

TABLE DES MATIERES

INTRODUCTION GENERALE.....	1
CHAPITRE I : INGENIERIE DES BESOINS.....	
1.1 Introduction.....	5
1.2 Définition.....	5
1.3 Les enjeux.....	5
1.4 Document d'exigences	6
1.4.1 Définition du document d'exigences.....	6
1.4.2 Définition des exigences	6
1.4.3 Contenu d'un document d'exigences	7
1.5 IEEE pratique recommandée pour les spécifications d'exigences de logiciel.....	7
1.6 Missions et Objectifs de l'ingénierie des besoins.....	8
1.7 Conclusion.....	12
CHAPITRE II : DONNEES SEMI STRUCTUREES.....	
2.1 Introduction.....	14
2.2 Qu'est-ce que le XML ?	14
2.2.1 Définition.....	14
2.2.2 Un exemple de document XML.....	16
2.2.3 Les différentes technologies XML.....	16
2.2.3.1 DTD XML.....	16
2.2.3.2 SCHEMA XML.....	17
2.2.3.3 XPATH XML.....	19
2.2.3.4 XSL XML.....	20
2.2.3.5 XQUERY XML.....	21
2.2.4 Les avantages de XML.....	22
2.2.5 Conclusion.....	23

CHAPITRE III : CONTRIBUTION.....	
3.1 Introduction.....	25
3.2 Structuration logique.....	25
3.3 Visualisation.....	29
3.3.1 MultiMedia.....	29
3.3.2 Hypertexte.....	29
3.3.3 Hypermédia.....	30
3.3.4 Texte dynamique.....	30
3.4 Interrogation des données.....	30
3.5 Autres outils.....	31
3.5.1 Sigil.....	31
3.5.1.1 Table des matières avec Sigil.....	32
3.5.1.2 L'index avec Sigil.....	32
3.5.2 Calibre.....	32
3.5.2.1 Gestion de bibliothèque.....	32
3.5.2.2 Conversion d'e-books.....	34
3.5.2.3 Synchronisation avec de nombreux appareils.....	35
3.5.2.4 Téléchargement & conversion de news.....	35
3.5.2.5 Visionneuse d'e-books intégrée.....	36
3.5.3 Conclusion.....	36
CONCLUSION GENERALE.....	38
REFERENCES BIBLIOGRAPHIQUES	40

LISTE DES FIGURES

Figure 1.1	Le fondement de l'ingénierie des besoins.....	9
Figure 1.2	Le cycle <acquisition, abstraction, validation>.....	10
Figure 2.1	Arborescence document xml.....	16
Figure 2.2	DTD XML.....	17
Figure 2.3	XSD ADVANTAGE.....	18
Figure 2.4	XML AND XPATH.....	20
Figure 2.5	Principe de XSLT XML.....	21
Figure 2.6	Principe de XQUERY XML.....	22
Figure 3.1	Document XML du scénario principal du cas d'utilisation « faire une réservation ».....	25
Figure 3.2	Document XSD du Scénario principal des cas d'utilisations.....	26
Figure 3.3	Document XML des exigences fonctionnelles du cas d'utilisation «faire une réservation».....	26
Figure 3.4	Document XML des exigences Non fonctionnelles du cas d'utilisation «faire une réservation».....	27
Figure 3.5	Document DTD du Document d'exigence des cas d'utilisations.....	27
Figure 3.6	Document XSD du document d'exigences des cas d'utilisations.....	28
Figure 3.7	Document XML du cas d'utilisation «faire une réservation».....	28
Figure 3.8	Document XML du projet général regroupant tous les cas d'utilisations.....	29

Figure 3.9	Structure de Sigil.....	31
Figure 3.10	Gestion de la bibliothèque calibre.....	31
Figure 3.11	Comment convertir PDF en EPUB, MOBI.....	34
Figure 3. 12	Visionneuse de livre numérique de calibre.....	36

INTRODUCTION GENERALE

La réussite d'un projet dépend avant tout, des délais, des couts et de la qualité... de la bonne traduction du besoin exprimé par le métier en exigences auxquelles le futur système devra répondre.

Une mauvaise gestion et expression des exigences est souvent le premier facteur d'échec des projets avec des impacts forts sur les couts et les délais.

Les principaux éléments d'échec sont :

- des attentes non réalisables des utilisateurs.
- des exigences incomplètes.
- des exigences qui ne répondent plus (ou pas) à des besoins.
- des exigences et des spécifications changeantes.

L'Ingénierie des Besoins (IB) est concernée par l'identification de buts assignés au système envisagé et l'opérationnalisation de ces buts en contraintes et exigences imposées au système et aux agents qui en assureront le fonctionnement. L'IB peut être vue comme le processus qui permet de transformer une idée floue en spécification précise des besoins servant de support à la spécification du système et de ses interfaces avec l'environnement. Le processus d'IB inclue donc des activités de découverte, de spécification, de négociation et de validation des besoins. Au cours de l'IB, les besoins à l'égard du système sont découverts, négociés, validés, spécifiés et répertoriés dans des documents de spécification des besoins. Produire un document des besoins de qualité est difficile et crucial. De nombreuses enquêtes confirment en effet l'importance de l'IB dans la réussite des projets de systèmes d'information.

On a pris l'habitude d'exprimer ce qu'on attend d'un système ou d'un produit sous forme d'un ensemble d'exigences. Une exigence prescrit une propriété dont l'obtention est jugée nécessaire. Son énoncé peut être une fonction, une aptitude, une caractéristique ou une limitation à laquelle doit satisfaire un système, un produit ou un processus.

Les exigences guident toutes les étapes du processus de développement depuis le cahier des charges jusqu'à l'acceptation du système. Elles formalisent les besoins et contraintes ainsi que les engagements des parties prenantes.

L'Ingénierie des Exigences consiste, au travers de méthodes, règles et processus, à établir et maintenir un référentiel unique qui s'affine et se complète au cours du développement et qui est maintenu tout au long de la vie du système :

Les parties prenantes utilisatrices et exploitantes expriment leurs besoin et contraintes qui sont formalisés en exigences initiales (généralement dans un cahier des charges).

Le maître d'œuvre en déduit, en tenant compte des contraintes des parties prenantes impliquées dans la réalisation, les exigences techniques système qui prescrivent ce que devra faire le système, avec quelles performances et sous quelles contraintes, dont les exigences industrielles (le système doit être fabricable, vérifiable, maintenable, etc). Ces exigences constituent la spécification du système.

Au cours de la conception, les exigences système sont allouées aux éléments de la décomposition du système, tandis que les choix techniques induisent de nouvelles exigences. On obtient ainsi les exigences techniques de constituants (ainsi que celles des produits et processus contribuant au cycle de vie du système et de ses constituants).

Au cours de la réalisation, les constituants et, in fine, le système sont vérifiés par rapport à leurs exigences techniques et validés par rapport à leur exigence allouée (aux exigences initiales pour le système complet).

Dans le travail en cours [22] à l'Université de Virginie, les auteurs considèrent les questions liées à l'automatisation de la gestion de la documentation du logiciel pour mieux augmenter son utilité. Ils décrivent un système prototype, SLEUTH, actuellement à l'étude en tant que véhicule pour la gestion de la documentation du logiciel. Le prototype conserve la documentation du logiciel comme un hypertexte avec des liens typés aux fins de la navigation par les utilisateurs ayant des besoins différents. Ces liens sont générés automatiquement par le système et conservés précis dans la mise à jour. Des requêtes ad hoc sont fournies sur les hyperdocumentation.

Selon les auteurs dans [23] différents modes de réalisation d'un système et d'un procédé pour générer des documents logiciels sont décrits. Un générateur de documentation logicielle peut intégrer une variété de sources d'informations. Le générateur de documentation logicielle peut analyser chaque source pour déterminer son type et utiliser cette information pour extraire des données de documentation de la source en fonction du format des données incluses dans cette source. Le générateur de documentation logicielle peut reformater les

données de documentation extraites de chaque source dans un format uniforme et agréger les données uniformément formatées en un document d'entrée agrégé uniformément formaté. Le générateur de documentation logicielle peut transformer ce document en un ou plusieurs ensembles de documentation logicielle dans un format spécifique.

Le présent travail de master rentre dans le domaine de l'ingénierie des besoins. Il consiste à faire une réingénierie du document des exigences du logiciel selon une approche de données semi-structurées. En effet la manipulation de ce document complexe dans un format non révisable tel que le format PDF ou même révisable tel que le format Word, pose beaucoup de problèmes, dont le principaux : la difficulté de chercher un contenu précis dans ce document, de mettre à jour son contenu et enfin de le lire. Ainsi nous considérons dans le contexte de ce mémoire le contenu de ce document comme un ensemble de données semi-structurées.

Le présent mémoire est structuré comme suit. Après cette introduction le chapitre 1 présente le domaine d'ingénierie des besoins.

Dans le deuxième chapitre, nous présentons en détails le langage XML et ses technologies qui ont permis la conception et la mise en place de notre document électronique.

Dans le dernier chapitre, nous expliquons la mise en place technique en définissant les outils utilisés, en y joignant des captures d'écran de notre document électronique final.

CHAPITRE I

INGENIERIE DES BESOINS

1.1 Introduction

L'ingénierie des besoins (IB) constitue une étape cruciale dans la construction de systèmes. Elle couvre différentes activités comme la collecte, l'analyse, la spécification, la vérification et la validation des exigences. Cependant, dans un monde où les systèmes sont de plus en plus complexes, leur taille croît, leur degré de criticité devient de plus en plus élevé, leur inter-connectivité est grandissante, ainsi la maîtrise de cette complexité constitue une difficulté majeure. Cela passe nécessairement par une gestion approfondie et efficace des exigences auxquelles ces systèmes sont censés répondre [1].

1.2 Définition

L'ingénierie des besoins et des exigences est la principale cause de l'échec (annulation avant la fin) ou de difficultés (hors budget/délais, moins de fonctionnalités) d'un projet mais... L'ingénierie des besoins et des exigences est également la principale cause du succès d'un projet ! [2]

On peut définir l'ingénierie des besoins comme suit :

- Transformer une "idée floue" en une "description précise"
- Prendre en considération les besoins exprimés par une communauté d'utilisateurs (au sens large)
- Définir la relation existante entre un système et son environnement.

1.3 Les enjeux

Face à ce constat, l'IB doit répondre à de nouveaux défis pour prendre en compte en particulier le caractère pluridisciplinaire des exigences de systèmes complexes ainsi que l'évolution continue et inévitable de ces mêmes systèmes. L'IB requiert une expertise de plus en plus poussée dans de nombreux domaines, habituels comme l'informatique ou les systèmes d'information, mais aussi dans des domaines comme l'ingénierie système en général ou la gouvernance et l'économie des entreprises, la sociologie, la linguistique, etc. En effet, l'IB nécessite de savoir communiquer et faire communiquer des acteurs de ces différentes disciplines afin de comprendre, modéliser et analyser les exigences des systèmes à construire ainsi que d'appréhender les contextes dans lesquels ces systèmes seront utilisés. De plus, dans un monde en constante évolution, où les utilisateurs ont de plus en plus de besoins, les organisations des entreprises changent, de nouvelles lois et réglementations apparaissent, sans parler du développement rapide des nouvelles technologies et de l'apparition de nouveaux paradigmes de développement comme les méthodes agiles ou les lignes de produits, assurer que les systèmes continuent de satisfaire aux propriétés attendues est un véritable défi. Ce défi

adresse, entre autres, la problématique de la gestion des changements et de l'évolution des exigences [3].

1.4 Document d'exigences

1.4.1 Définition du document d'exigences

Un document d'exigences explique pourquoi un produit est nécessaire, met le produit dans le contexte et décrit à quoi le produit fini ressemblera. Une grande partie du document d'exigences est la liste formelle d'exigences [4].

1.4.2 Définition des exigences

Les exigences incluent des descriptions de propriétés de système, spécifications pour comment le système devrait travailler et les contraintes placées après le processus de développement. Généralement, les exigences sont des déclarations de ce qu'un système devrait faire, plutôt que comment il devrait le faire. Les réponses à comment les questions tombent dans le royaume de design. Les spécifications d'exigences ne devraient pas inclure des solutions de design (à part les exigences d'interface, qui incluent souvent le design fixé).

