisation d'XML pour les échanges entre systèmes d'information apporte différents avantages.

# 1.3. Les atouts d'XML

## Présentation

Le passage d'une économie "matérielle" à une économie "immatérielle" a des conséquences très importantes sur la vie des entreprises et leurs évolutions. L'enjeu stratégique se situe au niveau du partage de l'information. Tous les domaines sont concernés, qu'il s'agisse :

- du Business to Consumer (BtoC), avec l'augmentation de l'efficacité du processus de vente ;
- du Business to Business (BtoB), avec l'augmentation de la réactivité ;
- du Business to Administration (BtoA), pour faciliter le respect des législations en termes de paiement, sécurité et douane.

Avec HTML, la possibilité de naviguer sur Internet et de visualiser des documents a marqué l'avènement d'un nouveau média reconnu par tous comme une véritable révolution dans l'accès à l'information.

Point de jonction des technologies Internet et EDI, le Web EDI apporte aujourd'hui une première réponse aux attentes des entreprises souhaitant mettre en place des échanges commerciaux avec leurs différents partenaires, qu'ils soient fournisseurs ou clients.

XML vient faciliter et simplifier les échanges interapplications. Combiné à Internet, le réseau universel qui permet une connectivité globale de tous les partenaires potentiels, XML va sans doute provoquer une révolution semblable à celle déclenchée par HTML.

En effet, le champ d'application d'XML s'étend du simple logiciel, tels que traitement de texte ou tableur pour la gestion de documents, à des échanges de données informatisés entre applications internes (EAI ou Entreprise Application Integration) et des échanges entre systèmes d'information externes aux

entreprises ou aux organisations.

XML offre non seulement à l'utilisateur la possibilité de définir l'information et sa présentation, mais également la description de chaque information (métadatas) : article, identification, quantité, date de livraison... Il permet ainsi d'enrichir ou de modifier la syntaxe et la sémantique de structuration des documents. Il permet en outre de transmettre cette description.

En d'autres termes, XML rend les documents intelligents.

### 1.3.1. Un standard qui concerne le plus grand nombre

La liste des utilisateurs concernés par XML est longue, qu'il s'agisse des PME ou des grands comptes, en passant par les fournisseurs, les fabricants, les industriels, les partenaires, les détaillants, les distributeurs, les sociétés de VPC, l'administration et les clients.

XML est un nouveau standard de présentation et d'échanges d'information sur Internet. Les travaux de standardisation de l'utilisation de XML pour les échanges de données sont conduits dans les mêmes organisations que celles qui avaient élaborées les standards de l'EDI.. Ainsi, les travaux effectués depuis plus de vingt ans seront pérennisés et les entreprises utilisant déjà l'EDI, évolueront plus facilement vers ce nouveau standard, lorsqu'elles le jugeront opportun.

### 1.3.2. Un vocabulaire libre et intelligible

Les techniques de l'Internet évoluent en permanence. Parmi les évolutions les plus significatives de ces dernières années, le développement du langage XML est sans doute la plus importante pour le commerce en ligne.

Le langage qui précède XML est HTML, utilisé par la majorité des sites pour la réalisation des pages qui sont affichées à l'internaute. Mais HTML n'est qu'un langage de présentation.

XML apporte une dimension supplémentaire qui sera utilisée dans les échanges de données. Au moyen de balises, il permet en effet de qualifier le contenu informationnel des données d'une page.

L'expression <nom> Toto </nom>, par exemple, indique que la chaîne de caractères " Toto " est un nom. À partir de cette possibilité, il est facile de construire des messages équivalents aux messages de l'EDI conventionnel.

### 1.3.3. Une syntaxe flexible et communicable

Les messages XML peuvent être contrôlés à partir de structures de message sur lesquelles les partenaires de l'échange se sont mis d'accord. Ces structures, utilisant un vocabulaire librement défini, constituent des règles de contrôle de la syntaxe des messages.

Non seulement, les utilisateurs peuvent ainsi établir la définition de leur propre vocabulaire, mais ils peuvent également définir leur propre grammaire et vérifier la conformité des données transmises.

### 1.3.4. Des capacités supplémentaires par rapport à l'EDI conventionnel

Par rapport à l'EDI conventionnel, XML facilite :

- la transmission des données multimédia (image et vidéo pour l'illustration de catalogues en ligne, par exemple) ;
- la présentation des données sous une forme affichable par l'utilisation de feuille de style (pour les entreprises qui ne voudraient pas intégrer les

messages directement dans leurs systèmes d'information) ;

- une conversion facile d'une structure de message dans une autre (facilitant ainsi l'intégration des données dans des applications existantes), avec l'utilisation des technologies Internet de moindre coût.

C'est la raison pour laquelle XML est considéré comme un langage d'avenir pour le développement massif des échanges électroniques, qu'ils s'agissent d'échanges entre grandes entreprises, entre grandes entreprises et PME, ou entre PME.

Figure 2 : Comparaison entre XML et l'EDI conventionnel

## Des capacités supplémentaires par rapport à l'EDI conventionnel (2)

La structure d'un message XML est décrite dans un fichier spécifique (DTD ou Schéma XML). Ce fichier peut être utilisé par les partenaires pour contrôler la structure du message. Dans l'EDI conventionnel, cette structure est décrite dans un Guide d'utilisation de message, elle doit être paramétrée dans un traducteur pour que les contrôles en émission et en réception puissent être exécutés. C'est une des raisons de la souplesse d'utilisation d'XML par rapport à l'EDI conventionnel.

Dans les messages XML, l'utilisation de balises significatives est préconisée, c'est-à-dire de balises qui donnent un nom en clair à la donnée transportée. Cette recommandation facilite la lecture " humaine " du message. Si cela n'est pas le cas, on utilise un schéma d'interopérabilité qui fait le lien entre une balise et sa définition sémantique. Cette technique permet d'envisager l'utilisation de balises significatives dans les langages humains différents tout en assurant la lisibilité des messages par les différents systèmes d'information. Dans l'EDI conventionnel, ces fonctions sont assurées par le Guide d'utilisation de message.

Les messages XML seront naturellement transportés par le réseau Internet et les protocoles qui lui sont associés (protocoles IP). Internet n'étant pas

fortement sécurisé, les protocoles utilisés doivent intégrer des couches supplémentaires pour assurer des niveaux de sécurité compatibles avec les échanges professionnels. L'EDI conventionnel utilise les techniques éprouvées des réseaux à valeur ajoutée et du protocole X.400. Il est enfin possible de transporter des messages XML avec les outils de l'EDI conventionnel.

Les messages XML intègrent des balises " en clair ". Ces messages sont, de ce fait, beaucoup plus volumineux que ceux des messages de l'EDI conventionnel.

Les techniques liées aux échanges sont standardisées pour les messages XML car ces échanges utilisent une infrastructure déjà existante : celle d'Internet. Ce n'est pas le cas pour l'EDI conventionnel où les communautés peuvent décider du choix des techniques d'échanges propres à la communauté.

Les données métier de l'EDI conventionnel sont standardisées au sein du langage EDIFACT. Cela peut ne pas être vrai pour les messages XML. Il convient, en effet, de faire la distinction entre l'utilisation d'XML dans un cadre normalisé et son utilisation dans un cadre non normalisé. Dans un cadre normalisé, tel celui d'ebXML, des répertoires sémantiques pour les données d'affaires sont créés et permettent d'obtenir un niveau de précision identique à celui des échanges de l'EDI conventionnel.

## 1.3.5. Une utilisation universelle

### Présentation

Les technologies de l'Internet permettent d'accéder simplement et à moindre coût à l'information au moyen des navigateurs Web. Ainsi, les entreprises les plus petites ont accès à l'EDI.

Le même type de document pourra ainsi être :

- échangé en XML entre le système d'information d'un donneur d'ordres et le système d'information du partenaire ;

- consulté, créé ou modifié directement dans le système d'information de ce donneur d'ordres par le partenaire. Celui-ci utilise dans ce cas un accès HTTP au moyen d'un navigateur Web ou d'un outil bureautique, tel qu'un traitement de texte ou un tableur.

La deuxième solution permet d'échanger avec les plus petits partenaires. Ceux-ci ont la possibilité de visualiser des informations (heure d'arrivée, incidents, etc.) ou d'injecter des documents dans le système à l'aide de formulaires. Les données destinées à ces petits partenaires pourront être échangées en HTML ou en XML.

L'échange de données XML implique que le partenaire utilise un logiciel capable de traiter des données XML : un navigateur ou un outil bureautique ou encore un logiciel de gestion adapté.

## Schema

Le schéma suivant montre comment une entreprise peut, à partir d'une même application, entrer en relation avec des partenaires en EDI conventionnel et des partenaires qui sont équipés de systèmes plus simples.

Dans cet exemple, le système unique est situé du côté du vendeur. De la même manière, il pourrait être situé du côté de l'acheteur qui verrait alors tous ses fournisseurs avec une même interface.

