

Présentation générale de XML

Table des matières :

- 1 INTRODUCTION
- 2 BALISE STRUCTURELLE
 - 2.1 SEPARATION DU CONTENU ET DE LA PRESENTATION
 - 2.2 DTD ET SCHEMA
 - 2.3 DOCUMENT XML « BIEN FORME » ET DOCUMENT XML "VALIDE"
- 3 VUE D'ENSEMBLE
 - 3.1 HISTORIQUE
 - 3.2 PRINCIPES DE XML
 - 3.3 NOTIONS DE BASE
- 4 GLOSSAIRE: XML, HTML, XHTML ET CSS
 - 4.1 HTML
 - 4.2 XHTML
 - 4.3 CSS
 - 4.4 VERS LE DHTML
- 5 CONCLUSION

Date :2010

1 Introduction

XML (*eXtended Language Markup*) est devenu très rapidement un terme "branché" dans les conversations gravitant autour du Web. Les éditeurs de logiciels ou de sites Web tentent de faire croire qu'en intégrant du XML dans leurs applications, ils seront à la pointe de la technologie informatique. Le terme a été très vite galvaudé, et fait partie désormais du langage courant des informaticiens et marchands d'informatique.

OUI, le XML apporte de nombreuses solutions à des problématiques complexes en les abordant avec une nouvelle approche.

OUI, le XML va permettre de développer des applications plus rapidement avec plus de souplesse évolutive.

OUI, le XML est un langage puissant et son essor est légitime.

Mais NON, le XML n'est pas la solution à tous les maux : il ne règlera pas les problématiques de projets mal conçus, de Systèmes d'Information mal cartographiés, de programmes objets mal identifiés.

Il est nécessaire de ne pas tomber dans la gadgétisation de cette technologie en l'employant à mauvais escient.

Initialement utilisé lors d'échanges de données entre différents Systèmes d'Information hétérogènes, XML s'est largement imposé aussi dans des applications aussi variées que l'édition et la publication de documents techniques, de génération graphiques et comme une norme quasi incontournable pour le développement d'applications Web.

2 Balise structurelle

Le **balisage structurel** est un concept assez simple à appréhender. Il consiste à admettre que l'organisation d'un document texte permet à l'humain de l'interpréter grâce à des règles d'analyse intuitives comme :

Les marques typographiques.

Ex : la ponctuation qui rythme les phrases.

Des conventions de présentation.

Ex : les titres de chapitres et de paragraphes ont un corps plus élevés que le texte.

Des connaissances pragmatiques ou culturelles.

Ex : le contenu se lit de gauche à droite et de haut en bas. Les pages sont numérotées par ordre croissant. Les chapitres contiennent une introduction et une conclusion.

Ces intuitions ne sont qu'apparentes car elles ont été acquises et ancrées dans notre subconscient dès l'enfance avec l'apprentissage de la lecture. Un ouvrage sera présenté différemment dans une autre civilisation ayant une culture et une forme d'écriture très différente.

Les lecteurs peuvent lire un ouvrage car il respecte les règles de présentation et d'organisation des contenus en chapitres, sous-chapitres avec toutes les conventions de typographie créées depuis l'invention de l'écriture.

C'est la combinaison de l'utilisation de ces trois règles qui rend le document final cohérent et interprétable. L'image générée par l'application de ces règles impose par contre que le support soit au format papier. La présentation d'un ouvrage à destination d'un autre média (Web, PDA...) ou pour une autre utilisation devra être reprise dans son intégralité. Un non-voyant voulant accéder à l'ouvrage aura encore plus de difficulté.

2.1 Séparation du contenu et de la présentation

Le document XML va répondre à cette problématique en séparant le contenu strictement pragmatique des informations de mise en forme et de présentation.

Les "données" seront contenues dans le document XML et la "mise en forme" dans la feuille de style ou XSL (**eXtensible Stylesheet Language**).

Cette séparation de la description de la structure des documents et de leur présentation offre des avantages en termes de facilité d'échange, de production de documents, d'indépendance vis-à-vis des logiciels de traitement de l'information et aussi en termes de pérennité des documents sur le très long terme. Enfin, cette architecture ouvre des possibilités importantes en termes de traitement et d'échange d'information entre systèmes hétérogènes.

Le programme qui traitera le contenu se basera sur le document XML alors que l'humain qui souhaitera accéder à celui-ci s'aidera de la ou des feuille(s) de présentation pour visualiser le document.