Les exigences viennent des utilisateurs finaux, des clients et quelquefois des promoteurs. Les utilisateurs finaux ont tendance à exposer des exigences dans le descriptif ou les termes de récit ("je voudrais un écran bienvenu avec les liens vers les choses que j'utilise régulièrement et je veux être capable de choisir de deux ou trois différents arrangements de couleurs pour l'écran bienvenu") qui aurait besoin d'être décomposé en déclarations d'exigence individuelles. Les clients, qui peuvent bien se distinguer des utilisateurs finaux, sont les gens qui paient pour le développement du système. Leurs exigences seront souvent exposées du point de vue des prix. Les promoteurs pourraient faire rattacher des exigences à la performance de système et à d'autres thèmes techniques.

Les classes d'exigences incluent d'habitude les exigences d'utilisateur, les exigences de système et les exigences d'interface ; d'autres classes d'exigences sont aussi incluses comme nécessaires. Les exigences d'utilisateur sont écrites du point de vue d'utilisateurs finaux et sont généralement exprimées dans la forme de récit : L'utilisateur doit être capable de changer l'arrangement de couleurs de l'écran bienvenu. Les exigences de système sont des spécifications exposées en détail décrivant les fonctions que le système a besoin de faire. Ceux-ci sont plus techniques d'habitude dans la nature : Le système inclura quatre arrangements de couleurs programmés pour l'écran bienvenu. Les couleurs doivent être

spécifiées pour le fond de page, le texte, a visité des liens, des liens non visités, des liens actifs et des boutons (la base, le point culminant et l'ombre). Les exigences d'interface spécifient comment l'interface (la partie du système que les utilisateurs voient et communiquent) regardera et se comportera. Les exigences d'interface sont souvent exprimées comme les maquettes d'écran ; les récits ou les listes sont aussi utilisés [4].

1.4.3 Contenu d'un document d'exigences

En plus de la liste des exigences, le document devrait expliquer pourquoi le produit est nécessaire, et décrit à quoi le produit fini ressemblera.

Pour mettre le produit dans le contexte pour le développement, on décrit les outils de développement qui seront utilisés, les situations de recrutement et de budget de projet et le programme de développement approximatif. On inclue des détails de comment le produit fini sera distribué ou accédé et autres renseignements pertinents qui affectent le développement. Le document devrait inclure une vue d'ensemble du produit fini dans le récit et/ou la forme de maquette graphique pour ceux qui veulent recevoir une prise du projet sans lire toutes les exigences. Une vue d'ensemble est aussi utile pour les gens qui liront les exigences, pour qu'ils puissent avoir une compréhension générale du produit avant de creuser dans les spécifications.

Évidemment, l'échelle du projet influencera les contenus et la longueur d'un document d'exigences. Les projets grands, complexes auront probablement plus d'exigences. Les projets plus petits, plus rapides auront moins d'exigences et auront besoin des documents d'exigences plus courts, qui peuvent doubler comme les propositions de projet [4].

1.5 IEEE pratique recommandée pour les spécifications d'exigences de logiciel

Les documents de Normes d'IEEE sont développés dans les Sociétés IEEE et les Comités de Coordination de Normes de l'Association de Normes IEEE (IEEE-SA) la Planche de Normes. L'IEEE développe ses normes par le biais d'un processus de développement de consensus, approuvé par l'Institut de Normes national américain, qui réunit des volontaires représentant des points de vue variés et intéresse accomplir le produit fini. Les volontaires ne sont pas nécessairement des membres de l'Institut et du service sans compensation. Pendant que l'IEEE administre le processus et établit des règles de promouvoir l'impartialité dans le

processus de développement de consensus, l'IEEE n'évalue pas de façon indépendante, évalue ou vérifie l'exactitude de n'importe lequel des renseignements contenus dans ses normes.

L'utilisation d'une Norme IEEE est entièrement volontaire. L'IEEE nie la responsabilité pour toute blessure personnelle, propriété ou d'autre dommage, de toute nature du tout, si spécial, indirect, important, ou compensateur, en provenant directement ou indirectement de la publication, l'utilisation de, ou la dépendance sur cela ou autre document Standard IEEE. Cette pratique est basée sur un modèle dans lequel le résultat du processus de spécification d'exigences de logiciel est un document de spécification non équivoque et complet. Il devrait aider

- a) Les clients de logiciel pour exactement décrire ce qu'ils veulent obtenir ;
- b) Les fournisseurs de logiciel pour comprendre exactement ce que le client veut ;
- c) Les ingénieurs de besoins, concepteurs, développeurs, testeurs...

1.6 Missions et Objectifs de l'ingénierie des besoins

La plus ancienne définition de l'IB contient les « ingrédients » essentiels. Dans leur article prémonitoire, Ross et Schoman écrivent « requirements definition is a careful assessment of the needs that a system is to fulfil. It must say why a system is needed, based on current and foreseen conditions, which may be internal operations or an external forces. It must say what system features will serve and satisfy this context. And it must say how the System is to be constructed » [6].

En d'autres termes, l'ingénierie des besoins doit de prime abord s'intéresser aux buts du contexte organisationnel du système à développer parce qu'ils permettent de comprendre les raisons justifiant son développement. L'IB doit poser la question POURQUOI développer un système et aider les parties prenantes du projet à y répondre.

Le rôle de l'IB est ensuite de déterminer les fonctionnalités que le système doit mettre en œuvre pour aider à la satisfaction de ces buts et identifier les contraintes qui restreignent la mise en œuvre de ces fonctions. Ces buts, fonctions et contraintes, constituent les 'besoins' qui doivent ultérieurement être convertis en une spécification précise permettant le développement du système. L'évolution des besoins au cours du temps et la répercussion de leur évolution sur la spécification du système doit aussi être prise en compte [7].

La Figure 1.1 empruntée à Axel van Lamsweerde [Lamsweerde00] schématise cette vue de

l'ingénierie des besoins centrée sur les questions POURQUOI (WHY) et QUOI (WHAT) ainsi que la mise en correspondance des réponses inhérentes à l'une et à l'autre.

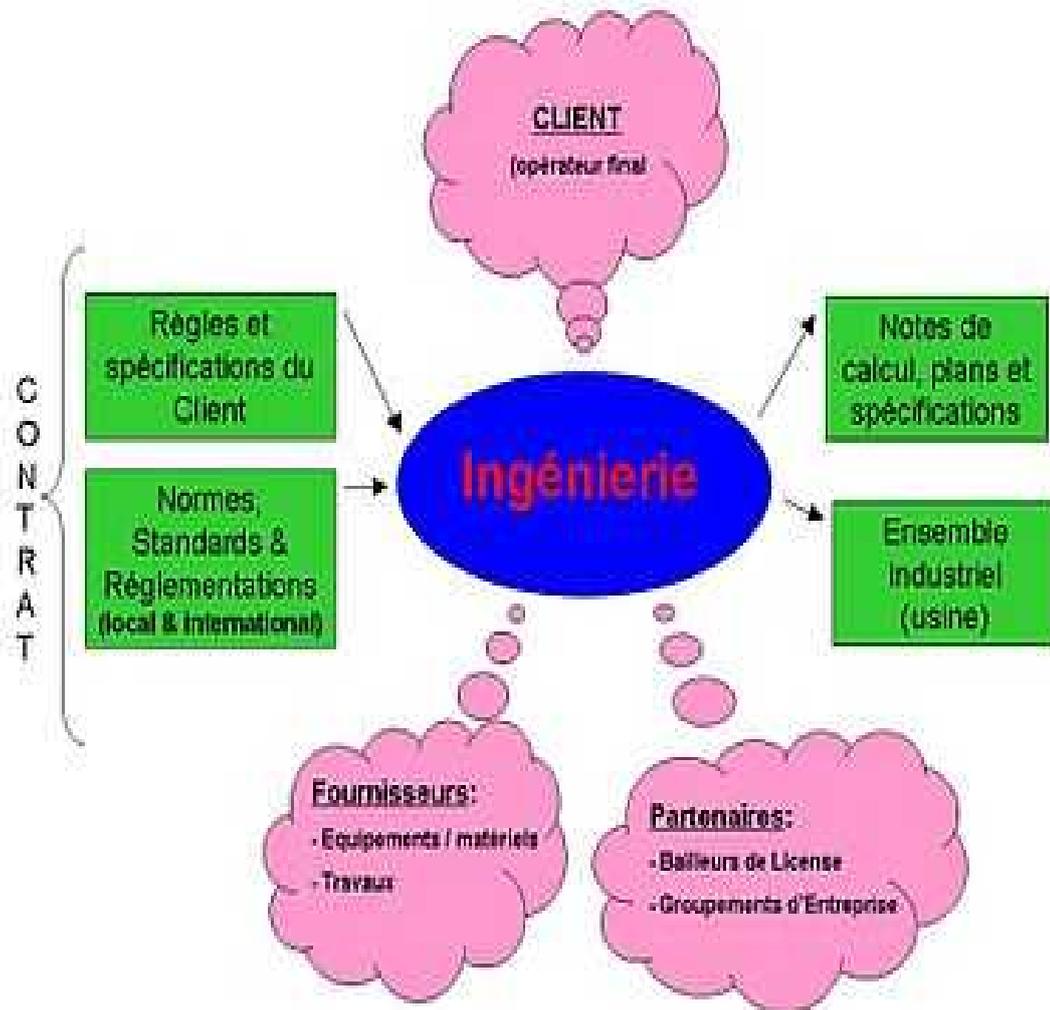


Figure 1.1 : Le fondement de l'ingénierie des besoins

Se centrer sur le POURQUOI doit permettre de découvrir les besoins correspondants à la mission et aux objectifs de l'organisation. Si l'on veut éviter de développer des systèmes techniquement parfaits mais inutilisés parce qu'inadaptés aux besoins réels de leurs utilisateurs, il faut se donner les moyens de comprendre à quoi le système va servir dans son contexte organisationnel. C'est la mission de l'IB. Mais ce n'est pas la pratique courante comme nous tentons de le montrer dans la section suivante. De la nécessité de revisiter les approches traditionnelles

Dans les méthodes héritées des années 80 telles que MERISE, l'ingénierie des besoins est partie intégrante de l'étape 'd'analyse' aboutissant à la construction d'une représentation abstraite des données, des traitements et des interfaces en suivant une approche de modélisation conceptuelle. L'objectif principal est de décrire ce que le système doit faire, c'est à dire ses fonctionnalités, dans un schéma conceptuel. Comme le montre la Figure 1.2, l'IB fondée sur la modélisation conceptuelle est centrée sur la question QUOI à laquelle on répond par le cycle <acquisition, abstraction, validation>. La tâche d'acquisition de connaissance de domaine et des besoins s'appuie sur un cahier des charges et des interviews. Elle permet d'abstraire de la connaissance acquise, la spécification des fonctionnalités attendues du système. Le schéma conceptuel résultant sert de support à la validation des besoins par des techniques telles que le maquetage et le prototypage.