Figure 3 : EDI conventionnel et EFI

## Transformation des documents

Un même document XML peut donc être transformé pour être utilisé de différentes manières. Il peut être affiché, par exemple, sur un navigateur Web, un assistant personnel ou un téléphone mobile. Il peut également être directement intégré dans un logiciel de gestion adapté.

Ces transformations peuvent être des enrichissements de documents, des combinaisons de documents ou des filtres. Elles peuvent s'appliquer au niveau d'un serveur Web ou d'un serveur d'applications, par des programmes accédant aux documents XML par le biais d'un parseur ou par l'intermédiaire de feuilles de style.

Ces traitements peuvent être appliqués au niveau du serveur, et / ou au niveau du client. Le traitement au niveau du client nécessite une configuration informatique dotée d'un navigateur récent.

Ce système de transformations permet d'associer à un même document XML une série de transformations à appliquer. Ainsi, un même document peut être transformé en fonction :

- de filtres pour ne sélectionner que les données qui intéressent le destinataire (certaines informations ne sont pas traitées par le système du destinataire car elles ne lui sont pas utiles ou ne peuvent pas être traitées) ;
- de filtres pour ne sélectionner que les données correspondant à la qualité de service souscrite par le destinataire (seules les données utiles au système d'information du destinataire sont sélectionnées) ;
- de filtres pour ne sélectionner que les données correspondant aux habilitations du destinataire (dans le cas d'une organisation à plusieurs niveaux, il est possible que certains niveaux aient accès à des données que les autres niveaux n'ont pas à connaître) ;
- de tris en fonction des choix du destinataire (le message est éclaté en fonction des besoins des utilisateurs, les données commerciales sont dirigées vers le service commercial, les données techniques au service de fabrication... ) ;
- d'un enrichissement permettant d'ajouter des données manquantes (cette fonction est souvent utilisée pour rendre explicite des données implicites. Par exemple, les valeurs monétaires n'ont pas de code devise car implicitement la devise de l'entreprise est l'Euro) ;
- d'une mise en forme particulière (pour une adaptation aux besoins ou habitudes de l'utilisateur) ;
- ...

Ces différents traitements peuvent être appliqués en même temps ou de manière consécutive en fonction des destinataires du message.

Toutes ces possibilités peuvent être mises en œuvre pour constituer un système d'accès aux données, permettant de prendre en compte les critères de sécurité (habilitations), de souscription de services et d'intérêt ou de choix personnels.

Cette dernière utilisation des transformations pourra être mise en œuvre au moyen d'un système de profils d'utilisation. Dans ce cas, un outil " éditeur de profil " est utilisé pour définir les profils. Chaque utilisateur dispose de plusieurs profils pour un même document. Il peut ainsi visualiser les données qui l'intéressent, celles auxquelles il a droit ou auxquelles il a souscrit. Elles sont mises en forme selon ses souhaits et ses besoins.

## 1.4. L'environnement Internet

### Présentation

L'utilisation d'XML pour les échanges électroniques professionnels s'inscrit dans la mouvance Internet et bénéficie de tous les outils qui y sont accessibles.

Les outils d'Internet à destination du grand public sont souvent gratuits ou disponibles à des prix bas. Les initiatives utilisant XML pour les échanges électroniques s'appuient sur les outils techniques d'Internet pour la sécurisation et le transfert des messages, pour leur affichage lorsque cela est nécessaire et pour leur transformation dans des formats exploitables par le système d'information de l'entreprise.

Parmi tous ces outils, ceux concernant la sécurisation des échanges sont certainement les plus importants. Ils permettent, par la mise en œuvre de procédures de signature numérique et de chiffrement, de garantir l'authenticité des partenaires, la confidentialité et la non-répudiation des messages. Les

fonctionnalités apportées par ces outils permettent ainsi de répondre aux interrogations concernant la sécurité des transactions sur Internet.

Les risques encourus lors d'échanges de documents sur Internet impliquent que ces échanges soient protégés des pertes, des altérations, des intrusions, des utilisations frauduleuses...

La configuration des outils d'interface (protocoles de communication et services de messagerie), ainsi que la définition des processus d'échanges entre partenaires permet de protéger efficacement les échanges contre tous les risques actuellement connus.

## Schema des fonctions de sécurisation

La présentation générale des différentes fonctions de sécurisation des échanges est résumée dans le schéma suivant :

Figure 4 - Les différents services de sécurité dans une interface de messagerie

## Le protocole SOAP (Simple Object Access Protocol) (1)

Ces différents niveaux de sécurisation sont gérés dans le protocole SOAP (Simple Object Access Protocol) qui assure aux échanges sur Internet un niveau de sécurité équivalent ou supérieur à celui qui est connu en EDI conventionnel.

Les services d'authentification permettent de s'assurer que l'émetteur et le destinataire sont effectivement les bons interlocuteurs.

L'autorisation contrôle l'habilitation des partenaires à réaliser un type de transaction.

La non répudiation à l'émission comme à la réception évite que l'un des partenaires affirme ne pas être partie prenante à la transaction.

Le chiffrement assure la confidentialité du message alors que la signature digitale confirme son engagement formel.

Figure 5. - Structure de protocole SOAP

## Le protocole SOAP (Simple Object Access Protocol) (2)

SOAP utilise les protocoles de communication d'Internet et la fonction de messagerie MIME.

SOAP est structuré en trois parties :

1. le package du message qui constitue l'enveloppe de transport physique ;
2. l'en-tête du conteneur qui comprend des données pour le routage du message, sa traçabilité, l'information générale des partenaires... et une description de ce qui est transporté dans le conteneur de documents ;
3. Le conteneur qui est l'espace dans lequel seront mis les différents documents (ou messages) qui constituent l'envoi.

Au-delà des protocoles de communication et des couches sécuritaires qui peuvent y être associées, Internet met à la disposition des utilisateurs de nombreuses fonctions qui pourront être utilisées dans le cadre des échanges électroniques :

- le navigateur qui permet de visualiser des documents directement à l'écran ;
- le parseur qui est utilisé pour contrôler la structure du document par rapport à un modèle ;
- les langages de manipulation de données qui sont utilisés pour transformer un message d'une structure dans une autre structure qui peut être le format d'entrée dans une application ;
- les fonctions d'interrogation de services distants qui permettent de contrôler, d'enrichir, de valider... des données par rapport à des informations et des services qui ne sont pas disponibles dans l'entreprise comme : le contrôle d'existence, l'appel à un service de signature digitale, le paiement par carte bancaire, la recherche de données sur la disponibilité d'un produit...

## 1.5. XML pour permettre l'accès aux échanges électroniques au plus grand nombre d'entreprises

### Les particularités de EDI

L'EDI conventionnel est réservé aux entreprises d'une certaine taille dont les échanges, en termes de volume, sont réguliers et significatifs. C'est pour cette raison que l'EDI conventionnel est devenu un outil incontournable pour les activités situées " au cœur du métier " des grandes entreprises. Sans l'EDI, des secteurs entiers de l'économie (fabrication et distribution des produits de grande consommation, industrie automobile, industrie chimique, électronique, ...) ne pourraient pas fonctionner.

L'EDI conventionnel a permis des gains de productivité très importants et la réingénierie des processus d'affaires dans de nouveaux concepts de management comme le juste à temps, l'ECR, le Cross Docking, la traçabilité, l'amélioration de la relation avec les clients.

### Limites de EDI

Cependant, l'EDI conventionnel montre ses limites :

- lorsqu'il faut échanger de petits volumes avec un grand nombre de partenaires. Il est donc difficile de dématérialiser les flux avec les fournisseurs de services, de consommables ou de fournitures de bureau ;

- lorsqu'il est nécessaire de concevoir de nouvelles applications utilisant le multimédia ou qui nécessite une grande flexibilité d'adaptation comme la gestion de catalogues produits multisectoriels ;
- pour des applications nécessitant des échanges avec des délais de réponse court (applications orientées temps réel).

## XML et les échanges occasionnels entre entreprises

L'utilisation d'XML permet de répondre à ces besoins d'échanges occasionnels entre entreprises. La légèreté des infrastructures à utiliser permet d'envisager l'extension des échanges électroniques professionnels au plus grand nombre et la conception de nouvelles applications utilisant des données non textuelles, des structures variables et / ou des " applications temps réel "

L'utilisation d'XML pour les échanges de données entre partenaires permet de dépasser ces limites pour les raisons suivantes :

- l'infrastructure Internet est peu coûteuse et équipe la grande majorité des micro-ordinateurs des PME et TPE. À partir d'XML, le récepteur du message peut choisir entre une intégration des données dans son système d'information ou leur visualisation à l'aide d'une feuille de style dans un navigateur ;
- le métalangage XML permet de définir des métamodèles qui eux-mêmes permettent de définir d'autres modèles jusqu'à la définition de la structure des messages utilisés par les entreprises. Cette technique autorise la définition de structures génériques qui servent de cadre aux développements d'applications présentant une grande flexibilité de mise en œuvre ;
- l'infrastructure Internet pour l'interrogation des sites Web a démontré sa capacité à mettre en œuvre des applications interactives. Cette capacité est utilisable pour des échanges où le temps de réponse devient un élément déterminant. À partir de cette possibilité, des échanges peuvent être envisagés avec un retour d'information presque immédiat. Ainsi, par exemple, l'interrogation sur la disponibilité d'un article, la réservation d'un service, le contrôle de solvabilité...