2.2 DTD et Schéma

Si une communauté d'utilisateurs souhaite optimiser les possibilités d'échange et de traitement de documents, il va falloir définir une structure garantissant l'homogénéité des documents. Cette structure s'appelle la **DTD (Document Type Definition)** ou **Schéma**, elle est appliquée systématiquement aux documents et vérifie le respect de la syntaxe et de la sémantique des documents.

La DTD ou Schéma est la description exhaustive de tout ce qui est autorisé ou interdit au sein du document. Cette description est faite dans un fichier au format XML. Ainsi un document XML se conformant à cette description est **valide**.

2.3 Document XML « bien formé » et document XML "valide"

Il ne faut pas confondre document "bien formé" avec document "valide". Le premier respecte simplement la syntaxe XML alors que le second respecte en plus les contraintes lexicales, sémantiques et grammaticales du contenu.

Ces concepts autour du XML enrichis d'un certain nombre de mécanismes, de langages et d'outils vont permettre d'imaginer de nouvelles applications logicielles.

XML n'est pas simplement un concept ou encore une norme mais bien un langage à part entière. C'est pour cette raison qu'il est au cœur d'autant d'ouvrages aussi variés les uns que les autres et qu'il a pris aussi rapidement une place très importante dans l'évolution constante des Systèmes d'Information des entreprises.

3 Vue d'ensemble

XML est l'aboutissement de différents axes de recherches qui grâce au W3C (**World Wide Web Consortium**) ont été rassemblés dans ce qui est désormais appelé le XML.

3.1 Historique

Le W3C a pour objectif de formaliser des concepts pour améliorer les échanges d'informations ou de données sur le Web. Des normes sont publiées périodiquement et leur utilisation garantit aux éditeurs de logiciels notamment, la bonne interprétation des informations publiées par l'ensemble des acteurs concernés.

Les travaux autour de la conception de XML, coordonnés par le W3C, ont débuté véritablement en 1996 avec la création d'un groupe de travail dédié à cette tâche. Les acteurs majeurs, industriels ou organismes publics, ont constaté les limites du langage HTML. À cette époque, la norme internationale pour l'échange de documents structurés était SGML (**Standard Generalized Markup Language**). Le lancement du XML est le 10 février 1998, avec la publication de ce que le W3C appelle une "recommandation", pour la version 1.0 du langage XML. En réalité, cette recommandation est une norme industrielle de fait, adoptée et soutenue par l'ensemble des pairs. Des recommandations successives complèteront régulièrement les années suivantes la version initiale du 10 février 1998.

Les mécanismes normalisés associés aux feuilles de style jouent un rôle plus important qu'avec le HTML. Le CSS (**Cascading Style Sheet**) en 1998 normalise l'affichage de contenu sur des plates-formes Web. Un autre mécanisme encore plus puissant que CSS est le XSL (**eXtended Stylesheet Language**). Sa première recommandation date de 1999.

XML est une norme en constante évolution et enrichissement avec comme contrainte industrielle la compatibilité croissante des recommandations. Seul le W3C dispose de la liste des publications à jour (www.w3c.org).

XML s'inscrit dans l'histoire de l'informatique et se positionne dans le prolongement des grands langages comme Fortran, Cobol, Lisp, ou plus récent comme le C++ ou Java.

La version 1.0 de HTML est restée trop longtemps sans réelles évolutions (presque six ans). Ramené à l'échelle de temps de l'informatique, c'est beaucoup trop.

Les évolutions des différents langages de programmation ont été permanentes, mais le processus s'est accéléré après 1995.

L'histoire du XML est récente mais son passé est déjà bien riche en évolutions comme en projets développés.

L'axe du temps n'est pas figé : de nouvelles recommandations sont publiées périodiquement. Elles sont consultables sur le site du W3C.

3.2 Principes de XML

La balise structurale n'est pas suffisante à elle seule pour décrire un langage dont le but principal est de **stocker, fournir, véhiculer, publier** de l'information ou du contenu. Il est nécessaire d'enrichir ce concept avec une notion supplémentaire de :

- hiérarchie et d'arborescence avec les **nœuds**,
- type de contenu avec les **éléments** ou **balises**,

- propriétés de ces éléments grâce à des **attributs**,
- contenu dans le texte et enfin d'une déclaration indispensable avec le **prologue**.

Ce sont ces principales notions qui définissent le langage XML dans sa version 1.0.