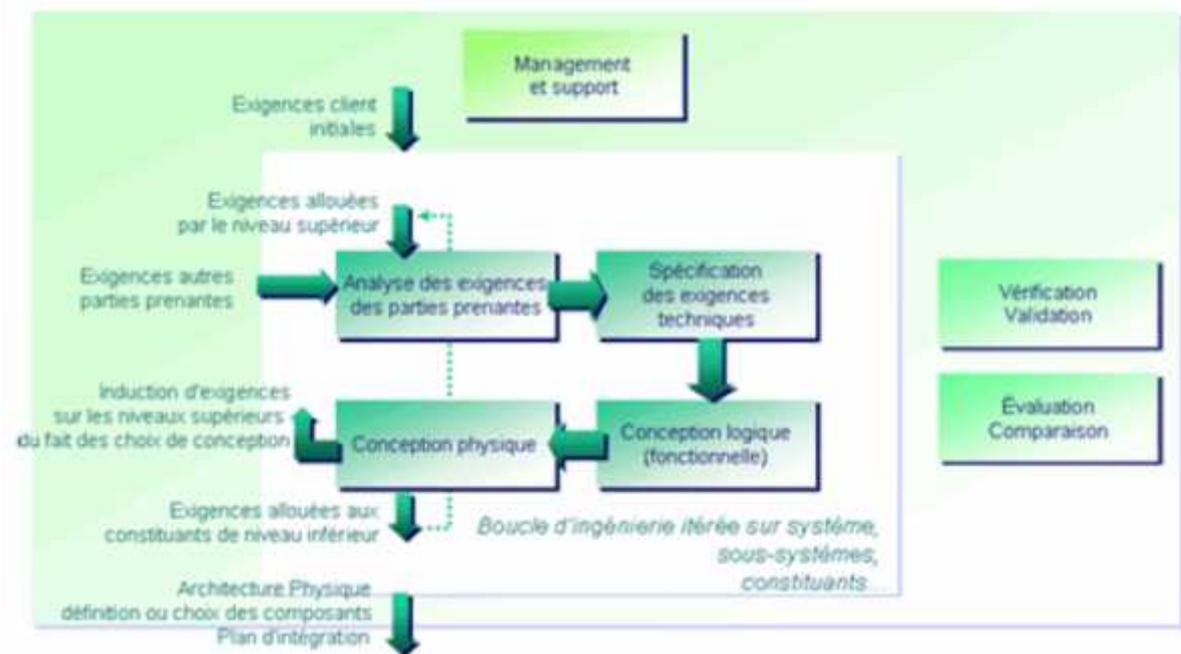


Figure 1.2 : Le cycle <acquisition, abstraction, validation>

L'IB fondée sur la modélisation conceptuelle n'est pas réellement conforme à la définition donnée. On peut comprendre qu'un centrage exclusif sur le QUOI inhibe les vraies questions du POURQUOI et aboutisse à la production de la spécification conceptuelle d'un système inadapté aux réelles attentes de ses usagers. Par ailleurs, le schéma conceptuel est l'expression d'une solution fonctionnelle et non des besoins à l'égard du système. L'absence d'une expression des besoins eux-mêmes ne peut que rendre difficile la validation par les

parties concernées. En outre la pratique du cycle précédent repose sur des hypothèses (le plus souvent implicites) qui ne semblent plus du tout valides aujourd'hui :

Les fonctionnalités d'un système sont stables, elles n'évoluent que peu dans le temps. En conséquence, le schéma conceptuel d'un système est lui aussi stable.

Les besoins relatifs à un système sont donnés au départ. Les utilisateurs expriment leurs besoins dans un cahier des charges, le problème est donc de construire le système qui répond à ce cahier des charges ; ce sont les analystes (la maîtrise d'œuvre) qui sont en charge de ce travail.

La validation des besoins peut se faire en référence aux fonctionnalités du système. En d'autres termes, le schéma conceptuel est le support privilégié pour communiquer, négocier et aboutir au consensus nécessaire entre les différentes parties impliquées dans le développement du système.

Si dans les années 80/90, ces hypothèses étaient valides, elles ne le sont plus aujourd'hui. En réponse aux pressions économiques et à l'émergence constante de nouvelles technologies, les organisations changent plus rapidement que par le passé. En conséquence, ce que les utilisateurs attendent du système évolue bien plus rapidement qu'auparavant. Leurs besoins ne sont donc pas stables [24]. On sait même que les besoins évoluent au cours du projet [25] et qu'il est donc nécessaire de se doter de moyens permettant d'établir un lien conceptuel entre les buts et objectifs de l'organisation (réponse au POURQUOI), les besoins qui en résultent (réponse au QUOI) et les spécifications fonctionnelles du système qui les supportent.

On sait que le rôle central de l'analyste système a été reconsidéré et que la maîtrise d'ouvrage vient contrebalancer le rôle de la maîtrise d'œuvre. En fait, il est clair aujourd'hui que l'ingénierie des besoins requiert la participation d'un grand nombre d'acteurs de l'organisation, chacun apportant sa vision sur ce que le système devrait faire. On distingue par exemple, les utilisateurs finaux du système – ceux qui utiliseront le système pour mener à bien leurs activités au sein de l'organisation –, les responsables de l'organisation qui ont décidé de la mise en place du système, les personnes responsables de la mise en place et de la maintenance du système, etc. ; en fait tous ceux pour qui le développement du système constitue un enjeu (les stakeholders » dans la terminologie anglo-saxonne).

Enfin, la validation des besoins sur la base du schéma conceptuel n'est pas satisfaisante. L'expérience l'a montré. On peut accepter l'idée que cette validation soit bien plus efficace si elle est basée sur l'expression des besoins eux-mêmes et non sur la spécification abstraite et difficile à comprendre des spécifications fonctionnelles du système qui en résultent. Ceci justifie l'introduction d'un document des besoins dans lequel les besoins spécifiés peuvent être discutés, négociés et validés [8].

1.7 Conclusion

L'ingénierie des besoins et des exigences est une phase décisive de tout projet de développement logiciel. Son résultat est en effet le cahier des charges du ou des applicatifs attendus, document utilisé tout au long du cycle de vie du projet d'informatisation. En plus de leur intérêt économique, car la correction a posteriori des erreurs liées au non-respect des exigences est coûteuse une fois le système livré, les exigences ont un intérêt légal, technique, de communication entre la MOA et la MOE et de gestion de projet.

Rapport-Gratuit.com

CHAPITRE II

DONNEES SEMI STRUCTUREES

2.1 Introduction

- Les données semi-structurées sont partout. Elles sont très importantes sur le Web.
- Le développement de XML accentue encore cet état de fait.

Données structurées :

- Les données contiennent intrinsèquement leurs propres significations et leur structure
- Leur structure est importante et évolutive.
- La structure est descriptive et non prescriptive : les violations sont autorisées.
- Le typage des données n'est pas strict : des objets différents peuvent avoir un attribut de même nom mais de types différents.

XML est un langage d'expression pour les données semi-structurées. Il est composé :

- D'objets complexes (ensemble d'objets complexes et atomiques)
- D'objets atomiques (chaîne, nombre, ...)
- Les données semi-structurées sont partout. Elles sont très importantes sur le Web.
- Le développement de XML accentue encore cet état de fait.

Données structurées :

- Les données contiennent intrinsèquement leurs propres significations et leur structure
- Leur structure est importante et évolutive.
- La structure est descriptive et non prescriptive : les violations sont autorisées.
- Le typage des données n'est pas strict : des objets différents peuvent avoir un attribut de même nom mais de types différents.

XML est un langage d'expression pour les données semi-structurées. Il est composé :

- D'objets complexes (ensemble d'objets complexes et atomiques)
- D'objets atomiques (chaîne, nombre, ...)
- D'attributs

2.2 Qu'est-ce que le XML ?

2.2.1 Définition

- XML est l'acronyme de "Extensible Markup Language", expression signifiant "langage de balisage extensible".
- Il s'agit d'un langage de création de fichiers de données basé sur le SGML et ayant la particularité de permettre la création et la définition de balises à la demande (d'où l'origine de l'expression "extensible"). Comme HTML, XML utilise les chevrons ouvrants (" $<$ ") et

fermants (">") pour encadrer les balises qui composent sa structure et des attributs pour l'affectation de valeurs aux paramètres qui caractérisent les balises.

- XML a été conçu pour permettre les transferts d'informations d'une plateforme à une autre (deux applications non interfacées, par exemple). Il évite ainsi les fastidieux et coûteux développements d'interfaces [9].

2.2.2 Un exemple de document XML

```
<person> <id=1, age=55>
  <name>Peter</name>
  <address>4711 Fruitdale Ave.</address>
  <child>
    <person> <id=3, age=22>
      <name>John</name>
      <address>5361 Columbia Ave.</address>
      <hobby>swimming</hobby>
      <hobby>cycling</hobby>
    </person>
  </child>
  <child>
    <person> <id=4, age=7>
      <name>David</name>
      <address>4711 Fruitdale Ave.</address>
    </person>
  </child>
</person>
<person> <id=2, age=38, child=4>
  <name>Mary</name>
  <address>4711 Fruitdale Ave.</address>
  <hobby>painting</hobby>
</person>
```

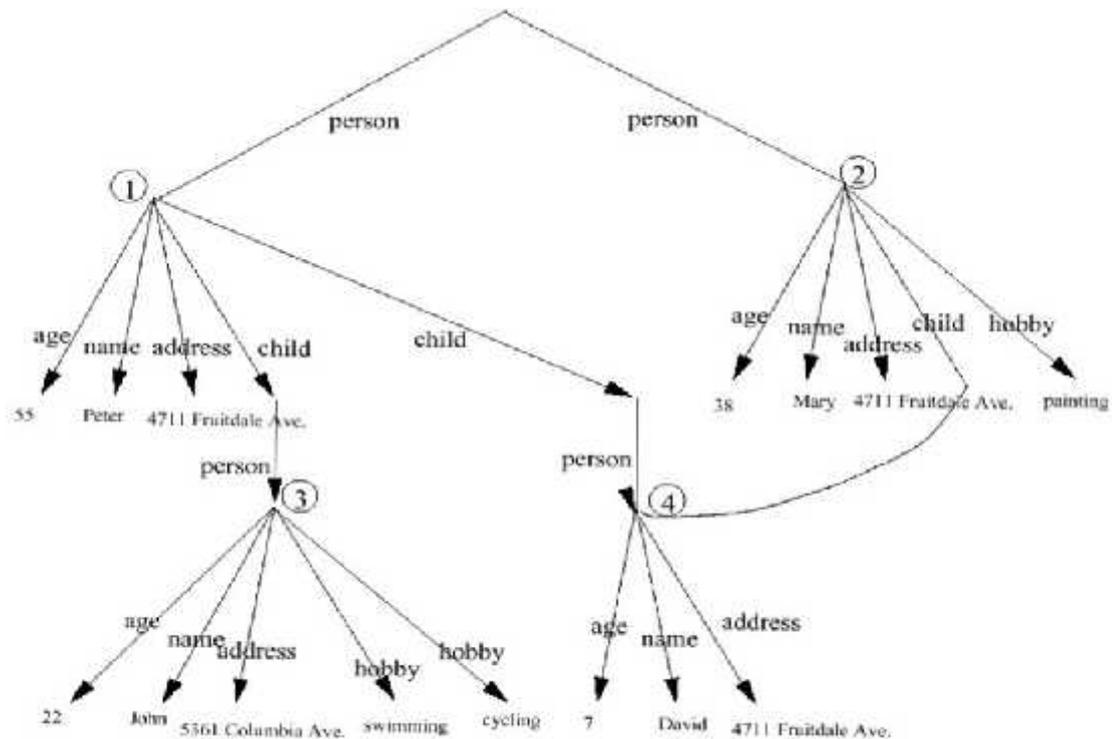


Figure 2.1 : Arborecence document xml

2.2.3 Les différentes technologies XML

2.2.3.1 DTD XML

XML permet d'utiliser un fichier afin de vérifier qu'un document XML est conforme à une syntaxe donnée. La norme XML définit ainsi une définition de document type appelée DTD (Document Type Définition), c'est-à-dire une grammaire permettant de vérifier la conformité du document XML. La norme XML n'impose pas l'utilisation d'une DTD pour un document XML, mais elle impose par contre le respect exact des règles de base de la norme XML.

Ainsi on parlera de :

- Document valide pour un document XML comportant une DTD
- Document bien formé pour un document XML ne comportant pas de DTD mais répondant aux règles de base du XML

Une DTD peut être définie de 2 façons :

- Sous forme interne, c'est-à-dire en incluant la grammaire au sein même du document

- Sous forme externe, soit en appelant un fichier contenant la grammaire à partir d'un fichier local ou bien en y accédant par son URL [10]

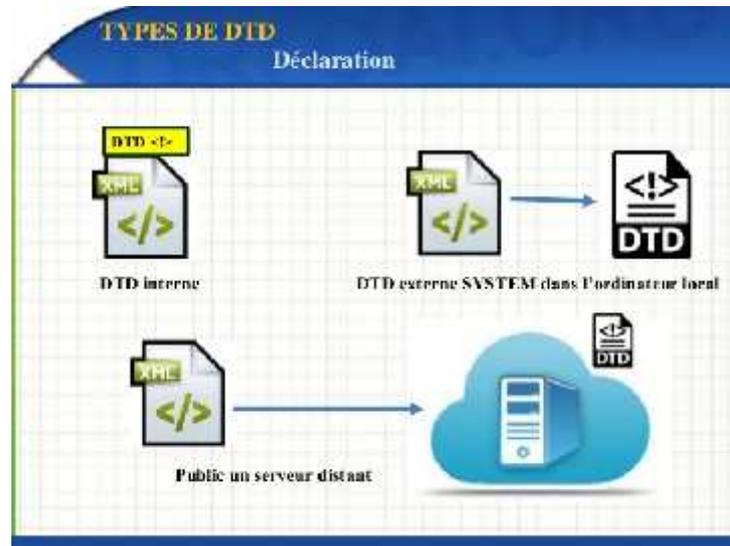


Figure2.2 : DTD XML

2.2.3.2 SCHEMA XML

XML Schéma publié comme recommandation par le W3C en mai 2001 est un langage de description de format de document XML permettant de définir la structure et le type de contenu d'un document XML. Cette définition permet notamment de vérifier la validité de ce document.