## Comparaison entre XML et l'EDI Conventionnel

Figure 6 - Comparaison entre XML et l'EDI Conventionnel

## Comparaison entre XML et l'EDI Conventionnel (2)

Ce schéma montre le parallélisme existant entre les traitements des transferts de messages en EDI conventionnel et en XML sur Internet.

Les opérations à mettre en œuvre pour échanger des données entre partenaires sont les suivantes :

- extraction des données à transmettre ;
- mise en forme des données en respectant les règles de syntaxe et de vocabulaire du langage utilisé ;
- transmission vers un réseau à valeur ajoutée ou Internet ;
- circulation sur le réseau vers la boîte aux lettres du destinataire ;
- récupération des messages par téléappel de la boîte aux lettres ;
- contrôle des messages à partir des règles de syntaxe et de vocabulaire du langage utilisé ;
- transformation des messages en un format utilisable par l'application de gestion réceptrice ;
- exploitation des messages dans l'application de gestion réceptrice.

La branche XML permet, en outre, la présentation du message dans un format lisible par un homme et la possibilité d'obtenir une réponse en temps réel si le système d'échange a été prévu pour cela.

## 2. Les composantes techniques pour

# l'utilisation d'XML dans le cadres des échanges électroniques professionnels

## Présentation

Ce chapitre a pour objectif de fournir au lecteur des informations suffisantes pour qu'il soit en mesure de comprendre l'intérêt du métalangage XML dans le cadre des échanges électroniques entre entreprises et entre systèmes d'information.

Si les spécifications d'XML sont désormais stabilisées, ce langage fait encore l'objet de nombreux compléments qui viendront en enrichir les possibilités. Ce chapitre fait donc un point sur l'état de l'art au premier trimestre de l'année 2002 et présente les évolutions prévisibles en fonction des travaux actuellement en cours. Les auteurs n'ont pas souhaité spéculer sur des évolutions qui n'auraient pas connu un début de formalisation.

La présentation des éléments du langage XML dans la première partie est complétée par une présentation des méthodes d'analyse qui sont actuellement recommandées pour la spécification des échanges électroniques.

Comparés à ceux de l'EDI conventionnel, les projets utilisant XML intègrent de nouveaux concepts. Ces concepts demandent une évolution dans les méthodes de modélisation. Ces dernières sont présentées dans la deuxième partie du chapitre.

La troisième partie décrit les étapes de mise en oeuvre d'un projet utilisant XML pour des échanges électroniques.

Enfin, la dernière partie donne des informations sur les outils disponibles pour la mise en oeuvre de ces nouvelles technologies.

## 2.1. Les éléments du métalangage XML

XML est un métalangage, c'est-à-dire un langage qui permet de définir d'autres langages. Il a été conçu dans le cadre du World Wide Web Consortium (W3C : <http://www.w3.org>).

Les spécifications de ce métalangage sont accessibles en français à l'adresse suivante : [http://babel.alis.com/web\\_ml/xml/REC-xml.fr.html](http://babel.alis.com/web_ml/xml/REC-xml.fr.html)

### 2.1.1. Les composants d'XML

#### Presentation

XML est un métalangage qui peut être utilisé pour de nombreuses autres applications que celles concernant le transfert de données entre systèmes d'information. Ce chapitre ne présente que la partie des instructions XML permettant de comprendre son utilisation dans les échanges de données.

#### Prologue XML

Cette ligne, la première d'un document XML, donne la version de syntaxe XML qui est utilisée ainsi que le mode d'encodage des caractères du message.

```
<?xml version="1.0" encoding="UTF-8"?>
```

#### Les éléments XML

Un document XML est organisé en structure hiérarchique dont certains nœuds peuvent définir les niveaux de la hiérarchie et d'autres contenir des données significatives.

```
<buyer>
<PartyIdentification>
<GlobalLocationNumber>4325335000003</GlobalLocationNumber>
</PartyIdentification>
<BuyerAlignmentIdentification>Buyer</BuyerAlignmentIdentification>
</buyer>
```

Dans cet exemple, le nœud est identifié par la balise de début <buyer> et la balise de fin </buyer>

Au plan suivant, on trouve la balise <PartyIdentification> qui indique que ce qui va suivre est l'identification du partenaire. Au niveau inférieur, la balise <GlobalLocationNumber> indique que 4325335000003 est un code d'identification.

Cet exemple montre les balises XML peuvent jouer le rôle de qualifiant des données transportées.

Cet exemple permet d'appréhender la facilité apportée par XML pour exprimer toutes les structures de messages dont les partenaires ont besoin.

## Les attributs XML

Les attributs XML permettent de qualifier les balises pour en préciser la signification.

```
<Member IdCodeType="GLN" MemberId="888888888">
<PartyName>AjubaSolutions</PartyName>
<CompanyTelephone>650-210-0100</CompanyTelephone>
```

</Member>

Dans cet exemple, le qualifiant `IdCodeType` qui prend la valeur "GLN" indique que le code qui suit `MemberId` est un code GLN.

L'utilisation de qualifiant permet de définir des structures plus génériques. `MemberId` est une structure générique qui doit être qualifiée pour déterminer la nature du code manipulé.

## Elément ou attribut ?

L'exemple précédent pourrait également être exprimé de la manière suivante :

<Member>

<IdCodeType>GLN</IdCodeType>

<MemberId>888888888</MemberId>

<PartyName>AjubaSolutions</PartyName>

<CompanyTelephone>650-210-0100</CompanyTelephone>

</Member>

Le choix entre une balise qualifiante ou une balise générique à laquelle on ajoute un attribut pour la qualifier doit être dicté par la volonté de définir des structures qui seront facilement réutilisables dans d'autres parties du message ou dans d'autres messages.

Si cette structure doit être réutilisée dans de nombreux contextes différents, il est préférable d'utiliser un attribut pour la qualifier. Si, au contraire, l'utilisation est toujours identique, il est préférable de donner un sens à la balise elle-même.

## Des balises définies par les utilisateurs

XML encadre les données entre des balises. Par exemple :

```
<date>2000-05-11</date>
```

Chaque donnée est placée entre une balise de début (<date>) et une balise de fin (</date>). Le nom placé dans ces deux balises permet de définir la sémantique de la donnée.

L'ensemble " données et balises " forment un élément. Un élément peut contenir des données, des éléments ou une combinaison des deux (mixed element).

Certains éléments peuvent être vides (ne pas contenir de donnée, ni d'élément), dans ce cas, une balise d'élément vide peut être utilisée :

```
<saut-de-page/>
```

XML est un métalangage permettant de définir d'autres langages.

Chaque langage définissant ses balises autorisées et les règles qu'elles doivent suivre.

Un grand nombre de langages ont été, sont et vont être définis à partir d'XML dans les domaines les plus variés.

Les données peuvent être imbriquées, formant une hiérarchie stricte jusqu'à une racine commune qui définit le fichier ou message XML, appelé document :

```
<client>
<nom>Issy Magasins</nom>
<adresse>
<rue>120 place de Budapest</rue>
<ville>Paris</ville>
<code-postal>75009</code-postal>
<pays>France</pays>
</adresse>
</client>
```

Les balises de début et les balises d'éléments vides peuvent contenir un ou plusieurs attributs :

```
<temperature echelle='Celsius'>23</temperature>

<date-livraison-au-plus-tard          format='ISO          8601'>
2000-05-20T14:00+2:00</date-livraison-au-plus-tard>
```

Des balises d'éléments vides peuvent être utilisées pour stocker des données lors d'une instanciation (au sens langage Objet) du message.

```
<temperature echelle='Celsius' valeur='23'/>
```

Dans cet exemple la balise echelle et la balise valeur peuvent être vides. Lors de l'utilisation, ces balises sont utilisées pour indiquer l'échelle de température et la valeur de la température par rapport à cette échelle.

## Les espaces de nommage

Comme nous l'avons vu au paragraphe précédent, les utilisateurs d'XML ont la capacité de créer leurs propres balises pour définir les concepts qu'ils souhaitent manipuler. De plus, pour que deux systèmes puissent échanger, il est nécessaire qu'ils utilisent un même vocabulaire et une même grammaire.

Si plusieurs groupes d'utilisateurs créent leur propre vocabulaire et si un message doit utiliser des " mots " issus de plusieurs de ces groupes, des risques de confusion peuvent alors se présenter.

Les espaces de nommage permettent d'indiquer l'origine des mots utilisés et donc d'éviter toute confusion concernant leur sens.

L'espace de nommage permet de dire : " Ce qui suit, appartient à tel référentiel... ".