La notion de document structuré peut s'illustrer de la manière suivante :

Soit un contenu présenté dans un document au format texte enrichi (RTF dans wordpad) et qui est le suivant :

```
{\rtf1\ansi\ansicpg1252\deff0\deflang1036{\fonttbl{\f0\fswiss\fpr
q2\fcharset0 Arial;}}{\f1\fswiss\fcharset0 Arial;}}
{\*\generator Msftedit
5.41.15.1515;}\viewkind4\uc1\pard\sbl20\qj\f0\fs22 Objet\~:
Livre\par
Titre\~: Xml et Transformation \par
Editeur\~: EBO\par
Auteur\~: auteur1\par
Auteur\~: auteur2\par
\pard\f1\fs20\par
}
```

Le RTF est un format que les éditeurs tels que Word, Wordpad ou autres outils du monde libre savent interpréter car il est normalisé.

Dans un document RTF, le contenu et la présentation sont "mêlés" car il n'y a pas de séparation entre le fond et la forme. Seuls les éditeurs sachant interpréter cette norme sauront lire ces documents. Pour que la présentation soit identique quel que soit l'outil de lecture choisi, il est impératif que tous interprètent la norme de la même manière. Or, sur les fonctions évoluées du format RTF, ce n'est pas toujours vérifié.

Voici la même information enregistrée dans un document au format XML :

```
<?xml version="1.0" encoding="iso-8859-1" ?>
<document>
  <objet>Livre</objet>
  <titre>Xml et transformation</titre>
  <editeur>EBO</editeur>
  <auteurs>
    <auteur>auteur1</auteur>
    <auteur>auteur2</auteur>
  </auteurs>
</document>
```

Au moins trois informations fondamentales apparaissent à la lecture de ce document :

Il n'y a pas de notion de définition de présentation apparente.

Il existe une notion de structure de document avec une définition sémantique du contenu.

Une hiérarchie existe entre les différentes valeurs. Les deux auteurs sont regroupés sous une même entité **auteurs** (au pluriel).

En comparant la présentation RTF et XML, il en ressort que dans le RTF c'est la présentation graphique qui indique au lecteur qu'il existe un lien entre les notions affichées

(l'identification de l'ouvrage avec son titre et son éditeur). Si ce contenu avait été éparpillé dans la page, la relation entre les valeurs n'aurait pas été nécessairement intuitive.

Alors que dans le document XML, les liens entre les valeurs sont donnés dans la façon de ranger l'information. Pour ce faire, l'utilisateur doit décider des règles à appliquer.

Les mécanismes du XML permettent tout d'abord de saisir du contenu ou de le mettre à jour :

- Sans se préoccuper de la présentation ou des traitements qui seront appliqués.
- Sans avoir à saisir des libellés tels que "auteur", "objet", "titre".

Puis, il est possible de générer automatiquement :

- Différentes formes de présentations (liste, énumération, tableaux...).
- En appliquant si nécessaire des tris, sélections, réorganisations...
- Sur différents supports ou médias (écran, papier, terminal Braille...).

Les logiciels pourront interpréter et exploiter avec intelligence le contenu de ces documents explicite grâce à un balisage spécifique et uniquement lié à l'information qu'il encadre et non lié à l'application qui traite cette information.

3.3 Notions de base

La syntaxe de XML s'appuie sur les définitions suivantes.

3.3.1 Les éléments

Une **balise** ou **élément** va encadrer le contenu syntaxique qui lui correspond. Le nom de la balise, comme le nom d'un champ dans une table relationnelle peut avoir une signification.

```
<titre>XML bible</titre>
```

Cette notion de signification de nom de balise doit être respectée au mieux car le but du XML est bien que le fichier puisse être interprété par l'humain sans aucune aide. Dans l'exemple, appeler la balise qui contient le titre de l'ouvrage "titre" est une information supplémentaire.

3.3.2 Les nœuds

Les balises vont être hiérarchisées entre elles sous forme d'arbre. Certaines balises n'auront pas nécessairement de contenu mais une liste de balises. Ces balises sont dites les balises "enfants" de la balise "mère". Le lien entre une balise "mère" et "enfant" s'appelle un nœud. La structure ainsi définie décrit toutes les valeurs attachées à un objet.

Une balise "enfant" devient elle-même "mère" si elle possède des balises "enfants".