Il est possible de décrire une organisation de vocabulaires d'origines différentes, par l'usage des espaces de noms. Il est possible de combiner les schémas eux-mêmes, et d'exprimer une combinaison pour le document contenu, comme quelqu'un qui parlerait de géographie et de sociologie dans un même texte.

Il est également possible, après une validation, de savoir avec quelle règle une information particulière a été testée : il s'agit du jeu de validation post-schema, ou PSVI (post-schema-validation infoset).

Une définition se compose d'un ou plusieurs documents XML, usuellement nommée (XML Schema Définition en anglais, ou fichier XSD).

Une instance d'un XML Schéma est un peu l'équivalent d'une définition de type de document (DTD). XML Schéma amène cependant plusieurs différences avec les DTD : il permet par exemple de définir des domaines de validité pour la valeur d'un champ, alors que cela n'est pas possible dans une DTD ; en revanche, il ne permet pas de définir des entités ; XML Schema est lui-même un document XML, alors que les DTD sont des documents SGML.

Ce langage de description de contenu de documents XML est lui-même défini par un schéma, dont les balises de définition s'auto-définissent (c'est un exemple de définition récursive)1.

La recommandation du W3C 1.0 se compose d'un document de présentation (non normatif), d'un document précisant comment définir la structure, et d'un document précisant comment définir les données. La dernière édition, de version 1.0, de cette recommandation, date de 2004. Le W3C travaille actuellement sur la version 1.1 dont l'objectif est de définir les notions de version de schéma, ainsi que des contraintes selon la présence de telle ou telle valeur [11].

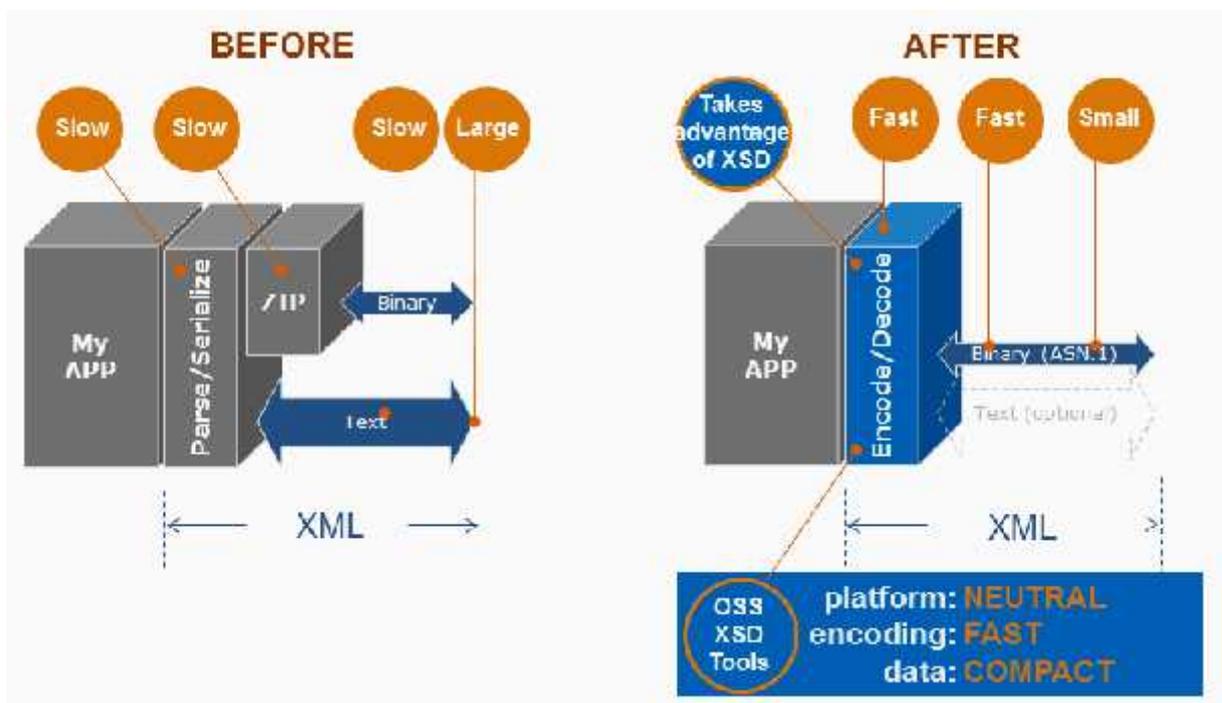


Figure2.3 : XSD ADVANTAGE

2.2.3.3 XPATH XML

XPath est un langage permettant de sélectionner des parties d'un document XML. Il est utilisé dans de nombreux dialectes XML.

Le langage XPath n'est pas un langage autonome. C'est un langage d'expressions utilisé au sein d'un autre langage hôte. Il ressemble, dans cet aspect, aux expressions rationnelles, appelées aussi expressions régulières qui est abrégé en regex telles qu'elles sont utilisées dans les langages de script tels que Perl ou Python.

La syntaxe de XPath n'est pas une syntaxe XML car les expressions XPath apparaissent en général comme valeurs d'attributs de documents XML. C'est en particulier le cas pour les schémas, les schematrons et XSLT.

XPath était au départ un langage permettant essentiellement de décrire des ensembles de nœuds dans un document XML. La version 1.0 de XPath comprenait quelques fonctions pour la manipulation de nombres et de chaînes de caractères. L'objectif était alors de pouvoir comparer les contenus de nœuds. La version 2.0 de XPath a considérablement enrichi le langage. Il est devenu un langage beaucoup plus complet capable, par exemple, de manipuler des listes de nœuds et de valeurs atomiques.

XPath est uniquement un langage d'expressions dont l'évaluation donne des valeurs sans effet de bord. Il n'est pas possible dans XPath de mémoriser un résultat. Il n'existe pas de variables propres à XPath mais une expression XPath peut référencer des variables du langage hôte. Les valeurs de ces variables sont alors utilisées pour évaluer l'expression. L'affectation de valeurs à ces variables se fait uniquement au niveau du langage hôte.

Le cœur de XPath est formé des expressions de chemins permettant de décrire des ensembles de nœuds d'un document XML. Ces expressions ressemblent aux chemins Unix pour nommer des fichiers dans une arborescence [12].



Figure 2.4 : XML AND XPATH

2.2.3.4 XSL XML

Le langage XSL (pour XML Stylesheet Language) a été conçu pour transformer des documents XML en d'autres formats comme PDF ou des pages HTML. Au cours de son développement, le projet s'est avéré plus complexe que prévu et il a été scindé en deux unités distinctes XSLT et XSL-FO. Le langage XSLT (pour XML Stylesheet Language Transformation) est un langage de transformation de documents XML. Le langage XSL-FO (pour XML Stylesheet Language - Formatting Objects) est un langage de mise en page de document. Le processus de transformation d'un document XML en un document imprimable, au format PDF par exemple, est donc découpé en deux phases. Dans la première phase, le document XML est transformé en un document XSL-FO à l'aide de feuilles de style XSLT. Dans la seconde phase, le document FO obtenu à la première phase est converti par un processeur FO en un document imprimable [13].

Même si le langage XSLT puise son origine dans la transformation de documents XML en document XSL-FO, il est adapté à la transformation d'un document de n'importe quel dialecte XML dans un document de n'importe quel autre dialecte XML. Il est souvent utilisé pour produire des documents XSL-FO ou XHTML mais il peut aussi produire des documents SVG. Le langage XSLT est aussi souvent utilisé pour réaliser des transformations

simples sur des documents. Il s'agit, par exemple, de supprimer certains éléments, de remplacer un attribut par un élément ou de déplacer un élément [14].

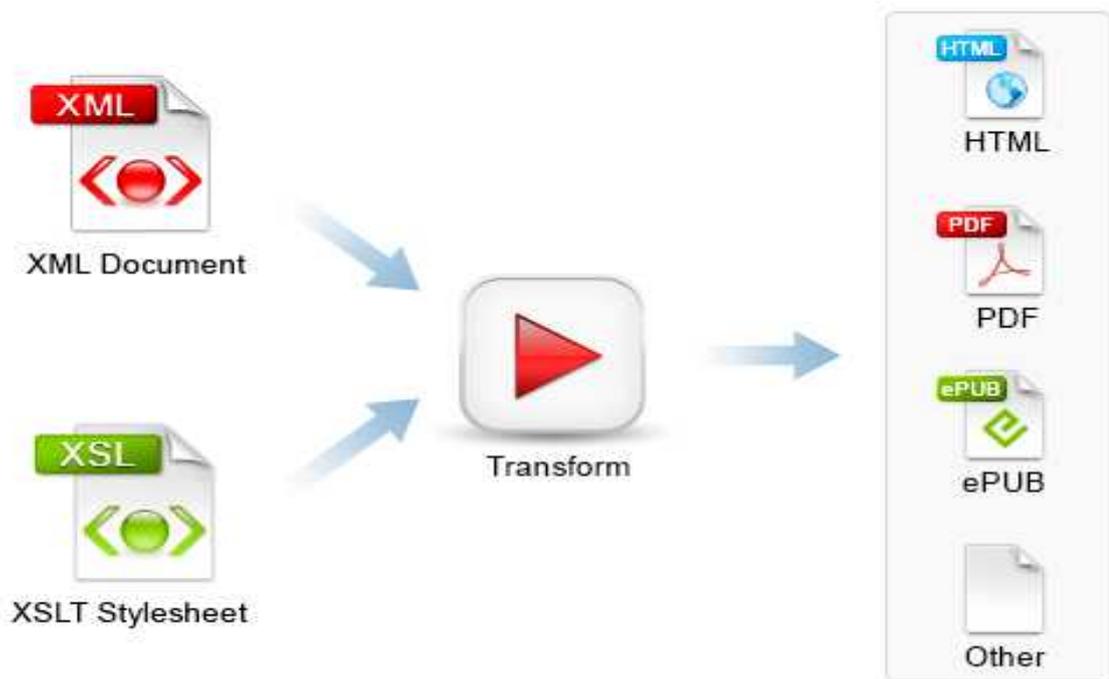


Figure 2.5 : Principe de XSLT XML

2.2.3.5 XQUERY XML

XQuery est un langage de requête informatique permettant non seulement d'extraire des informations d'un document XML, ou d'une collection de documents XML, mais également d'effectuer des calculs complexes à partir des informations extraites et de reconstruire de nouveaux documents ou fragments XML.

XQuery est une spécification du W3C dont la version 1.0 finale date de janvier 2007, et dont l'élaboration a demandé près de huit années. XQuery a été développé conjointement avec XSLT 2, une révision majeure du langage de transformation XML XSLT, avec lequel il partage le sous-ensemble XPath 2.0 (en).

XQuery joue par rapport aux données XML un rôle similaire à celui du langage SQL vis-à-vis des données relationnelles, et l'on peut trouver des analogies entre ces deux langages [15].