La syntaxe de l'attribut définissant un espace de nommage est la suivante :

```
xmlns:tp="http://www.ebxml.org/namespaces/tradePartner"
```

Les balises préfixées par tp (par exemple <tp:PartyInfo>) dans la structure imbriquée font référence à un espace de nommage qui s'appelle "http://www.ebxml.org/namespaces/tradePartner".

## Schéma et DTD

Les schémas et DTD (Document Type Definition) permettent de définir la structure des messages. Ils constituent une sorte de plan qui permettra de contrôler que le message possède une structure correspondant à celle qui est attendue.

Lors de la création du langage XML, le concept de DTD a été repris du langage SGML pour définir la structure des documents. Par la suite, les concepteurs du langage ont imaginé une nouvelle méthode permettant de définir la structure de document XML à partir d'autres documents XML que l'on nomme " schémas XML ".

Les DTD sont encore largement utilisées. C'est la raison pour laquelle les DTD, les schémas et une analyse comparative des deux techniques sont successivement présentés dans ce paragraphe.

Par la suite, il ne sera plus fait référence qu'à la syntaxe des schémas qui est la plus utilisée actuellement. Des outils de commerce permettent de transformer des DTD en schéma.

## Les DTD

DTD est l'acronyme de Document Type Definition (Définition de Type de Document)

Le rôle d'une DTD est de définir la structure d'un document XML. Une DTD est caractérisée par un ensemble de règles permettant de spécifier les éléments du document XML, ainsi que leur ordre et leur fréquence d'apparition.

Une DTD est décrite selon la norme EBNF (Extended Backus-Naur Form) ; à ce titre, elle n'est pas une application XML mais un héritage de SGML. De ce fait, elle n'est pas manipulable par les outils XML mais par des outils qui lui sont spécifiques.

La DTD est une caractéristique optionnelle d'un fichier XML. Si le fichier XML n'a pas de DTD, ce dernier devra alors être "bien formé" (well-formed) et respecter les règles imposées par XML. Si une DTD est rattachée au fichier XML, ce dernier est automatiquement considéré comme valide puisque la définition des balises du document y est incluse.

La DTD peut être incluse au fichier XML. On dit alors qu'elle est interne, ou

externe et dissociée du fichier XML, le nom du fichier contenant la DTD lui est alors associé.

Une DTD commence toujours par `<!DOCTYPE` et se termine par `>`. La racine du document est renseignée après `<!DOCTYPE` et doit être également définie comme un élément de la DTD.

La DTD est donc construite à partir d'un ensemble de déclarations permettant de définir le type, la nature et les contraintes liés à chaque nouvelle balise. Cet ensemble de déclarations comprend :

- la déclaration de types d'éléments, permettant de définir le contenu du fichier XML ;
- la déclaration de listes d'attributs, permettant d'enrichir la sémantique des éléments ;
- la déclaration d'entités ;
- la déclaration de notations.

Une DTD externe peut être modifiée ou complétée par une DTD interne.

Une DTD permet de définir les contraintes que doivent suivre les éléments (par exemple, un élément " adresse " est constitué d'un élément " rue ", d'un élément " complément d'adresse ", optionnel, d'un élément " code postal ", d'un élément " ville " et d'un élément " pays ", optionnel).

La DTD permet également de définir les attributs possibles avec éventuellement des valeurs fixées ou des valeurs par défaut.

## Les schémas XML

Comme une DTD, un schéma XML permet de définir un ensemble de règles

visant à définir un document XML, notamment les balises autorisées, leurs attributs et les relations des marqueurs entre eux.

Contrairement à une DTD, un schéma permet de définir des types pour les données. Par exemple, avec une DTD, il sera possible de définir une balise <NOM>, dont le contenu pourra être n'importe quel type de données.

La puissance des schémas réside dans la possibilité de " typer la balise ". Ainsi, on pourra obliger un utilisateur à ne saisir que des caractères pour la balise <NOM> si celui-ci est défini comme une chaîne de caractères. Il sera également possible, d'en définir la longueur minimale et maximale...

De plus, un schéma XML est un document XML à part entière. Il peut donc être édité et manipulé à partir de n'importe quel outil d'édition ou de traitement XML. Cette possibilité est intéressante pour les échanges entre systèmes hétérogènes. Elle permet de définir un schéma-pivot que l'on nomme " schéma d'interopérabilité " entre schémas différents concernant un même type de données. Le schéma d'interopérabilité assurera la transformation d'un message d'une structure vers l'autre structure. C'est par ce moyen que les différentes initiatives d'échanges électroniques peuvent envisager leur compatibilité.

## Comparaison entre les schémas et les DTD

Une DTD permet de définir la structure d'un document, mais ne permet pas de spécifier le type des données. Ses possibilités sont donc restreintes (énumérations limitées aux attributs, pas de possibilité de décrire un nombre de répétitions, ...). En outre, les DTD sont peu adaptées à la gestion des espaces de nommage.

La recommandation d'utiliser les schémas XML, approuvée par le W3C le 2 mai 2001, a pour objectif de permettre :

- de décrire la structure des documents dans un format XML ;
- de spécifier le typage des données ;
- de supporter les espaces de nommage ;

- de construire des types de données à partir d'autres types de données (héritage).

Les schémas XML permettent aussi d'ignorer des éléments qui n'ont pas été définis (notion de modèle ouvert). Cette fonction est particulièrement utile lors du changement de version d'un schéma, la compatibilité dans les différentes versions ascendantes étant alors assurée en ignorant les ajouts des nouvelles versions par rapport aux anciennes.

En résumé, on préférera les schémas XML aux DTD pour les raisons suivantes :

- une DTD est difficile à lire ;
- une DTD est non extensible, car elle n'est pas un document XML ;
- une DTD ne permet pas de " typer " les données ;
- une DTD ne supporte pas les espaces de nommage (Namespaces) ;
- une DTD est plus concise mais moins riche qu'un schéma XML.

## La création des schémas

Les schémas sont des documents XML commençant par un élément " schéma " :

```
<?xml version="1.0" encoding="utf-8"?>
```

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
```

## Les types de données dans les schémas

La puissance des schémas réside dans leur faculté à donner un type aux balises. Cette possibilité présente un avantage certain pour l'exploitation des

données dans les systèmes d'information. Les " développeurs " connaissent ainsi précisément la structure des données attendues. Par exemple :

```
<xsd:element name="LineItemNumber" type="xsd:nonNegativeInteger"/>
```

```
<xsd:element name="Price" type="xsd:decimal" minOccurs="0"/>
```

L'expression `type="xsd:nonNegativeInteger "` indique que la donnée est une chaîne de caractères positifs. Il aurait également été possible d'en préciser la longueur minimale et maximale.

Il est possible dans les schémas XML de définir des types complexes c'est-à-dire incluant plusieurs éléments. Par exemple :

```
</xsd:element>
```

```
<xsd:element name="LineItem" maxOccurs="unbounded">
```

```
<xsd:complexType>
```

```
<xsd:sequence>
```

```
<xsd:element name="LineItemNumber" type="xsd:nonNegativeInteger"/>
```

```
<xsd:element name="Price" type="xsd:decimal" minOccurs="0"/>
```

```
<xsd:element name="Quantity" type="xsd:integer"/>
```

```
</xsd:sequence>
```

```
</xsd:complexType>
```

```
</xsd:element>
```

L'élément " Lineltem ", de type complexe, comprend des éléments en séquence : LineltemNumber, Price (optionnel), et Quantity.

Il est également possible de définir des types simples comprenant uniquement des valeurs :

```
<xsd:simpleType name="AlternateltemIdListType">  
  
<xsd:restriction base="xsd:string">  
  
<xsd:enumeration value="COUPON_FAMILY_CODE"/>  
  
<xsd:enumeration value="SUPPLIER_NUMBER"/>  
  
</xsd:restriction>  
  
</xsd:simpleType>
```

L'élément " AlternateltemIdListType " ne comprend aucun élément ou attribut. Les valeurs qu'il peut prendre sont COUPON\_FAMILY\_CODE et SUPPLIER\_NUMBER.

## La cardinalité

La cardinalité permet de définir le nombre de répétitions possibles pour un élément.

Dans l'exemple qui suit, le nombre de répétitions pour définir les personnages de l'ouvrage n'est pas limité et peut aller de 0 à l'infini.

```
<xsd:element ref="character" minOccurs="0" maxOccurs="unbounded" />
```

Les informations sur la cardinalité sont précieuses aux " développeurs " pour définir la structure de la base de données nécessaire à l'exploitation des messages.

## Le choix

L'élément " choice" est à utiliser si un élément peut prendre une valeur parmi plusieurs, ou si, dans un groupe, il est possible de prendre un élément parmi plusieurs. Par exemple :

```
<xsd:group name="nameTypes">  
  
<xsd:choice>  
  
<xsd:element name="name" type="xsd:string"/>  
  
<xsd:sequence>  
  
<xsd:element name="firstName" type="xsd:string"/>  
  
<xsd:element name="middleName" type="xsd:string" minOccurs="0"/>  
  
<xsd:element name="lastName" type="xsd:string"/>  
  
</xsd:sequence>  
  
</xsd:choice>  
  
</xsd:group>
```

Cet exemple montre qu'il faut choisir soit un nom (name), soit la séquence

(firstName, middleName, lastName). Dans cette liste, middleName est conditionnel car son occurrence minimum peut être égale à 0.

## L'ordonnement des éléments

La commande " sequence " définit l'ordre des éléments dans un groupe et la commande " all " indique que l'ordre est libre.