La présentation tabulée dans les différents éditeurs ou navigateurs facilite la lecture humaine.

```
<livre>
  <titre>XML bible</titre>
  <refauteur>A1</refauteur>
  <prix>32</prix>
</livre>
```

La balise "livre" est mère des balises "titre", "refauteur" et "prix", et l'organisation de cette hiérarchie se fait par un nœud.

Sans les nœuds, les balises du document XML ne seraient pas affectées au bon objet. Dans l'exemple, si le document contient la description de plusieurs ouvrages, il est nécessaire de

regrouper entre elles les balises faisant référence au même livre. Le document aura la forme suivante :

```
<livre>
  <titre>XML bible</titre>
  <refauteur>A1</refauteur>
  <prix>32</prix>
</livre>
<livre>
  <titre>UML et bases de données</titre>
  <refauteur>A2</refauteur>
  <prix>52</prix>
</livre>
```

3.3.3 Les attributs

Il est possible d'ajouter aux balises une notion supplémentaire appelée *attribut*. Elle correspond à une notion de propriété. À une balise peuvent correspondre plusieurs attributs, mais l'attribut prend une valeur unique. Dans l'exemple suivant, l'attribut "monnaie" ne peut être que € pour la balise qu'il décrit.

```
<prix monnaie = "€" >32</prix>
```

Exemple de création de plusieurs balises "prix" avec des attributs "monnaie" de valeurs différentes.

```
<prix monnaie = "€" >32</prix>
<prix monnaie = "$" >26</prix>
```

3.3.4 Le texte

L'intérêt du document XML est sa souplesse dans l'intégration de contenu texte car aucune longueur ne doit être spécifiée. Il est aussi possible d'intégrer de la mise en forme HTML.

```
<resume>Un ouvrage décrivant la syntaxe et la manipulation de
documents XML. Des exemples et des définitions sont proposés.</resume>
```

Voici le même document intégrant les balises HTML de présentation de contenu.

```
<resume><html>Un ouvrage décrivant la <i>syntaxe</i> et la
manipulation de documents <b>XML</b>.</br>Des exemples et des
définitions sont proposés. </html></resume>
```

3.3.5 Le prologue

Il est indispensable de spécifier l'environnement dans lequel le document XML est défini. Cette syntaxe est impérative dans chaque document XML. Sans cette syntaxe les parseurs, validateurs ou autres navigateurs ne sauront pas relire "intelligemment" le document (c'est-à-dire en interprétant tout le sens décrit dans la structure XML). Les documents seraient alors lus comme de vulgaires fichiers textes.

Voici le prologue le plus généralement rencontré dans les fichiers XML.

```
<?xml version="1.0" standalone="no" encoding="iso-8859-1"?>
```

La version est toujours "1.0". Le paramètre **encoding** spécifie les tables de caractères utilisées dans le document. L'attribut **standalone** spécifie si pour son interprétation le document est autonome ou s'il doit faire référence à d'autres documents (DTD, schémas, XSL...).

4 GLOSSAIRE: XML, HTML, XHTML et CSS

XML est lui-même né du langage SGML (**Standard Generalized Markup Language**). Et à partir de l'expérience acquise du langage XML il s'est décliné au moins un autre langage le XHTML (**eXtensible Hyper Text Markup Language**).

4.1 HTML

Le HTML (**Hyper Text Markup Language**) date de 1990. Il a été inventé par un scientifique nommé Tim Berners-Lee, pour la création de pages Web. Il utilise une structure construite avec des balises permettant une certaine mise en forme du contenu (texte, images, objets...) dans la page. Pour visualiser le contenu et apprécier le rendu, il est nécessaire de passer par un navigateur interprétant ce langage. Le HTML est le langage universel pour le développement des sites Web. Il se doit d'être simple pour qu'un plus grand nombre puisse le comprendre. La richesse de sa syntaxe permet la création de pages avec des chartes graphiques évoluées. Par contre, toute modification de présentation ou de contenu nécessite une modification et validation du code source. Le concept principal est le **tag**, le HTML est donc un langage de marque logique.

Les différents navigateurs présents sur la toile n'interprètent pas tous les différents tags de la même façon et peuvent en avoir de spécifiques. Une même page peut avoir des rendus différents selon les navigateurs.

Voici un exemple de code HTML :

```
<html><h1>XML et XSL</h1>Un ouvrage décrivant la <i>syntaxe</i>
et la manipulation de documents <b>XML</b>.</br>Des exemples et
des définitions sont proposés. </html>
```

Il est possible de présenter du texte au kilomètre avec des balises de séparation mais il est surtout possible de présenter des tableaux et des listes ordonnées.

