

Publication de documentation utilisateur basée sur XML

Historique du projet

Schlumberger¹, comme beaucoup de multinationales, a mis en place un projet appelé BASIS, qui consiste à mettre en œuvre SAP², un progiciel de gestion intégré. Ce progiciel a été paramétré et modifié en interne par Schlumberger afin qu'il fonctionne selon nos processus (comptabilité, ventes, finances, etc.). Comme le produit est assez complexe à utiliser, l'utilisateur final a besoin, en plus de la formation initiale, d'une documentation en ligne (aide en ligne).

Cette aide en ligne a été conçue initialement au format WinHelp, format standard sous Windows, et en QuickHelp pour le Macintosh. Environ 5 000 pages de documentation avaient été écrites et mises à jour en 2 ans, avec une charge de 12 hommes-année. Cette documentation était diffusée à 1 000 exemplaires sous la forme d'un cédérom en Amérique du Nord. La dernière diffusion cédérom a été la version 5 en avril 1998. La documentation est rédigée par les formateurs SAP interne. Ces formateurs sont des personnes du terrain (vendeurs, managers, comptables, logisticiens, etc.) qui ont appris, en venant dans le projet BASIS, à former les utilisateurs et à écrire la documentation. Ils rédigeaient sous Word, en suivant un canevas prédéfini, puis transmettaient leurs documents, par messagerie électronique, à l'administrateur de la documentation. Ce dernier transformait ces fichiers en RTF en ajoutant des codes spécifiques, créait des index avec des macros Excel, puis compilait le tout avec le logiciel ForeHelp en plusieurs centaines de fichiers au format WinHelp (puis transformait ce format en QuickHelp pour le Mac).

Cette solution n'était pas satisfaisante, car la distribution était déficiente et le processus de mise à jour du contenu n'était pas efficace. En effet, il était impossible de garantir que tous les sites de

Schlumberger avaient reçu leur cédérom et que les utilisateurs avaient paramétré leur ordinateur pour permettre l'aide en ligne d'apparaître contextuellement depuis SAP vers le bon document contenu dans le cédérom. De plus, les mises à jour du contenu étaient envoyées par messagerie électronique par les rédacteurs et non sous la forme de fichiers Word. Il leur était impossible de repartir du format Word, car le contenu des fichiers RTF était truffé de codes spécifiques à WinHelp. L'administrateur de la documentation devait donc réaliser les modifications lui-même et retourner les fichiers WinHelp pour validation. Et bien sûr, les rédacteurs envoyaient toutes leurs remarques et ajouts quelques jours avant le bouclage...

Compte tenu des problèmes liés à la situation actuelle, la mise en œuvre d'un nouveau système éditorial basé sur XML a été décidé en septembre 1997. En effet, il était impossible d'utiliser les outils de marché permettant de créer des aides en ligne (Doc-To-Help³, ForeHelp⁴, RoboHELP⁵...), car ils ne sont pas conçus pour gérer des volumes importants de documentation, rédigés en même temps par plusieurs personnes, diffusables aux formats HTML et PDF, comprenant de multiples index. De plus, le format HTML a été choisi comme format de visualisation sur l'Intranet, car il est efficacement diffusé par des serveurs Web, il est lisible par tous les navigateurs du marché (Netscape, Microsoft,...), et il fonctionne quel que soit le système d'exploitation (Windows, Unix, Macintosh,...), ce qui n'est pas le cas du format HTML-Help⁶ (remplaçant du format WinHelp qui n'est plus supporté par Microsoft).

Le système éditorial XML est devenu opérationnel en août 1998. La mise à jour a commencé en sep-

¹ <http://www.slb.com>

² <http://www.sap.com>

³ <http://www.wextech.com>

⁴ <http://www.forehelp.com>

⁵ <http://www.blue-sky.com>

⁶ <http://msdn.microsoft.com/workshop/author/htmlhelp/default.asp>

tembre 1998. Son contenu est consultable sur l'Intranet de Schlumberger depuis janvier 1999.

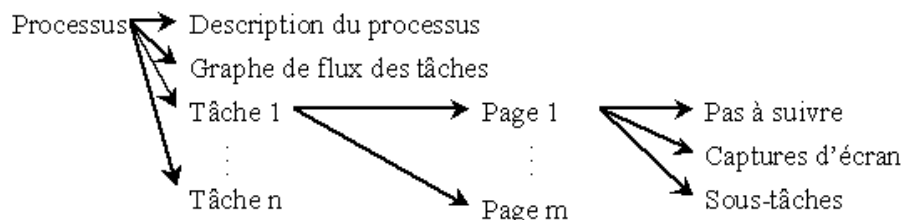
La documentation utilisateur

La documentation en ligne a pour objectif d'aider l'utilisateur final à réaliser des tâches, en lui fournissant :

- Une vue large des processus de l'entreprise.
- Des procédures détaillées pour réaliser chaque tâche.