```
<xsd:complexType name="bookType">  
  
<xsd:all>  
  
<xsd:element name="title" type="xsd:string"/>  
  
<xsd:element name="author" type="xsd:string"/>  
  
<xsd:element name="character" type="characterType"  
  
minOccurs="0" maxOccurs="unbounded"/>  
  
</xsd:all>  
  
</xsd:complexType>
```

Les éléments title, author, character doivent apparaître et dans n'importe quel ordre ; character est optionnel, les autres éléments sont obligatoires.

## 2.2. Les normes complémentaires à XML

### Presentation

Comme le montre la figure suivante, les spécifications liées à XML peuvent être classées, d'une part en spécifications d'outils complétant XML (recommandations complémentaires), d'autre part en spécifications de langages de description bâties en utilisant le métalangage XML (recommandations dérivées). Ces dernières sont manipulables directement avec les outils XML.

Figure 7 Autour d'XML

### 2.2.1 Les parseurs XML

Les parseurs sont des logiciels qui permettent de lire des documents XML. Cette lecture peut être faite avec un contrôle du document par rapport à son schéma ou à sa DTD. Dans ce cas, le parseur est dit " validant ".

Il existe des parseurs pour les différents langages (Java, Perl, C++, Visual Basic, JavaScript, Python, ...) et pour les différents environnements (ActiveX, Java Virtual Machine).

Trois types d'interfaces sont définis entre un parseur et l'application qui l'utilise :

- l'interface SAX (Simple Access to XML) qui enchaîne un ensemble de rétro appels vers l'application au fur et à mesure que les objets XML sont traités (balise de début, attribut, texte, balise de fin, ...)
- l'interface DOM (Document Object Model) qui construit une hiérarchie d'objets représentant le document en mémoire (document, éléments, attributs, contenus...)
- les interfaces utilisant des objets Java correspondant au document XML parsé.

L'interface SAX (<http://www.megginson.com/SAX/Java/sax2.zip>) a été établie par un groupe de programmeurs participant à la liste de diffusion XML-DEV (celle des développeurs XML).

Dans cette catégorie d'interface, on peut citer :

- SAX et XAF de Megginson ;
- Expat de James CLARK.

L'interface DOM a été recommandée par le W3C mais elle oblige à maintenir une hiérarchie d'objets représentant tout le document en mémoire, ce qui peut être contraignant pour les grands documents. Les parseurs DOM utilisent souvent un parseur SAX en interne. Deux versions de DOM sont opérationnelles. La troisième version est en cours de normalisation.

Dans cette catégorie d'interfaces, on peut citer :

- MSXML de Microsoft distribué avec Internet Explorer ;
- XERCES, utilisable en complément des serveurs Apache.

Les parseurs générant des objets Java correspondant au type de document parsé n'ont pas d'interface normalisée. Plutôt que de gérer une hiérarchie d'objets de type balise, attribut, instruction de traitement, ils génèrent, dans la phase d'analyse, un objet Java par type de document. Cet objet Java dispose d'attributs correspondant aux différents éléments imbriqués, conduisant à une programmation beaucoup plus naturelle.

Dans cette troisième catégorie d'interfaces, on peut citer :

- Breeze de Breeze Factor (<http://www.breezefactor.com/>) ;
- JDOM, supporté par l'éditeur O'Reilly (<http://www.jdom.org>).

Comparaison avec les traducteurs EDI

On compare souvent les parseurs et les traducteurs EDI. En fait, leurs dissemblances sont plus importantes que leurs ressemblances.

Un traducteur fonctionne dans deux sens :

- création d'un message à partir de fichiers applicatifs ;
- création de fichiers applicatifs à partir d'un message.

Les étapes de traitement dans un traducteur sont les suivantes :

Figure 8 Les fonctions d'un traducteur EDI

Le parseur ne réalise que la partie " contrôle de la structure ". Il est nécessaire de faire appel à d'autres outils pour gérer les transformations de structure. Ces outils utilisent le langage XSL. Sa mise en œuvre demande donc des compétences de programmation.

Un traducteur dispose généralement d'une interface graphique permettant son paramétrage, c'est-à-dire de définir les rapprochements entre les données applicatives et les données du message. Lors de ce rapprochement, le traducteur permet généralement de définir des règles de transformation : troncature, transcodage, mise en équivalence, création de valeur par défaut... Dans le cas d'XML, ces tâches nécessitent de la programmation.

Les industriels des logiciels d'interface, en toute probabilité, seront conduits à développer des outils XML de plus en plus complets et paramétrables et dont les fonctions seront progressivement identiques à celles des traducteurs.

### 2.2.2. Les feuilles de styles : XSL et XSLT

Comme nous venons de le voir, XML permet de structurer des données, mais il ne permet pas de les manipuler. Dans les extensions d'XML, un certain nombre de standards ont été créés pour en faciliter l'utilisation.

## XSL

Parce que des données d'un message ne sont pas affichables directement dans une présentation compréhensible par un non-informaticien, XML fait appel à un langage de présentation pour assurer la mise en forme des données qu'il formalise et transporte. Ce langage est XSL, abréviation de eXtensible Stylesheet Language.

XSL permet de manipuler les documents XML avec des feuilles de style afin de les rendre lisibles par un humain. Cette capacité de manipulation des données d'un message XML est aussi utilisée pour transformer un document XML en un autre document XML ou en une autre structure de fichiers comme un fichier applicatif dans le cas d'intégration de messages transmis d'un système à un autre système.

## XSLT

Un langage XML définit des données.

Afin de traiter un document XML, permettant ainsi de générer une présentation des données ou d'en effectuer un filtrage ou un enrichissement, il est possible de développer un programme utilisant un parseur pour accéder aux éléments et générer les données appropriées.

Une autre solution consiste à utiliser une feuille de style XSLT (eXtensible Stylesheet Language Transformations, <http://www.w3.org/TR/xslt>) qui est un langage déclaratif de transformation, décrit en XML.

La recommandation XSLT définit un ensemble de patterns (modèles) et de templates (gabarits) permettant de définir des règles de transformations.

XSLT permet :

- de modifier la structure d'un document XML. Cette fonctionnalité sera utilisée pour gérer les échanges entre entreprises appartenant à des secteurs différents et donc ayant défini des messages pouvant avoir des structures différentes ;
- de présenter le message dans une interface humaine via un navigateur Web, un document PDF de Adobe ou RTF de Microsoft...

### 2.2.3. L'accès à des sous-ensembles : XPath

La recommandation XPath (<http://www.w3.org/TR/xpath>) permet d'extraire un sous-ensemble d'un document XML, basé, par exemple, sur une valeur de contenu d'élément ou sur une valeur d'attribut.

XPath définit un mécanisme d'extraction destiné à être utilisé par d'autres composants comme le langage de transformation XSLT et le langage standard d'interrogation XML Xquery, en cours de définition.

Dans une application concernant les échanges de données, cette faculté pourra être utilisée pour aller rechercher des données complémentaires à un message dans une base de donnée XML. Par exemple, pour un message ne contenant qu'un code article, XPath permettra d'aller rechercher dans une base de données de produits, le nom, la description et les caractéristiques techniques de ce produit.

### 2.2.4. Les liens hypertextes : XLink et XPointer

Les liens possibles avec HTML restent assez limités et imposent une modification du document cible (définition d'une ancre) quand on veut pointer à l'intérieur de ce document. Ils ne permettent pas, d'autre part, de pointer vers plusieurs documents.

XML offre des possibilités de pointage supplémentaires avec XLink et Xpointer.

XLink (<http://www.w3.org/TR/xlink>) permet :

- de construire un mécanisme de liens identique à celui de HTML (pointage entre deux documents ou à l'intérieur d'un document sur une ancre) ;
- de construire des liens bidirectionnels (A pointe vers B et B pointe vers A) ;
- d'utiliser n'importe quel élément comme lien, ce qui évite d'avoir à définir des ancres ;
- de stocker les liens en dehors des documents en relation, ce qui permet de gérer les liens sans avoir à modifier les documents ;
- de pointer vers plusieurs documents.

Basée sur Xpath, la spécification XPointer (<http://www.w3.org/TR/xptr>), utilisée dans une expression de lien XLink permet de pointer sur un emplacement dans un document défini par sa position (par exemple, le troisième élément LIGNE\_FACTURE du quatrième élément FACTURE).

## 2.2.5. Le langage d'interrogation XQuery

Le développement de XQuery répond à deux impératifs :

- la nécessité de pouvoir retrouver des documents XML stockés dans une base de données ;
- la nécessité de pouvoir extraire des sous-ensembles ou créer de nouveaux documents XML à partir des documents stockés.

Répondant à ces besoins, le langage d'interrogation XQuery est en cours de définition et un certain nombre de documents de travail sont disponibles à l'adresse <http://www.w3.org/XML/Query> :

- expression des besoins : XML Query Requirements (<http://www.w3.org/TR/xmlquery-req>) ;

- définition du langage : XQuery: A Query Language for XML (<http://www.w3.org/TR/xquery>) ;