Voici un exemple de liste définie par les balises **ul** et **li** :

```
<ul>
  <li>Enumération 1</li>
  <li>Enumération 2</li>
  <li>Enumération 3</li>
  <li>Enumération 4</li>
</ul>
```

Voici un exemple de tableau de 2 colonnes et 2 lignes avec les balises **tr** et **td** :

```
<table>
  <tr>
    <td>Colonne 1 , ligne 1</td>
    <td>Colonne 2 , ligne 1</td>
  </tr>
  <tr>
    <td>Colonne 1 , ligne 2</td>
    <td>Colonne 2 , ligne 2</td>
  </tr>
</table>
```

La version du HTML est actuellement la 4.01. Le caractère figé de la page est désormais une contrainte majeure de ce langage. L'émergence de sites Web toujours plus complets et plus riche en contenu nécessite l'apport de moyens plus performants.

4.2 XHTML

Le XHTML (**eXtensible Text Markup Language**) est un langage hybride entre le XML et le HTML. C'est plus une reformulation du HTML pour le rendre compatible avec XML qu'un langage à part entière. Il comble le manque du HTML qui ne sait pas dissocier le fond de la forme. Le XHTML grâce à cette dissociation donne une plus grande place à la sémantique. La façon dont le contenu est organisé au sein du document lui donne un sens supplémentaire. Une partie du code est réservée à la définition du rendu. Il est donc facile de multiplier les formes de présentation pour un même contenu sans avoir à reprendre le document et plus particulièrement son contenu.

```
<?xml version="1.0" encoding="iso-8859-15" lang="fr"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
  'http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd'>
<html>
  <head>
    <title>XHTML Exemple</title>
  </head>
  <body>
    <p>Chapitre <b>I</b></p>
  </body>
</html>
```

Les navigateurs dans lesquels le contenu est affiché n'utilisent pas le même analyseur syntaxique si le code est du HTML ou du XML. Les analyseurs ne contrôlent pas de la même manière les syntaxes de documents XML ou HTML. Il en résulte qu'une page mal formée en HTML peut quand même apparaître à l'écran alors qu'en XML ou XHTML elle sortira en erreur et ne s'affichera pas.

Il y a une très bonne raison à cela et elle est stratégique. Le HTML a été conçu pour publier du contenu sur le Web et il est préférable de l'afficher même avec des erreurs que de ne rien pouvoir proposer.

Par contre, le XML a besoin d'être plus strict. Il a donc été convenu que si une erreur était détectée rien ne serait affiché. Il ne peut donc pas y avoir d'erreur d'interprétation du contenu.

4.3 CSS

Le CSS (**Cascades Style Sheet**) est le complément au XHTML et permet uniquement de traiter la présentation de la page Web. De manière schématique, le contenu est dans le XHTML, la présentation de la partie graphique dans la CSS et le rendu dans le navigateur est l'interprétation de l'un par l'autre.

L'atout majeur d'une telle séparation des "métiers" (organisation du contenu et présentation de celui-ci) facilite les mises en forme multiples en fonction des différents médias de destination ou périphériques de sortie (écran, imprimante...). Le principe de base est le suivant : à un fichier de contenu (XHTML) est lié un ou plusieurs fichier(s) de présentation (CSS). En fonction de critères subjectifs, l'un de ces fichiers CSS sera appliqué au document. Ces fichiers CSS vont contenir toutes les propriétés d'affichage de chacune des balises du document XHTML.

Les propriétés élémentaires sont classées par groupes, en voici quelques exemples :

Les polices de caractères :

```
font-style:italic;
font-variant:small-caps;
font-weight:bold;
font-size: small;
```

Les arrière-plans :

```
background-color:#f00;
```

```
background-image:url(background.gif);
```

Les bordures :

```
border-width:1px;  
border-style:solid;  
border-color:#000000;
```

Un intérêt supplémentaire à cette séparation de contenu et de présentation est le poids des fichiers générés. En effet cette organisation permet de fabriquer des fichiers beaucoup plus petits et donc plus rapides à transmettre et afficher sur le Web.

4.4 Vers le DHTML

Le "**D**" veut dire Dynamique.