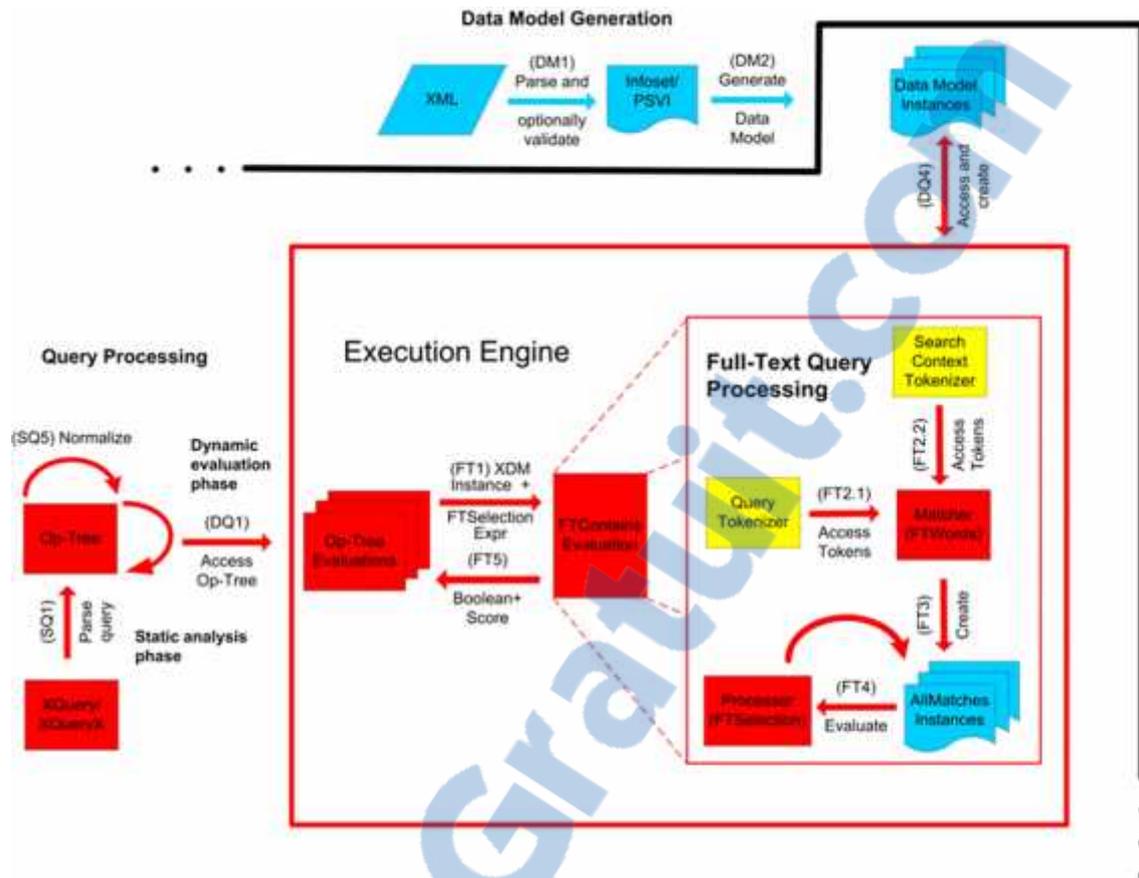


Figure 2.6 : Principe de XQUERY XML

2.2.4 Les avantages de XML

Voici les principaux atouts de XML :

- La lisibilité : aucune connaissance ne doit théoriquement être nécessaire pour comprendre un contenu d'un document XML
- Auto descriptif et extensible
- Une structure arborescente : permettant de modéliser la majorité des problèmes informatiques
- Universalité et portabilité : les différents jeux de caractères sont pris en compte
- Déployable : il peut être facilement distribué par n'importe quels protocoles à même de transporter du texte, comme HTTP
- Intégrabilité : un document XML est utilisable par toute application pourvue d'un parser (c'est-à-dire un logiciel permettant d'analyser un code XML)
- Extensibilité : un document XML doit pouvoir être utilisable dans tous les domaines d'applications

Ainsi, XML est particulièrement adapté à l'échange de données et de documents.

L'intérêt de disposer d'un format commun d'échange d'information dépend du contexte professionnel dans lequel les utilisateurs interviennent. C'est pourquoi, de nombreux formats de données issus de XML apparaissent (il en existe plus d'une centaine) :

- OFX : Open Financial eXchange pour les échanges d'informations dans le monde financier
- MathML : Mathematical Markup Language permet de représenter des formules mathématique
- CML : Chemical Markup Language permet de décrire des composés chimiques
- SMIL : Synchronized Multimedia Integration Language permet de créer des présentations multimédia en synchronisant diverses sources : audio, vidéo, texte,...

2.2.5 Conclusion

On peut affirmer que le langage XML est la nouvelle base du document numérique, grâce à ses nombreux avantages. Il permet de normaliser les échanges de documents grâce à un balisage sémantique indépendant des plates-formes et des langages de programmation. Actuellement, de nombreuses sociétés développent et utilisent des applications XML dans des domaines très variés allant de la chimie aux mathématiques (MathML, SVG, SMIL, CML, ...).

En tant que format d'échange universel, XML constitue également une alternative économique et fiable aux bases de données relationnelle de taille moyenne. Il existe pour cela des langages de requêtes (par exemple XQuery) permettant d'accéder aux composants des documents XML de la même manière que dans une base de données.

Enfin, la force de XML, qui n'est pas la plus sollicitée hors de l'informatique du document, est dans la validation, ce qui contribue à standardiser des types de documents à grande échelle, ne serait-ce que le Web et HTML.

CHAPITRE III

CONTRIBUTION

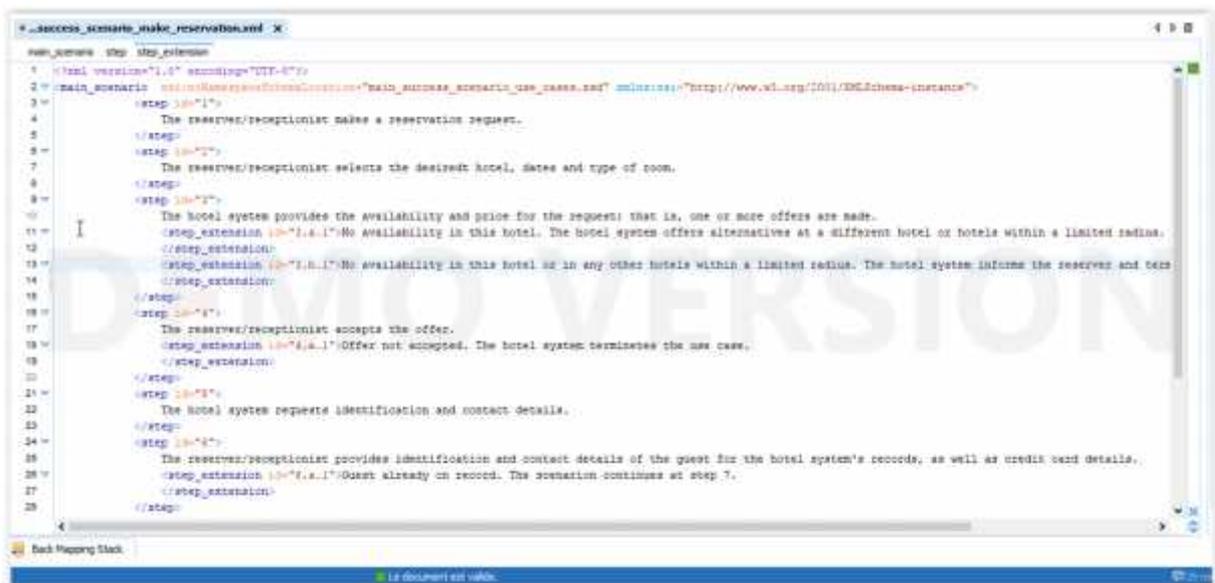
3.1 Introduction

Ce chapitre est consacré à la présentation des différents outils qui ont permis la réalisation de l'application et l'ensemble des fonctionnalités qu'offre cette dernière sous forme de prises d'écran avec des descriptions pour bien comprendre chaque étape de la réalisation et de la mise en place du document électronique d'exigences.

3.2 Structuration logique

En partant du cas d'étude 'système de réservation d'hôtel', nous avons considéré au début du projet un seul document XML qui regroupe toutes les informations du cas d'étude. Cette solution s'est avérée vite inefficace vu le volume important du document obtenu. Ce qui rend son exploitation très complexe. Du coup, nous avons entamé une autre approche en considérant un document maître et des documents esclaves. Nous avons utilisé pour la modélisation des deux catégories de documents une DTD pour le document maître et des schémas XML pour les documents esclaves.

Nous commençons par insérer une capture d'écran, du document XML Scenario Principal du cas d'utilisation « faire une réservation », sachant qu'il existe trois autres cas d'utilisation mais qui lui sont similaires.



```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <main_scenario xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
3   xmlns="http://www.w3.org/2001/XMLSchema-instance"
4   xsi:schemaLocation="http://www.w3.org/2001/XMLSchema-instance http://www.w3.org/2001/XMLSchema-instance">
5   <step id="1">
6     The reserves/receptionist makes a reservation request.
7   </step>
8   <step id="2">
9     The reserves/receptionist selects the desired hotel, dates and type of room.
10  </step>
11  <step id="3">
12    The hotel system provides the availability and price for the request; that is, one or more offers are made.
13    <step_extension id="3.1.1">No availability in this hotel. The hotel system offers alternatives at a different hotel or hotels within a limited radius.
14    </step_extension>
15    <step_extension id="3.1.2">No availability in this hotel or in any other hotels within a limited radius. The hotel system informs the reserves and tells
16    </step_extension>
17  </step>
18  <step id="4">
19    The reserves/receptionist accepts the offer.
20    <step_extension id="4.1.1">Offer not accepted. The hotel system terminates the use case.
21    </step_extension>
22  </step>
23  <step id="5">
24    The hotel system requests identification and contact details.
25  </step>
26  <step id="6">
27    The reserves/receptionist provides identification and contact details of the guest for the hotel system's records, as well as credit card details.
28    <step_extension id="6.1.1">Guest already on record. The scenario continues at step 7.
29    </step_extension>
30  </step>

```

Figure 3. 1 : Document XML du scénario principal du cas d'utilisation « faire une réservation »

Passons maintenant, au schéma XML qui concerne tous les cas d'utilisation qui se trouvent dans le cas d'étude.

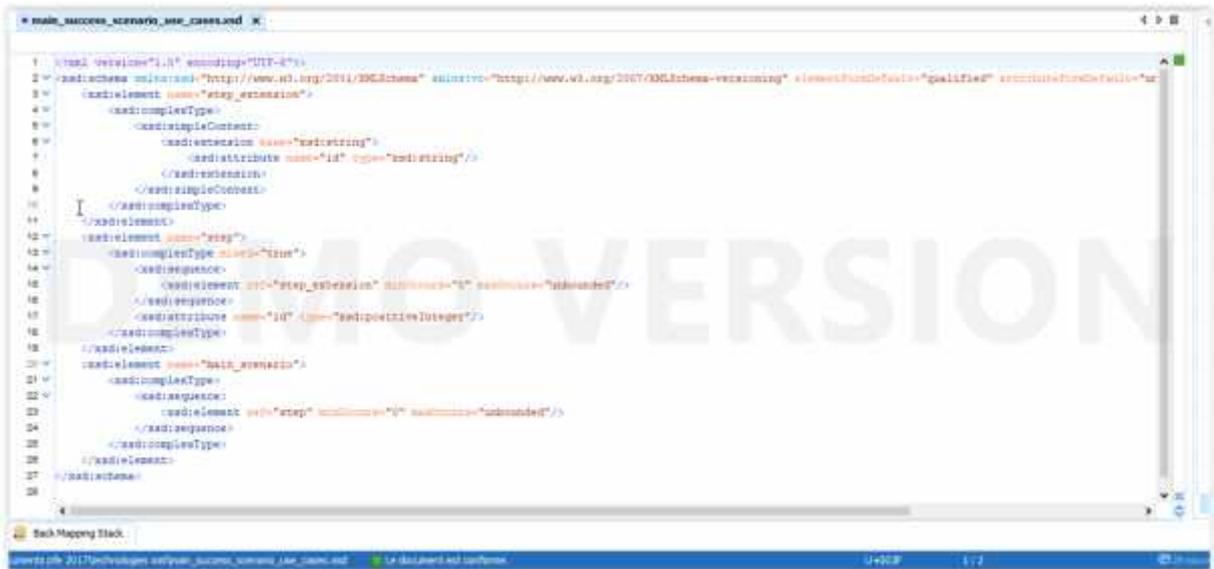


Figure 3. 2 : Document XSD du Scénario principal des cas d'utilisations

Enchainons avec les documents XML Exigence fonctionnelle et Exigence Non fonctionnelle du cas d'utilisation « faire une réservation ». Il faut savoir qu'il y a des cas d'utilisation qui ne possèdent pas ces deux documents.