- algèbre : The XML Query Algebra, (<http://www.w3.org/TR/query-algebra>) ;
- cas d'utilisation : XML Query Use Case, (<http://www.w3.org/TR/xmlquery-use-cases>) ;
- modèle de données : XML Query Data Model, (<http://www.w3.org/TR/query-datamodel>).

Ce langage sera particulièrement utile pour l'exploitation des base de données XML, c'est-à-dire des base de donnée qui stockent directement des document XML sous la forme d'arbres hiérarchiques.

## 3. L'utilisation d'XML pour les échanges électroniques professionnels

### Presentation

Dans cette partie, nous faisons régulièrement référence à ebXML dans la mesure où ebXML constitue un cadre global d'utilisation d'XML pour les échanges électroniques professionnels. Ce cadre sert de base de référence à la plupart des initiatives utilisant XML pour les échanges.

Celles qui existent déjà indiquent leur intention de l'adopter en totalité ou en partie pour continuer leur développement.

Les échanges électroniques professionnels (EEP) sont des applications informatiques particulières pour les raisons suivantes :

- les échanges sont externes à l'entreprise ;
- les partenaires peuvent ne pas se connaître ;
- les implications économiques et juridiques de ces échanges peuvent être très

importantes ;

- les échanges concernent généralement des fonctions essentielles à la vie de l'entreprise : commerciales, financières, juridiques et fiscales,... ;
- les systèmes d'information sont hétérogènes, il est donc nécessaire d'utiliser un langage intermédiaire permettant aux systèmes de communiquer ;
- les moyens techniques mis en œuvre sont hétérogènes.

## 3.1. La gestion de la complexité

### Presentation

Un des objectifs poursuivi par les initiatives utilisant XML pour les échanges entre systèmes d'information consiste à étendre ces techniques au plus grand nombre. L'EDI conventionnel a montré ses limites dans ce domaine du fait de sa complexité de mise en place et des coûts qui sont liés à cette complexité.

### Mise en œuvre des EEP en EDI

Pour bien comprendre les niveaux de simplification à apporter à la mise en œuvre des EEP nous allons détailler les étapes de mise en place d'un système en EDI Conventionnel et voir ensuite comment les initiatives XML souhaitent automatiser ces étapes pour les simplifier.

Avec cette description, il est possible de se rendre compte que la complexité est prise en charge par des outils techniques paramétrables manuellement. Chaque nouvelle implémentation a un coût, chaque nouveau partenaire à un coût. Ces coûts représentent essentiellement des coûts humains sur lesquels les gains de productivité réalisables seront fatalement limités. C'est la raison essentielle pour laquelle l'EDI conventionnel a montré ses limites à l'égard des PME et des TPE.

### Recherche d'automatisation des échanges dans ebXML et EDIFACT

Comparées à EDIFACT, les initiatives à base d'XML et plus particulièrement d'ebXML vont beaucoup plus loin dans la recherche d'automatisation des échanges.

Dans ce tableau, les cases en grisé indiquent les fonctions qui ne sont pas ou peu couvertes par le standard. Il est possible de s'apercevoir que les objectifs d'ebXML sont beaucoup plus ambitieux que ceux d'EDIFACT. Cette ambition traduit la volonté de définir un standard permettant une automatisation la plus complète possible du paramétrage des interfaces avec les systèmes d'information des partenaires. Une telle automatisation, si elle est mise en œuvre, permet un abaissement très significatif des coûts de démarrage. La complexité ne se situe plus au moment de la mise en œuvre mais lors de la spécification et de la modélisation des processus d'affaires.

## Mise en œuvre des EEP en ebXML

Si l'on reprend la grille d'analyse de la mise en œuvre des différentes étapes d'un projet EEP et que nous l'appliquons à ebXML, nous obtenons le tableau suivant :

## UMM (UN/CEFACT Modelling Methodology) et la notation UML (Unified Modeling Language)

Avec ebXML, la complexité se situe dans la conception et pas dans la mise en œuvre. C'est pour cette raison qu'il a été nécessaire de réaliser une méthodologie d'analyse adaptée à ce nouveau challenge. C'est ainsi que l'UN/CEFACT a créé un groupe de travail destiné à élaborer une méthodologie d'analyse permettant de prendre en compte plus précisément ces différents aspects. Cette méthodologie, connue sous le nom de UMM (UN/CEFACT Modelling Methodology), utilise la notation UML (Unified Modeling Language). C'est une démarche dite " top down ", c'est-à-dire qui va du général vers le particulier. L'objectif est de définir un enchaînement de processus d'affaires et de documents devant être échangés entre les partenaires, ainsi que la description des capacités d'échanges des partenaires afin de pouvoir échanger automatiquement un agrément de collaboration, ainsi que les paramètres de configuration des outils d'interface ebXML.

## Composants nécessaires à la découverte des processus d'affaires

Les principales étapes de la démarche sont décrites dans les schémas suivants :

Les différents composants nécessaires à la découverte des processus d'affaires

L'activité d'analyse est centrée sur la découverte des processus et des données d'affaires. Pour leur découverte, trois axes sont utilisés : - les données relatives aux besoins de l'entreprise et de son environnement (exigences, partenaires, standards existants, modèles existants, experts) ; - les méthodes (UMM, UML et les autres techniques d'analyse) ; - les moyens (analystes, outils supportant les méthodes d'analyse, réviseurs).

## La démarche " top down " d'analyse des échanges entre partenaires

Le processus d'analyse est un processus descendant, du plus général vers le plus particulier. Le but est de décomposer la réalité des échanges en éléments de plus en plus fins pour arriver à la définition des transactions élémentaires qui vont composer un processus d'affaires. A ces transactions sont rattachés les documents-échangés L'analyse est réalisée en utilisant la notation UML qui permet, à partir des diagrammes de classes, d'activités et de collaborations, de définir les processus et les données d'affaires. Ces diagrammes sont transformés en schémas ebXML qui sont des schémas XML s'inscrivant dans le métamodèle ebXML.

La démarche " top down " d'analyse des échanges entre partenaires

## Processus itératif d'analyse

L'analyse des processus d'affaires et des données d'affaires est itérative ; elle sera remise en cause jusqu'à ce que la compréhension des processus et des données soit totale. L'analyste poursuit son investigation entre la spécification des processus d'affaires et celle des documents jusqu'à ce que cette analyse reflète parfaitement les besoins des utilisateurs.

Processus itératif d'analyse

## Les trois axes de l'analyse des échanges

L'analyse est conduite selon trois axes principaux qui concernent les processus, les collaborations et les éléments économiques qui jouent le rôle de déclencheurs des processus et les transactions qui supportent les documents d'affaires :

Les trois axes de l'analyse des échanges

1. l'analyse centrée sur les domaines et les processus qui permet de connaître

la chorégraphie des échanges, c'est-à-dire leurs enchaînements possibles du début à la fin d'un processus ; 2. l'analyse centrée sur les événements économiques et les collaborations d'affaires. Cette analyse met en relief la formation des contrats et des engagements juridiques des différents partenaires, définit leurs rôles dans les échanges et permet l'élaboration des " profils de collaboration " entre partenaires ; 3. l'analyse centrée sur les transactions qui permet de connaître le contenu précis des messages participants aux échanges.

## La spécification des processus et des données d'affaires dans ebXML

Le processus d'analyse s'achève par la transformation du modèle d'échange décrit en UML par sa génération en schéma ou DTD XML. Les descriptions des processus, des collaborations et des documents doivent être exprimées en schémas XML contrôlés par rapport au métamodèle ebXML.

### La spécification des processus et des données d'affaires dans ebXML

Les processus et les données d'affaires ebXML appartiennent à un sous-ensemble du métamodèle UMM, ce sous-ensemble permet de définir des schémas UML des processus d'affaires et des documents d'affaires. Ces schémas UML sont transformés en schémas ebXML. Les signaux d'affaires sont des déclencheurs qui peuvent être internes ou externes au système d'information et qui permettent la mise en œuvre des transactions à l'intérieur des processus. Ces mécanismes permettent de prendre en compte la complexité des échanges entre partenaires, en particulier, en permettant de modéliser les cas d'exception. Les processus d'affaires sont mis en œuvre dans la collaboration entre partenaires. Ils mettent en œuvre des documents d'affaires qui eux-mêmes sont composés de données élémentaires normalisées et appartenant aux répertoires ebXML.

## 3.2. Les répertoires de données d'un langage XML

### Presentation

La plupart des initiatives d'échanges électroniques professionnels utilisant XML gèrent des répertoires dans lesquels les utilisateurs vont chercher les données