Le HTML a permis de construire des sites Web avec une richesse graphique qui les rendaient attractifs. Le XHTML a ajouté la possibilité de décliner les présentations jusque-là unique ou difficile à paramétrer. Mais la modification du contenu d'une page dans le navigateur nécessite un recalcul du rendu et un réaffichage complet de la page, le DHTML permet désormais de manipuler des éléments dans la page HTML à l'aide de scripts. Les actions sollicitées par l'internaute ne nécessitent pas lancer une nouvelle requête au serveur de l'application. Le poste client traitant le script en local, il est possible de s'affranchir des contraintes liées au réseau et à la performance du serveur sur lequel est hébergé le site Web. Le DHTML résulte en fait de la combinaison de pages HTML, de CSS et de scripts java et d'objets dynamiques.

Il va dynamiser le site Web par des actions plus souples dans l'application. Le site va ressembler de plus en plus à un programme installé en local sur le serveur avec des menus déroulants, des mises à jour de présentation immédiates et des traitements sophistiqués. Des portions de pages vont s'animer indépendamment les unes des autres et des pages vont pouvoir être construites dynamiquement en fonction de choix utilisateurs ou de choix de médias de destination.

Bien que la souplesse et la richesse des styles CSS font qu'ils sont l'avenir de la mise en forme sur le réseau, le HTML s'étant généralisé très largement pour la publication sur le Web, la norme CSS s'impose lentement et les sites en HTML pur ne sont pas prêts de disparaître. Les deux technologies vont cohabiter encore quelque temps.

4.4.1 Le javascript

Contrairement au HTML, dont l'objectif est de ne pas intégrer de code programme pour que même les acteurs non spécialistes puissent s'y intéresser, le code du DHTML, dans sa partie scripts, nécessite de très bonnes connaissances de la programmation java.

Techniquement, l'intégration du script java peut se faire au sein de la page mais ce n'est pas pertinent pour deux raisons :

- Le contenu de la page est alourdi.
- Le script n'est pas réutilisable dans une autre partie de l'application.

Il est donc nécessaire de le placer dans un fichier dédié avec une déclaration à son accès dans la page, de la manière suivante :

```
<script type="text/javascript" src="mesfonctions.js"></script>
```

Le fichier mesfonctions.js va contenir l'inventaire des fonctions programmées nécessaires dans la page. Comme dans tout programme, l'accès à des variables locales ou globales, à des paramètres et à des constantes est possible.

4.4.2 Les objets DOM

Le DOM (**Document Object Model**) permet d'accéder et de manipuler une page de l'application, des propriétés du navigateur dans lequel la page est affichée et enfin d'interagir avec les ressources locales du système d'exploitation du poste client.

Des "événements" sont captés et permettent une réelle interaction entre l'utilisateur et la page Web consultée. Il est possible par exemple d'appeler une fonction javascript par le simple passage de la souris sur un objet identifié de la page.

L'événement **onmouseover** est déclaré avec l'appel d'une fonction paramétrée :

```
<cite onmouseover="maFonction(event, this)">
</cite>
```

La déclaration de la fonction appelée dans l'événement peut être écrite dans un fichier java spécifique :

```
function maFonction(event, toto)
{
    alert("Voici un message Event déclenché par l'utilisateur")
}
```

Le DOM est une interface permettant de standardiser l'accès à un document balisé, entre autres une page d'un site Web. La hiérarchie définie dans la structure XML des documents organise toutes les propriétés ou méthodes nécessaires au développeur pour manipuler ou programmer des pages dynamiques.

Ces évolutions majeures vont autoriser une nouvelle manière d'appréhender le développement de sites Web, en les rendant beaucoup plus interactifs et conviviaux et en donnant aux développeurs les moyens de mutualiser leurs développements grâce aux concepts de programmation par objet. Le développement d'application Web rentre désormais dans un processus industriel mais réservé à une élite professionnelle compétente.

5 Conclusion

Il est à prévoir que l'usage d'XML va déborder largement le WWW en provoquant la convergence de deux mondes informatiques jusqu'ici séparés : celui des documents et celui des données. Il est très probable qu'il va de ce fait devenir très rapidement la "lingua franca" de l'informatique parlée tout autant par les SGBD que par les outils de bureautique et de documentation, les logiciels de gestion ou les applications techniques et scientifiques. Le XML va rendre possible une automatisation des activités administratives et logistiques sans commune mesure avec les possibilités actuelles. Et il va considérablement simplifier l'EDI (**Échanges de Données Informatisés**).