- Des trucs, astuces et FAQ (Foires Aux Questions).

Le but est d'apprendre à l'utilisateur à pêcher. Sa structure est organisée autour du concept de processus (Figure 1). Un processus et un rapport ont la même structure. La navigation dans l'aide en ligne est réalisée via des boutons, des liens hypertextes, des tables des matières, des index et de la recherche en texte intégral.



Total: 200 Processus, 100 Rapports, 500 Tâches, 2000 Captures d'écran

Figure 1 : Structure de la documentation

La documentation est modulaire. Il y a 800 documents texte (500 tâches, 300 descriptions), 2 000 captures d'écran, 200 graphes de flux. Ces fichiers sont reliés entre eux par plus de 4 000 liens hypertextes. Les documents textes sont maintenant au format XML. Chaque document XML et chaque graphe de flux contiennent des méta-data pour le décrire et pour définir son applicabilité⁷. Chaque fichier est identifié par une nomenclature à plusieurs niveaux.

L'aide en ligne sur l'interface Web disponible sur l'Intranet de Schlumberger est illustrée par les Figures 2 et 3.

XML

XML a été choisi comme format de stockage de la documentation, car il permet, entre autres :

- D'imposer une structure au contenu des documents.
- De publier sous différents formats (HTML, PDF, Word,...).
- De s'affranchir des formats propriétaires (Word, FrameMaker...).

XML signifie *eXtensible Markup Language*. C'est une recommandation du W3C (World Wide Web Consortium⁸) datée de février 1998. XML⁹ est un sous-ensemble de SGML (Standard Generalized Markup Language). Le but de XML est de permettre au SGML générique d'être diffusé, reçu et traité, comme cela est possible maintenant avec HTML.

HTML est un langage destiné à la visualisation de document dans un navigateur Web. HTML est une application de SGML (une DTD = *Document Type Definition*). XML est un méta-langage conçu pour définir le contenu de documents ; c'est une version abrégée de SGML, qui facilite la création de types de documents et la programmation de logiciel pour les traiter. Le haut de la Figure 4 décrit un début de page HTML qui affiche un titre centré, puis un sous-titre et une liste à puce avec un seul élément. Le titre de l'élément de la liste est en gras. Les balises sont entre "<" et ">". Elles sont génériques à HTML (par exemple pour gras). Le bas de la Figure 4 décrit la structure du même document en XML. Les balises sont spécifiques au besoin (par exemple <Purpose> identifie une structure spécifique). Les balises peuvent avoir des attributs (par exemple "ID=" sert à identifier une balise de manière unique avec un

⁷ Document est applicable dans un contexte précis, par exemple pour une région du monde, une langue, un « module » de SAP, un type d'utilisateur...

⁸ <http://www.oasis-open.org>

⁹ <http://www.xml.com/xml/pub>

code, ici 8837, afin de pouvoir placer des liens vers celle-ci). Il n'y a pas de formatage, car celui-ci dépend de l'interface (papier, navigateur Web, ...). On voit donc que XML permet de définir une structure qui pourra ensuite être traitée par

programme. Ce n'est pas le cas de HTML. On peut très facilement convertir automatiquement du XML vers du HTML. L'inverse n'est pas vrai.