élémentaires, les données composites, les messages et les processus d'affaires qui correspondent à leurs cas d'utilisation. Cette approche évite de re-développer l'existant et permet à l'ensemble des partenaires de faire des économies de temps et d'argent. Dans ebXML, l'objectif étant d'automatiser la découverte, l'initialisation et de démarrage d'un partenaire, on stocke dans les répertoires, les données qui permettent de retrouver le partenaire ainsi que les caractéristiques techniques de sa capacité à mettre en œuvre des échanges dématérialisés.

## Collaboration Protocol Profil

Toutes les informations caractérisant les capacités d'échanges d'une entreprise sont stockées dans un CPP (Collaboration Protocol Profile). Ce CPP contient les caractéristiques techniques utilisables lors des échanges ainsi que les processus d'affaires qui ont été paramétrés dans les outils d'interface du partenaire.

### Création d'un Collaboration Protocol Profil par un partenaire

Lorsqu'une entreprise souhaite travailler avec une autre entreprise, elle recherche son CPP dans les répertoires et le compare au sein de manière à définir les caractéristiques techniques et les processus d'affaires communs aux deux. De cette superposition découle un agrément entre les deux partenaires sur la manière dont ils vont pouvoir travailler ensemble, c'est le CPA (Collaboration Protocol Agreement).

## CPA (Collaboration Protocol Agreement)

Le schéma suivant montre le mécanisme de création et d'acceptation du CPA.

Comparaison de deux CPP pour la création d'un CPA (Collaboration Protocol Agreement)

Les répertoires contiennent les données qui vont permettre aux partenaires de stocker les données qui les concernent de manière à les rendre partageables.

## Les données d'un répertoire ebXML

Le schéma suivant montre que les données stockées dans les répertoires ebXML sont contrôlées par rapport au métamodèle ebXML et sont utilisées par les CPP pour indiquer aux partenaires leur manière de travailler. Les données sont stockées en conformité avec le métamodèle ebXML. Le CCP d'un partenaire fait référence à la librairie des processus d'affaires, des données élémentaires, des scénarios, des contraintes de messagerie et de sécurité qu'il souhaite mettre en œuvre.

Les données d'un répertoire ebXML

À partir de la consultation d'un CPP, il est possible de connaître précisément la manière de travailler d'une partenaire

## La phase de conception et la phase d'exécution

La démarche ebXML est donc conçue en deux parties, une partie conception, déjà présentée dans la partie méthodologique et une partie exploitation qui permet en utilisant un CPA, de paramétrer les outils d'interface. Ce découpage est illustré par la figure qui suit.

La phase de conception et la phase d'exécution

Pour la plupart des entreprises, seule la phase d'exécution sera visible. La phase de conception est plutôt réservée aux grandes entreprises et aux groupes sectoriels de standardisation.

# 4. La mise en oeuvre d'un projet XML

## Presentation

L'analyse présentée dans le chapitre précédent ne concerne que la modélisation des échanges entre partenaires. Pour chaque partenaire, la mise en oeuvre des techniques d'XML peut s'inscrire dans un contexte plus large de refonte ou de modification du système d'information existant.

Notre propos est donc de rappeler les étapes essentielles de conduite d'un projet informatique en les modifiant et en les adaptant au contexte particulier d'un projet qui porte sur les échanges de données entre partenaires avec l'utilisation d'XML comme métalangage de communication.

### 4.1. Les engagements d'un projet XML

Indépendamment du secteur d'activités de l'entreprise, la gestion d'un projet doit devenir un moyen de mobilisation de l'ensemble des acteurs et des partenaires. La réussite de cette mission qui repose sur une définition précise de l'objectif et une appréciation pointue des enjeux, est conditionnée par le respect des quatre engagements présentés dans la figure ci-dessous :

Les quatre engagements concernent : La sécurité dans les échanges et la sécurité des systèmes d'information des partenaires. La sécurité des systèmes

d'information des partenaires peut être assurée par des outils spécifiques comme les firewalls et par des procédés intégrés aux échanges tels l'authentification des partenaires, le chiffrement et l'utilisation des signatures digitales. La qualité est une nécessité dans les échanges entre partenaires pour l'image de marque de l'entreprise et pour la sauvegarde de bons rapports commerciaux. Pour atteindre un niveau de qualité suffisant l'entreprise introduira dans les processus d'affaires un temps maximum d'exécution, la gestion d'accusé de réception technique pour la remise et pour l'acceptation sémantique et syntaxique du message. De plus, des procédures de communication alternatives seront mises en place pour contourner des moyens ordinaires devenus défectueux. Les délais sont de deux sortes. Les délais de mise en place du nouveau système d'échanges tout d'abord. Ils doivent être compatibles avec les exigences des partenaires. Les délais dans l'exécution des transactions, ensuite. Ils constituent une caractéristique propre de la transaction. Leur non-respect entraînera une perte de confiance de la part des partenaires. La réflexion sur les coûts doit également être portée à deux niveaux : " les coûts de réalisation du projet. Ils doivent être compatibles avec les capacités financières de l'entreprise et avec ses espérances de retour sur investissements ; " les coûts de fonctionnement des échanges en particulier pour la partie logistique qui aura tendance à croître du fait de la dématérialisation des échanges. Un des gains de productivité pour les partenaires est lié à une gestion en juste à temps des réapprovisionnements. Dans une telle démarche, le nombre de commandes augmente alors que leur valeur relative diminue, cela se traduit fatalement par une augmentation des coûts logistiques supportés par les entreprises. Le schéma montre que ces différents engagements sont interdépendants, si l'entreprise souhaite une sécurité renforcée et une grande qualité, cela a une incidence immédiate sur les coûts et les délais...

## 4.2. La démarche

Le schéma suivant présente un organigramme de la démarche qu'il est logique de suivre pour mener à bien un projet de cette nature. Chaque étape de l'organigramme est reprise dans la liste de contrôle qui suit et qui donne pour chacune des étapes, la liste des questions auxquelles l'entreprise doit répondre, ainsi que les moyens techniques à mettre en œuvre pour mener l'étape à son terme.

Une démarche de conduite de projet dans la mise en œuvre d'échanges à partir d'XML

## 4.3. Liste de contrôle

Cette liste de contrôle reprend les différentes étapes du schéma précédent. Elle constitue une trame qui pourra être complétée en fonction des besoins spécifiques de l'entreprise et ses partenaires.

#### 4.4. L'intégration dans les applications

Les gains de productivité administratifs sont obtenus avec l'intégration directe des messages dans les applications. Cette intégration doit prendre en compte différents composants qui sont exposés dans cette partie.

Pour une entreprise, il peut y avoir plusieurs stratégies d'intégration de la non-intégration à l'intégration de système. Il est évident que plus l'intégration est poussée plus les gains de productivité administrative peuvent être importants.

Le niveau d'intégration représente un coût qu'il est nécessaire de comparer aux gains induits. Pour les exceptions, une certaine part de traitements manuels constitue un compromis acceptable.

Nous présentons dans la suite de ce chapitre trois niveaux d'intégration qui sont fatalement caricaturaux.

##### 4.4.1. Notion de profils de partenaires

Dans l'EDI conventionnel, la capacité de dialogue des partenaires est définie par un contrat d'interchange. Dans la plupart des initiatives d'échanges électroniques professionnelles basés sur l'utilisation d'XML, ce " contrat " peut être constitué automatiquement par échange d'information sur les capacités des différents partenaires.

Dans l'exemple que nous avons décrit à partir d'ebXML, c'est le CPA qui joue le rôle de contrat d'interchange et qui permettra de piloter les interfaces avec les applications de gestion.

Le profil d'un partenaire définit ses capacités techniques à échanger : protocole de communication, niveau de sécurité mis en œuvre, etc., et ses capacités fonctionnelles comme les processus d'affaires implémentés et par leur intermédiaire, les documents que le partenaire est capable d'échanger.

#### 4.4.2. Liaison applications de gestion / messages

La liaison entre les applications et les messages répond à deux problématiques différentes :

- en émission, il faut développer des applications qui vont extraire les données du système pour les mettre en forme afin de constituer un message XML ;
- en réception, l'utilisation d'un parseur et de feuilles de style XSLT doit permettre la génération de fichiers applicatifs.

Cependant, la capacité de créer et d'exploiter des messages XML ne résout pas tous les problèmes de communication avec les systèmes d'information. Un certain nombre de difficultés peuvent surgir :

- structure des données différentes ;
- codifications différentes ;
- données absentes ;
- données en trop...

Pour chaque cas, il sera nécessaire de trouver une stratégie d'adaptation qui pourra être soit interne au système, soit gérée dans l'interface.

#### 4.4.3. Les stratégies d'intégration dans les applications

Différentes stratégies d'intégration peuvent s'appliquer selon le volume des