Figure 3. 3 : Document XML des exigences fonctionnelles du cas d'utilisation «faire une réservation»

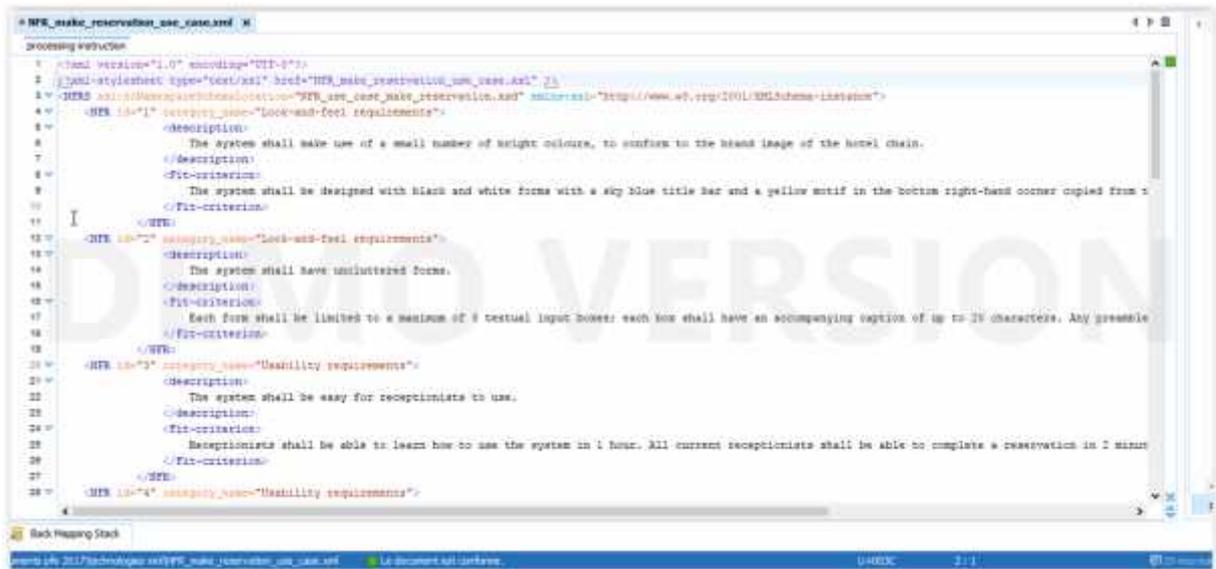


Figure 3. 4 : Document XML des exigences Non fonctionnelles du cas d’utilisation «faire une réservation»

Intéressons-nous maintenant à la DTD qui a permis l’utilisation de la notion de modularité ainsi que le schéma XML.

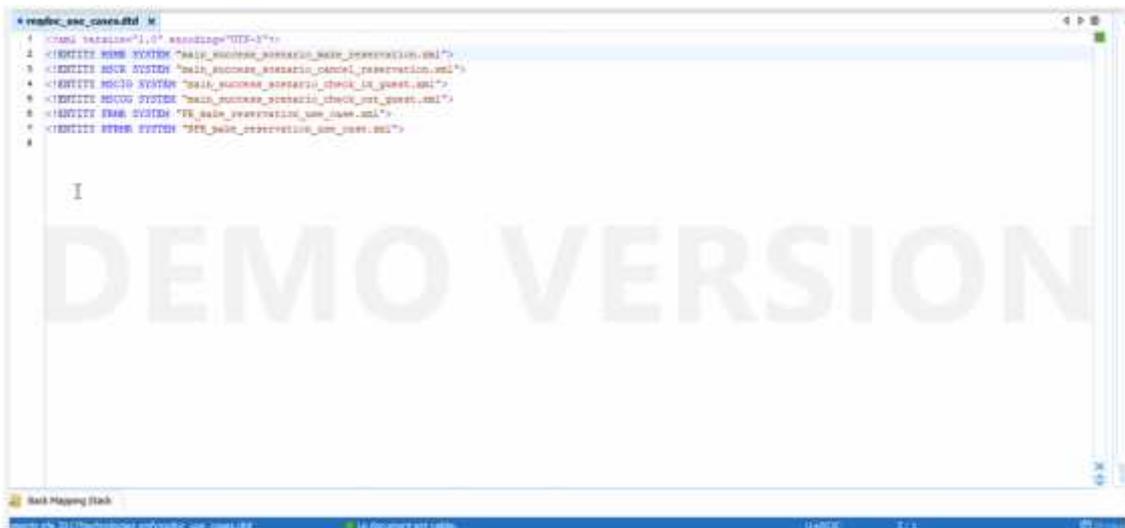


Figure 3. 5 : Document DTD du Document d’exigence des cas d’utilisations



Figure 3. 6 : Document XSD du document d'exigences des cas d'utilisations

Terminons enfin avec le document XML du cas d'utilisation «faire une réservation», ainsi que le document maître qui regroupe tous les cas d'utilisation.

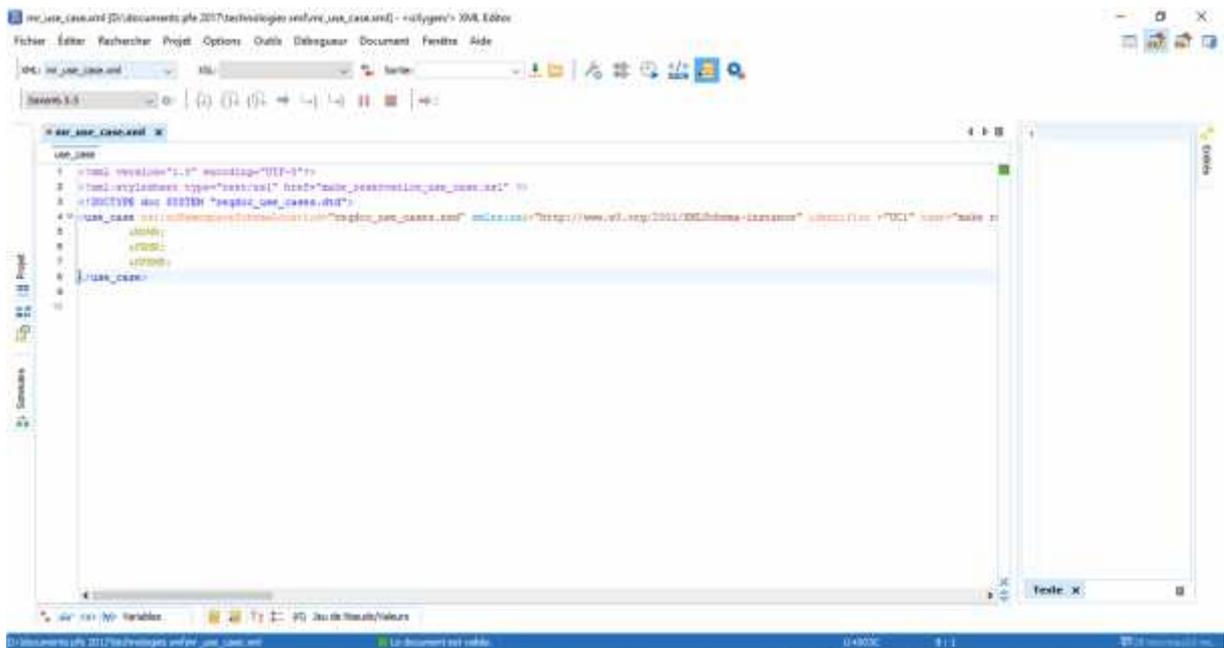


Figure 3. 7 : Document XML du cas d'utilisation «faire une réservation»

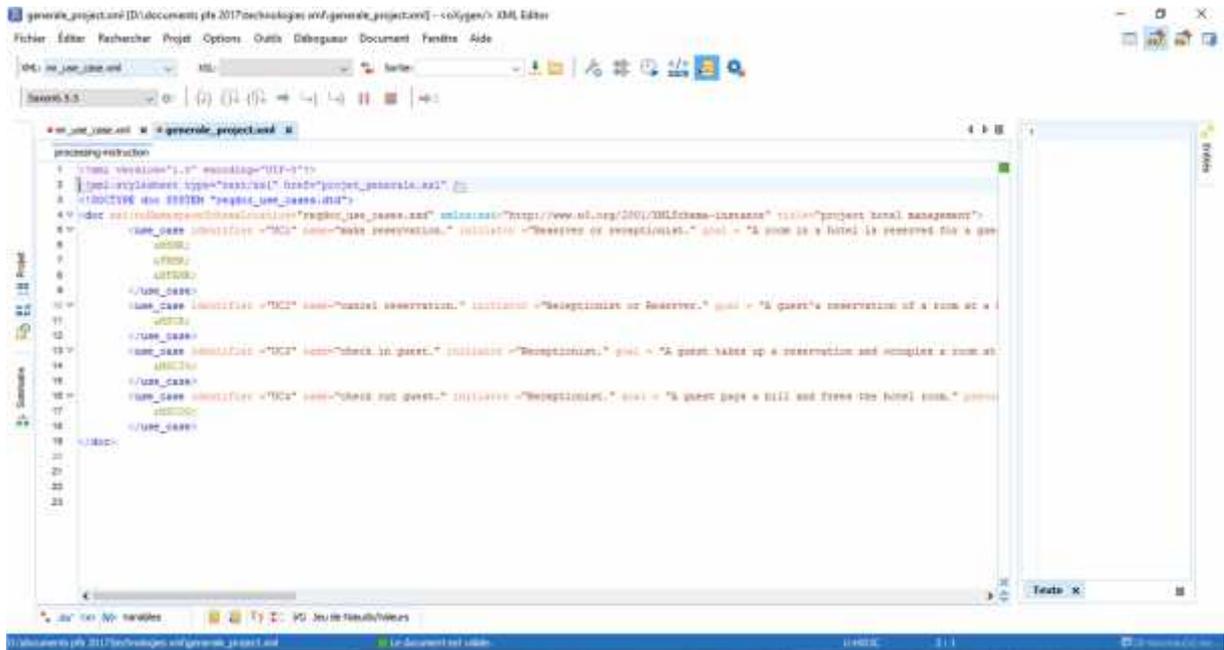


Figure 3. 8 : Document XML du projet général regroupant tous les cas d'utilisations

3.3 Visualisation

3.3.1 MultiMedia

Multimédia est un terme qui vient de la langue anglaise et qui se réfère à ce qui fait appel à plusieurs médias en même temps dans la transmission de l'information. Une présentation multimédia, par conséquent, peut comprendre des photos, des vidéos, des sons et des textes.

Le concept s'applique à des objets et à des systèmes qui font appel à de nombreux moyens physiques et / ou numériques pour communiquer leurs contenus. Le terme est également utilisé en référence aux propres moyens de stocker et de diffuser des contenus avec de telles caractéristiques.

D'un point de vue informatique, un produit multimédia fait appel à deux techniques voisines : l'hypertexte et l'hypermédia [16].

3.3.2 Hypertexte

Le terme « Hypertexte » fut proposé en 1965 par Ted Nelson pour décrire des documents traduisant la structure non linéaire des idées.

S'appliquant à une Bibliothèque de fichiers, la technique Hypertexte consiste à chaîner ensemble ces fichiers par un tissu de relations non séquentiel. Ces liens permettent ainsi à l'utilisateur de naviguer parmi des sujets variés sans tenir Compte de l'ordre dans lequel ils sont rangés [17].

3.3.3 Hypermédia

La technique Hypermédia, similaire au mode hypertexte, s'applique aux documents composés de textes, de sons, d'images et de toutes combinaisons de ces derniers. De la même manière que la technique hypertexte, l'hypermédia présente toutes ces données sous forme de stockage-recherche d'informations [18].

3.3.4 Texte dynamique

Un texte dynamique est un texte placé sur une mise en page de la carte qui change en fonction des propriétés du projet, d'une fenêtre cartographique, d'une carte, etc. On peut ajouter du texte dynamique pour des éléments tels que les suivants :

- Un nom d'utilisateur.
- La date du dernier enregistrement du projet.
- Le chemin d'accès du projet.
- Les métadonnées d'une carte sur la page.

Lorsqu'on insère un texte dynamique, il affiche automatiquement la valeur actuelle de sa propriété respective. Lorsque cette propriété est mise à jour, le texte dynamique l'est aussi, automatiquement. Le texte dynamique fonctionne par le biais de l'utilisation de balises, telles que HTML. Cela permet de combiner du texte dynamique et du texte statique en un élément textuel unique et d'appliquer les options de mise en forme disponibles pour personnaliser l'affichage de texte résultant [19].

3.4 Interrogation des données

Le langage XQuery est un langage de requêtes déclaratif et modulaire faisant partie de la famille des langages fonctionnels. Il est aux données XML ce que SQL est aux données relationnelles. XQuery a été développé au sein du W3C (World Wide Web Consortium) pour interroger les données XML qui sont de plus en plus présentes sur le Web et dans les systèmes d'information. Il peut être utilisé, soit pour interroger des documents XML enregistrés dans de simples fichiers, soit pour interroger des données XML enregistrées dans des bases de données XML natives ou des bases de données relationnelles.

3.5 Autres outils

3.5.1 Sigil

Sigil est un éditeur open source de documents numériques au format EPUB. Développée par Strahinja Markovic en 2009 et maintenue par John Sember depuis 20112, c'est une application multiplate-forme distribuée sous la licence GNU GPL.

Cet éditeur « WYSIWYG » qui est l'acronyme de la locution anglaise « what you see is what you get », signifiant littéralement en français « ce que l'on voit est ce que l'on obtient » ou tel écran, tel écrit, offre deux modes de visualisation. Le mode simplifié permet de placer les éléments indépendamment du code source. Le mode évolué donne la possibilité de travailler directement avec le code du document numérique ; il s'agit alors d'écrire manuellement les balises HTML et XML3.

Ce logiciel permet de personnaliser l'apparence du document numérique en modifiant différents paramètres de mise en page : alignement du texte, numérotation des pages...

Ce logiciel offre un support pour l'Unicode ainsi que pour la création d'un index ou d'une table des matières.



Figure 3. 9 : Structure de Sigil

3.5.1.1 Table des matières avec Sigil

Une table des matières interactive est nécessaire pour aider les lecteurs à se retrouver facilement dans l'ebook (et répondre aux spécifications Epub). Sigil en génère automatiquement tous les aspects techniques normés.

3.5.1.2 L'index avec Sigil

La création d'un index se fait en 2 étapes : la première étape consiste à marquer les mots ou phrases qui seront indexés et la deuxième étape c'est la création à proprement parler de l'index.

Il existe deux méthodes pour indexer un document, le marquage manuel et le marquage par un fichier de concordance.

3.5.2 Calibre

Le logiciel Calibre est un puissant gestionnaire d'e-books. Il simplifie l'organisation des informations de la bibliothèque et intègre un convertisseur rapide et performant. La visionneuse incluse dans Calibre permet d'afficher l'ensemble des formats d'e-book disponibles. Il comprend un support complet pour la gestion de la table des matières, des signets, des CSS et autres fonctionnalités pratiques.

Le programme peut servir de pilote de périphérique modulaire permettant l'ajout de support pour différents dispositifs e-reader. Il se charge également de la mise à jour des métadonnées à partir des balises données dans la bibliothèque. Calibre prend en charge la majorité des formats de documents électroniques. Supportant la conversion par lot de fichiers, il procède à une vitesse assez surprenante. Des options de redimensionnement de la taille des polices et d'insertion d'une jaquette sont liées à cette fonctionnalité. L'utilisateur de Calibre peut librement accéder à sa collection d'e-books en ligne en se servant du serveur Web intégré. Il est autorisé à envoyer des documents et télécharger d'autres. La compatibilité du logiciel avec les Smartphones optimise le téléchargement des documents pour les appareils mobiles [20].

3.5.2.1 Gestion de bibliothèque

- Calibre est basé sur le concept de “document logique” : une entrée unique dans la bibliothèque peut correspondre à plusieurs versions d'un même e-book dans différents formats. Non seulement, il propose de nombreuses options de tri...Auteur
- Titre
- Editeur

- Langue
- Format
- Date d'ajout
- Date de publication
- Taille
- Note
- Série

...mais il supporte également des métadonnées telles que les étiquettes (tags) et commentaires (entrées de "texte long", dans lesquelles on peut insérer description, notes, critiques, etc.).

Les options de recherche sont nombreuses (options de tri + métadonnées). Une option de recherche avancée permet d'effectuer des requêtes très puissantes.

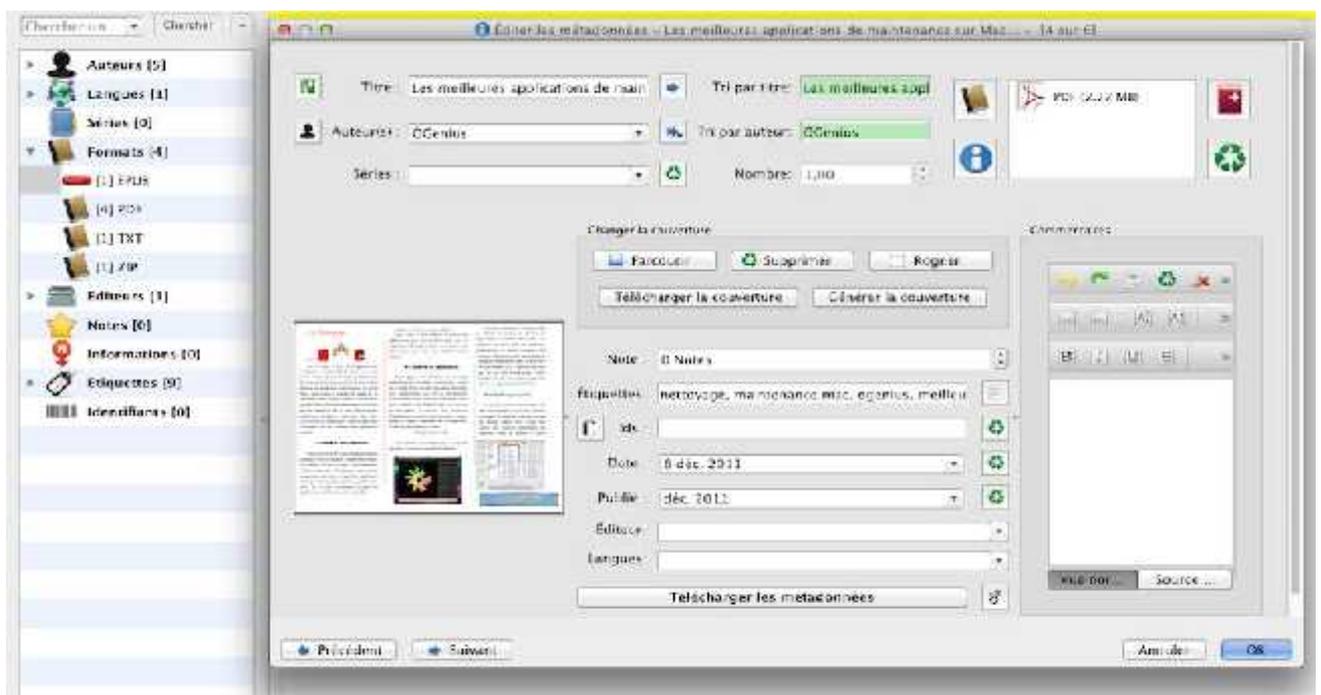


Figure 3. 10 : Gestion de la bibliothèque calibre

Nous pouvons également exporter des parties de notre bibliothèque sur notre disque dur, en utilisant une structure de dossier totalement personnalisable.

Enfin, Calibre peut rechercher sur internet les métadonnées d'un e-book, en se basant sur le titre, l'auteur ou les informations ISBN. Il peut télécharger automatiquement différents types de métadonnées, ainsi que les couvertures de vos documents. Calibre utilise un système

de métadonnées basé sur des plug-ins, ce qui facilitera l'ajout de types et sources de métadonnées [21].

3.5.2.2 Conversion d'e-books

Calibre peut convertir un très grand nombre de formats dans un très grand nombre de formats... Il supporte tous les principaux :

Source : Z, CBR, CBC, CHM, DJVU, EPUB, FB2, HTML, HTMLZ, LIT, LRF, MOBI, ODT, PDF, PRC, PDB, PML, RB, RTF, SNB, TCR, TXT, TXTZ.

Cible : EPUB, FB2, OEB, LIT, LRF, MOBI, HTMLZ, PDB, PML, RB, PDF, RTF, SNB, TCR, TXT, TXTZ.

Les meilleurs résultats de conversion sont obtenus avec les formats source suivants (du meilleur au ...moins bon) : LIT, MOBI, EPUB, FB2, HTML, PRC, RTF, PDB, TXT, PDF (un des formats les plus difficiles à convertir correctement...).

On note que cela ne permettra pas de contourner les problèmes de DRM...

Le moteur de conversion utilisé par Calibre dispose de puissantes fonctionnalités. Il peut redimensionner toutes les polices, afin de rendre le fichier de sortie plus lisible, détecter / créer la structure d'un document –chapitres, table des matières, insérer les métadonnées au début d'un document, etc [21].

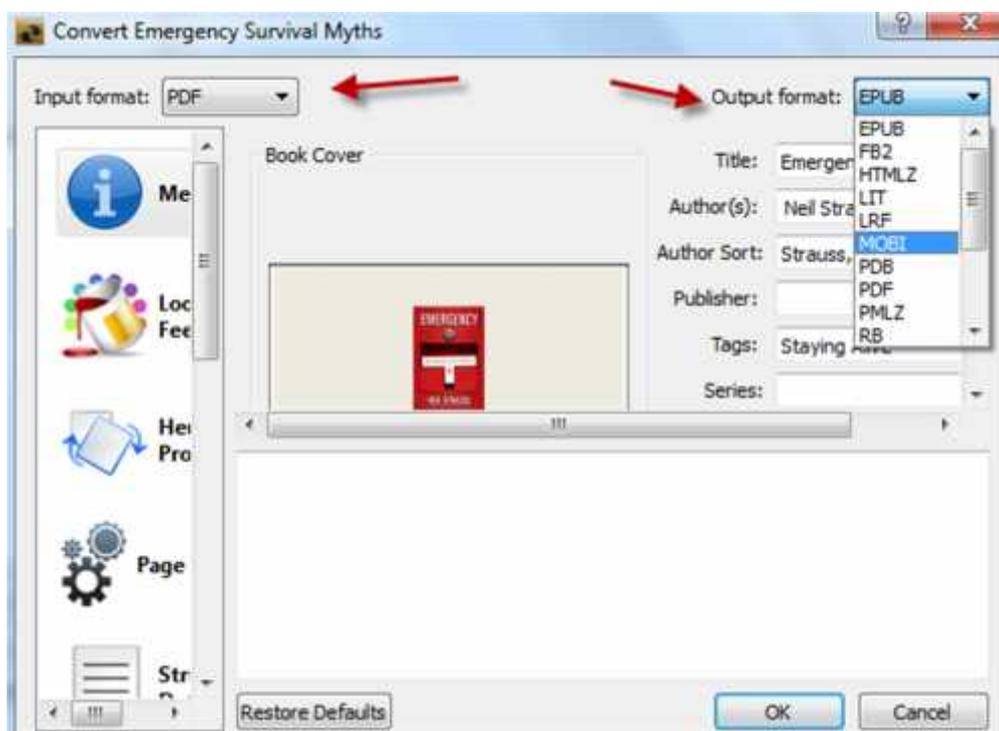


Figure 3.11 : Comment convertir PDF en EPUB, MOBI

3.5.2.3 Synchronisation avec de nombreux appareils

Calibre dispose d'un pilote modulaire, ce qui facilite l'ajout de nouveaux appareils.

Il en supporte déjà beaucoup, parmi lesquels : iPad / iPhone, SONY PRS, Barnes & Noble Nook, Cybook Gen 3/Opus, Amazon Kindle, Entourage Edge, Longshine ShineBook, Ectaco Jetbook, BeBook/BeBook Mini, Irex Illiad/DR1000, Foxit eSlick, PocketBook line, Italica, eClicto, Iriver Story, Airis dBook, Hanvon N515, Binatone Readme, Teclast K3 et clones, SpringDesign Alex, Kobo Reader, plusieurs mobiles Android.