messages à échanger et la capacité d'adaptation des systèmes d'information. On peut définir les trois grandes stratégies suivantes :

#### L'intégration " fax "

Cette première approche est une stratégie défensive : il faut agir comme si l'entreprise était capable de communiquer en échanges électroniques professionnels. De fait, cette solution est souvent une solution d'attente avant de pousser plus loin l'intégration des fichiers. La forme " humaine " peut être évidemment l'affichage du message dans un navigateur ou son impression à partir d'une feuille de style.

#### L'intégration de fichier

Cette solution, vraisemblablement la plus commune, permet d'éviter la ressaisie des informations tout en conservant un contrôle humain pour leur intégration. Elle présente les avantages de ne pas remettre profondément en cause le système d'information existant, ni de modifier fondamentalement les habitudes de travail. Elle permet également de conserver une maîtrise sur les flux de données. Elle limite cependant les gains de productivité que l'entreprise est en droit d'attendre d'un système d'échanges électroniques professionnels.

#### L'intégration de système

Cette dernière solution est celle qui apporte les gains de productivité les plus importants. Les messages sont directement incorporés au système d'information qui effectue tous les contrôles courants. Si le système rencontre une anomalie, il la signale à l'opérateur pour un traitement manuel. Cette démarche n'est possible que si l'on a, au préalable, étudié toutes les modifications réalisées par les personnes travaillant sur les documents tout au long de la chaîne de traitement et si l'on a pu modéliser les modifications les plus répétitives. Dans ce cas, les gains de productivité peuvent être très significatifs. Cette solution est viable lorsque les volumes de messages échangés deviennent très importants.

## 5. Les outils techniques d'XML

### Presentation

Les différents composants liés au traitement et au stockage de données XML peuvent être classés en cinq catégories : 1. les éditeurs XML ; 2. les éditeurs de

DTD (ou éditeurs de schémas) ; 3. les bases de données XML ; 4. les processeurs XML (ou "parseurs") ; 5. les processeurs XSLT. Les composants d'un logiciel XML :

D'autres composants logiciels vont pouvoir importer et exporter des documents XML ou intégrer des données XML... Ces outils sont classés sous la rubrique " composants divers ". Le monde XML évoluant rapidement, les éditeurs et les solutions qui sont cités dans cette partie peuvent changer. Avant d'effectuer un choix, il sera souhaitable de réactualiser ces informations pour prendre en compte les évolutions du marché. Les conseillers techniques de Gencod EAN France et / ou ceux d'Edifrance peuvent être consultés à cet effet.

### 5.1. Les éditeurs XML

Les éditeurs XML permettent de saisir des documents XML. Certains sont issus du monde SGML, d'autres ont été conçus spécialement pour XML. Ils peuvent incorporer un éditeur de DTD. Certains éditeurs sont des éditeurs " lourds " destinés à la publication de documents importants : - BladeRunner de e-content company ; - Epic d'Arbortext ; - XMetal de Corel/Softquad ; - DynaBase d'INSO ; - Adobe. D'autres sont des éditeurs permettant de définir de petits documents, tels que : - XMLNotePad de Microsoft ; - XED de HCRC Langage Technology Group ; - XML Spy d'Altova ; - XMLPro de Vervet Logic.

### 5.2. Les éditeurs de DTD

Ces éditeurs permettent de définir une DTD ou un schéma, soit de manière graphique, soit à partir d'un ou de plusieurs documents XML existants. Cette fonction peut être assurée par un éditeur XML mais les éditeurs spécialisés fournissent souvent des fonctions plus avancées.

Dans cette catégorie, il est possible de citer :

- Visual DTD d'IBM ;

- XMLAuthority de Tibco qui permet notamment de convertir entre les différents formats : DTD (normalisé), XML-Data (spécifique), schémas...

### 5.3. Les bases de données XML

Ces serveurs permettent de stocker et de retrouver des données au format XML.

On peut distinguer plusieurs types de stockage :

- le stockage dans une base de données relationnelle :

\* Oracle 9i d'Oracle ;

\* Tamino de Software AG ;

- le stockage dans une base de données objets :

\* eXcelon d'Object Design ;

- le stockage natif de documents XML :

\* Tamino de Software AG.

### 5.4. Les parseurs XML

Les parseurs associés à un programme applicatif permettent de récupérer les données, le texte des éléments et les valeurs des attributs.

On distingue deux modèles de processeurs XML :

- Les processeurs DOM (Document Access Model)

Le modèle DOM est une recommandation du W3C. Les interfaces sont réalisées en IDL CORBA défini par l'OMG. Un processeur DOM présente au programme applicatif ou au navigateur, une arborescence d'objets correspondant au document traité.

### Processeurs DOM :

- \* Paquetage Perl DOM ;
- \* Docuverse du W3C ;
- \* composant ActiveX MSXML XMLDOM de Microsoft intégré dans Internet Explorer 5 et utilisable aussi bien dans un navigateur (IE4 ou IE5) que dans un serveur Web ;
- \* XP de James Clark ;
- \* XML4J d'IBM ;
- \* Project X Java de Sun ;
- \* XML Parser for Java d'Oracle ;
- \* Saxon de Michael Kay ;
- \* Xerces d'Apache ;
- \* Aelfred de Microstar.

- Les processeurs SAX (Simple Access to XML) appellent des fonctions enregistrées par le programme applicatif, à chaque fois qu'un élément ou qu'un attribut particulier est rencontré.

### Processeurs SAX :

- \* Paquetage Perl Expat de James Clark ;
- \* Xerces d'Apache.

Les processeurs DOM (" DOM parsers ") sont le plus souvent rencontrés, une fois que les objets DOM sont maîtrisés. Ils permettent une navigation aisée dans un document, ils nécessitent cependant la présence du document entier en mémoire.

Les processeurs SAX (" SAX parsers ") présentent une interface de plus bas

niveau mais ils sont plus performants (les processeurs DOM utilisent souvent un processeur SAX en interne).

## 5.5. Les processeurs XSL

Un processeur XSL applique une feuille de style XSL à un document XML. Actuellement, seule la partie transformation de XSL, appelée " XSLT ", étant normalisée, les outils se limitent à la transformation pour générer soit du HTML (ou XHTML) soit du XML.

Les processeurs XSL sont souvent disponibles à la fois en tant que programmes générant des fichiers en sortie et comme des bibliothèques à intégrer à des programmes applicatifs ou à des navigateurs.

Parmi les processeurs XSL on peut citer :

- composant ActiveX XMLDOM de Microsoft intégré dans Internet Explorer 5 et utilisable aussi bien dans un navigateur (IE4 ou IE5) que dans un serveur Web ;
- XT de James Clark ;
- Saxon d'ICL ;
- Xerces d'Apache.

## 5.6. Les composants divers

Il s'agit de produits utiles dans la mise en œuvre de projet d'échanges électroniques professionnels à base d'XML.

- Objecteering de Softeam et Mega Suite de Mega International. Ces outils permettent de modéliser en UML et intègrent des fonctions d'importation et

d'exportation de modèles UML en XML ;

- Language Omnimark qui intègre les traitements de documents XML ;

- FOP pour la transformation XML en PDF ;

- Xdex de Sequoia, le moteur d'indexation XML.

# Les Echanges Electroniques Professionnels

## Présentation

Le terme générique d'EEP recouvre les différents moyens et les différentes techniques qui sont utilisés pour permettre à un système d'information de communiquer avec un autre système d'information avec un minimum d'interventions humaines.

L'EDI conventionnel constitue la première technique utilisée dans le cadre des EEP depuis plus de dix ans. Elle constitue aussi la technique la plus utilisée en volume puisque dans certains secteurs les EEP représentent plus de 80 % du volume des messages échangés.

Les autres techniques disponibles sont :

Les avantages des EEP sont les suivants :

Diminution des coûts administratifs : Les données sont issues directement du système d'information de l'émetteur et sont transmises vers le destinataire sans manipulation. Chez le destinataire, les données au lieu d'être ressaisies peuvent être incorporé directement dans son système d'information.

Diminution des erreurs et des litiges : la suppression des manipulations humaines minimise les risques d'erreurs et les litiges résultants liés à ces manipulations.

Diminution des encours : l'accélération des processus de traitement des informations dans l'entreprise permet une diminution des différents encours de l'entreprise (encours de stocks, encours financiers, encours clients ou fournisseurs...)

Capacité d'accéder à de nouveaux clients ou de nouveaux marchés : certains prospects ou certains marchés ne sont ouverts que si l'entreprise est capable de converser de manière électronique.

Les échanges de l'entreprise avec ses partenaires par fonction

Ces différents tableaux donnent en première approche des messages qui peuvent être échangés par l'entreprise avec ses différents partenaires. Selon les secteurs d'activités, la localisation, la typologie des produits et des

services... la nature et la structure des messages peuvent être différents.

Relations avec les clients

Relations avec les fournisseurs

Relations avec les prestataires logistiques et de transport

Relation avec les sous-traitants de fabrication

Échanges en rapport avec les fonctions comptables et financières

Relations avec les banques et les organismes financiers

Paie et gestion du personnel