Les métadonnées contenues dans l'appareil connecté sont synchronisées avec celles de la bibliothèque, et des collections sont créées à partir des tags étiquettes ("tags", en français...) que nous avons définis dans la bibliothèque. Si l'e-book est disponible dans plusieurs formats, Calibre choisit le plus adapté à l'appareil connecté. S'il ne trouve aucun format adapté, il convertit l'e-book avant de l'envoyer à l'appareil –en conservant bien sûr l'original [21].

3.5.2.4 Téléchargement & conversion de news

Calibre peut récupérer automatiquement les news de sites web ou de flux RSS, les formater sous forme d'e-books et les transférer sur le lecteur. Ces e-books reprennent le texte intégral de l'article, pas seulement le sommaire ! Parmi les sites d'actualités supportés, on peut citer :

- The New York Times
- The Wall Street Journal
- The Economist
- Time
- Newsweek
- The Guardian
- ESPN
- Et beaucoup, beaucoup d'autres...

Calibre a actuellement plus de 300 sources de news et ici encore, son architecture modulaire permet d'en ajouter très facilement et donc d'enrichir en permanence cette manne d'informations [21].

3.5.2.5 Visionneuse d'e-books intégrée

Calibre intègre sa propre visionneuse, qui supporte les principaux formats d'e-books. De qualité très correcte, elle offre table des matières, marque-pages, CSS, mode référence, impression, recherche, etc [21].

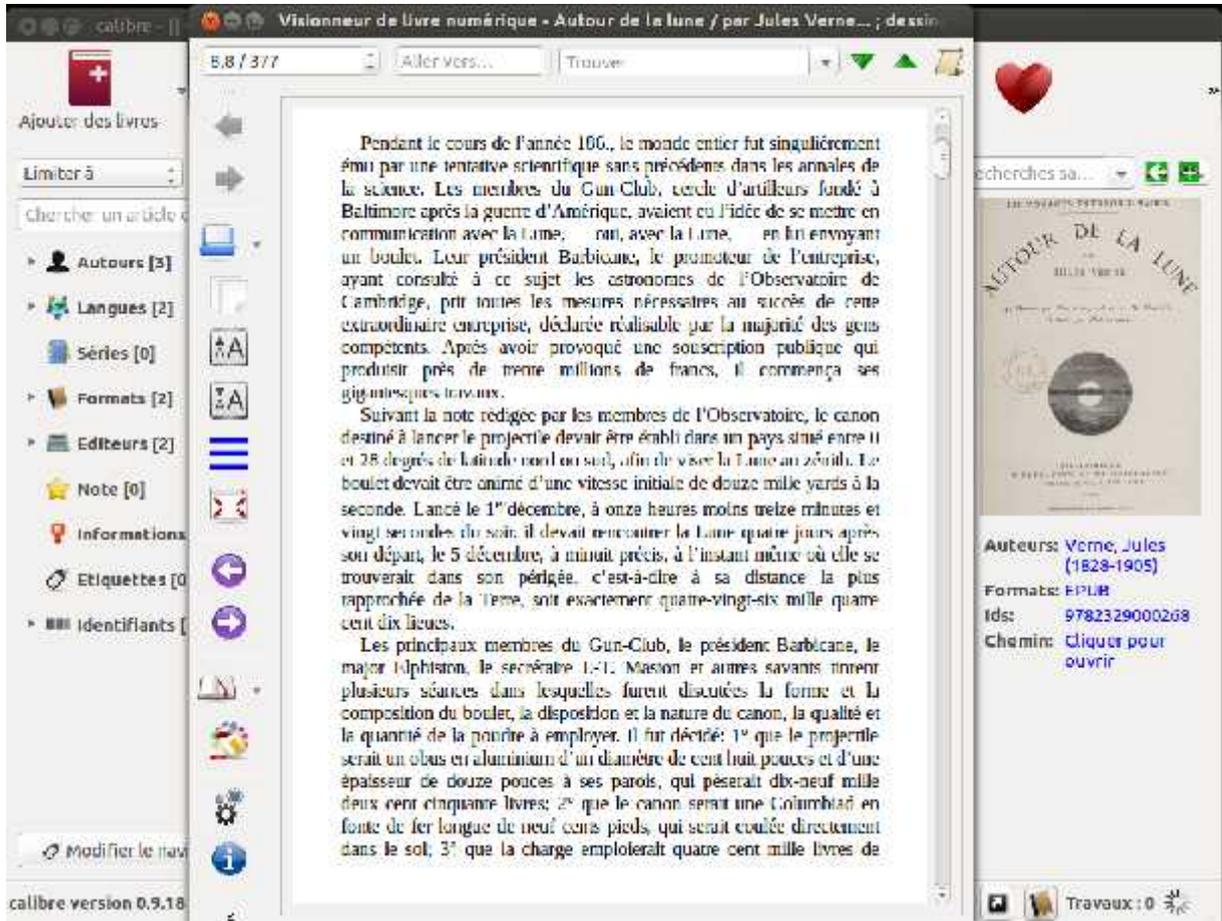


Figure 3. 12 : Visionneuse de livre numérique de calibre

3.5.3 Conclusion

Dans ce dernier chapitre, nous avons détaillé toutes les étapes liées à la construction du document électronique en y joignant les captures d'écran des différents fragments nécessaire a sa fabrication, tout en définissant les outils open source utilisés qui ont contribué à sa réalisation.



CONCLUSION GENERALE ET PERSPECTIVES

La conception de documents électroniques, vu leur complexité, nécessite d'entreprendre une approche systémique qui implique l'utilisation d'une méthodologie permettant, en premier lieu, de considérer le système avec toutes ces composantes, de concevoir et de spécifier ce système par la définition de ces composants et des liens qui existent entre eux. D'autre part, elle doit aussi donner la possibilité de décrire ces composants avec les outils existants des différents domaines technologiques. Nous nous sommes tournés vers des technologies web, en l'occurrence le langage XML, pour construire une méthodologie permettant d'entreprendre cette approche systématique supportant les activités d'ingénierie décrites dans le standard IEEE.

Au cours du développement des systèmes, les parties prenantes demandent l'application des évolutions, afin d'améliorer leurs fonctionnalités. L'occurrence d'une évolution affecte plusieurs aspects comme : la sécurité, le coût du développement et les délais. Lorsque le développement concerne un système complexe où le nombre des exigences est de l'ordre de dizaines de milliers, alors les demandes des évolutions rendent l'analyse d'impact du changement de plus en plus difficile. Malheureusement, la problématique de l'évolution des exigences n'est pas encore complètement résolue en milieu industriel.

Mots-clés :

Technologies de l'information - technologies web - ingénierie des besoins - standard IEEE

REFERENCES BIBLIOGRAPHIQUES

- [1] Cheng B.H.C., Atlee J.M., "Current and Future Research Directions in Requirements Engineering", In Design Requirements Engineering: A Ten-Year Perspective, LNBIP, Springer, Vol. 14, Part 1, p. 11-43, 2009a.
- [2] www.standishgroup.com, the Chaos report visité le 13/03/2017
- [3] Ingénierie des exigences <http://inforsid.fr/?q=node/13> visité le 14/03/2017
- [4] Writing a Requirements Document For Multimedia and Software Projects | Rachel S. Smith, Senior Interface Designer, CSU Center for Distributed Learning
- [5] The Institute of Electrical and Electronics Engineers, Inc. 345 East 47th Street, New York, NY 10017-2394, USA Publié en 1994
- [6] P. Zave, M. Jackson., Four dark corners of requirements engineering. ACM Transactions on Software Engineering and Methodology, 1-30. 1997
- [7] D. T. Ross, K. E. Schoman., Structured analysis for requirements definition. IEEE Transactions on Software Engineering , vol. 3, N° 1, 6-15, 1977.
- [8] Colette Rolland. Ingénierie des Besoins : L'Approche L'Ecriteiro. Journal Techniques de l'Ingénieur, 2003, pp.1. <hal-00706492>
- [9] MOSAIQUE Informatique : formations, créations web, rédaction d'ouvrages informatiques - Nancy Glossaire : web, référencement, infographie, multimédia, informatique X (glossaire informatique et multimédia) XML (Extensible Markup Language) - Définition <http://www.mosaïque-info.fr/glossaire-web-referencement-infographie-multimedia-informatique/x-glossaire-informatique-et-multimedia/528-definition.html> visité le 20/04/2017
- [10] Présentation des DTD Septembre 2015 <http://www.commentcamarche.net/contents/1330-presentation-des-dtd> visité le 22/04/2017
- [11] Les web services. Techniques, démarches et outils XML, WSDL, SOAP, UDDI, RosettaNet, UML. Hubert Kadima, Valérie Monfort, Dunod, 2003.
- [12] M. Kay. XPath 2.0 Programmer's Reference. Wiley Publishing, Inc.. Indianapolis. 2004.
- [13] P. Drix. XSLT fondamental. Eyrolles. 2002.
- [14] D. Tidwell. XSLT. O'Reilly. 2001.
- [15] XQuery <https://fr.wikipedia.org/wiki/XQuery> visité le 26/04/2017
- [16] <http://lesdefinitions.fr/multimedia> visité le 16/05/2017
- [17] <http://www.amba.fr/definition-multimedia-ref001115.html> visité le 18/05/2017
- [18] http://biomimi2001-yasmina.blogspot.com/2007/02/linformatique-et-linternet_13.html visité le 21/05/2017
- [19] <https://pro.arcgis.com/fr/pro-app/help/layouts/what-is-dynamic-text-.htm> visité le 24/05/2017

- [20] <http://www.commentcamarche.net/download/telecharger-34078358-calibre> visité le 30/05/2017
- [21] <http://www.opensourcemacssoftware.org/logiciel-bureautique/calibre-gestionnaire-ebooks-mac.html> visité le 04/06/2017
- [22] James C. French., Applying hypertext structures to software documentation
Information Processing & Management, Volume 33, Issue 2, March 2007, Pages 219-231
- [23] Mark L. Roth, A software documentation generator, Patent, US7568184 B1, Jul 28, 2009
- [24] Peter Haumer, Klaus Pohl, Klaus Weidenhaupt, Requirements Elicitation and Validation with Real World Scenes. To appear in IEEE Transactions on Software Engineering, Vol. 24, No. 12, Special Issue on Scenario Management, 1998.
- [25] M. Lubars, C. Potts, Charles Richter, A Review of the State of Practice in Requirements Modeling. Proceedings of the IEEE International Symposium on Requirements Engineering, RE'93, San Diego, USA, pp.214, 1993.

Résumé

Le présent travail de master rentre dans le domaine de l'ingénierie des besoins. Il consiste à faire une réingénierie du document des exigences du logiciel selon une approche de données semi-structurées. En effet la manipulation de ce document complexe dans un format non révisable tel que le format PDF ou même révisable tel que le format Word, pose beaucoup de problèmes, dont le principaux : la difficulté de chercher un contenu précis dans ce document, de mettre à jour son contenu et enfin de le lire. Ainsi nous considérons dans le contexte de ce mémoire le contenu de ce document comme un ensemble de données semi-structurées. Le travail de réingénierie consiste : (1) à lui associer un schéma (modélisation) pour veiller au respect de sa structure ; (2) à le rendre très modulaire grâce à la notion d'entité générale de XML ;(3) à lui assigner une feuille de style qui permet d'afficher son contenu sur le Web comme une page HTML ; (4)) à l'interroger via des requêtes Xquery ; (5) à le transformer en un document électronique PDF permettant une visualisation adaptée (PDF devenu un standard pour la lecture électronique). Enfin des techniques hypermédias ont été utilisées pour faciliter encore sa lecture électronique.

Abstract

The present work of master is in the field of the engineering of the needs. It consists of re-engineering the software requirements document using a semi-structured data approach. Indeed, the manipulation of this complex document in a non-editable format such as PDF or even revisable format such as the Word format poses many problems, the main one being the difficulty of looking for specific content in this document, Day its contents and finally to read it. Thus we consider in the context of this paper the content of this document as a set of semi-structured data. The reengineering work consists in: (1) associating a schema (modeling) to ensure the respect of its structure; (2) to make it very modular thanks to the