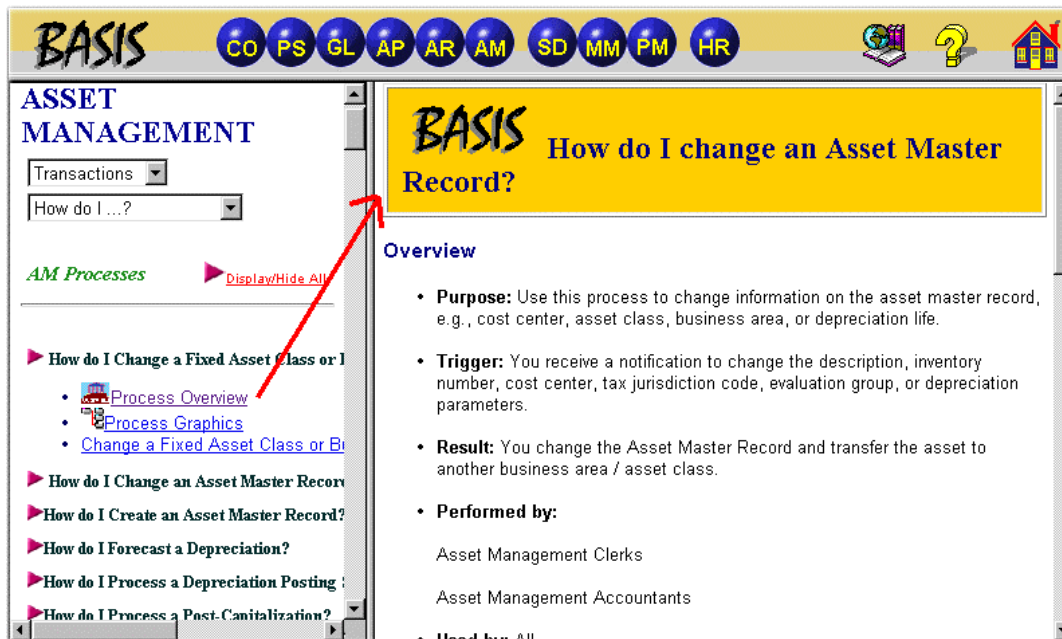


Figure 2 : Interface Web → une Description de Processus

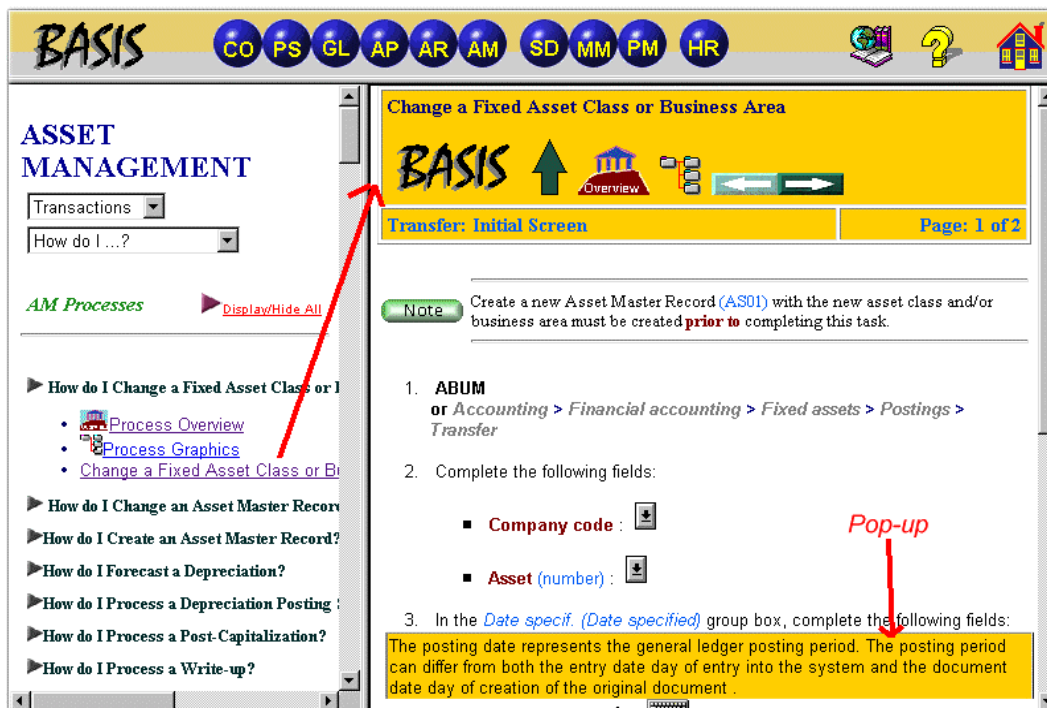


Figure 3 : Interface Web → une Tâche

```
HTML
<CENTER><H3>How do I change a Master...</H3></CENTER>
<H4><FONT COLOR="#006600">Overview</FONT></H4>
<UL>
<LI><B>Purpose: </B>Use this process to change...</LI>
</UL>

XML
<Title ID="8837"><Title-body>How do I change a Master...</Title-body></Title>
<Overview>
  <Purpose><Para><Text>Use this process to change...</Text></Para></Purpose>
  <Trigger><Para><Text>You receive a notification...</Text></Para></Trigger>
</Overview>
```

Figure 4 : Exemple de HTML et XML

Nouveau système éditorial XML

Le but du nouveau système éditorial basé sur XML est d'harmoniser l'édition des documents, faciliter les mises à jour, gérer les traductions, gérer les variantes des documents (par région du monde, par ligne de produit Schlumberger), publier sur différents média (Web, papier) et éditer des versions spécifiques à la demande.

Les outils suivants ont été choisis : un éditeur XML (*Adept Editor*, de Arbortext¹⁰), un logiciel de gestion documentaire XML (*NovaManage*, de NovaSoft¹¹ + le middleware SGML/XML *Balise*, de Inso¹²), un convertisseur de XML vers HTML (*HTML-Package*, de Inso), et un outil de composition de XML vers PDF (*FrameMaker + SGML*, de Adobe¹³). L'outil de dessin des graphes de flux est Visio¹⁴ et celui de capture d'écrans est Paint-Shop Pro¹⁵.

Spécifications et conversion

La première étape — la plus importante, car elle conditionne le succès du projet — consiste à analyser les documents existants pour en définir la structure. Les documents RTF existant étaient de deux types, des « Descriptions de Processus » et des « Tâches ». Les documents de chacun de ces types étaient assez homogènes, grâce à la qualité

du travail de l'administrateur de la documentation. Il a donc été décidé de décrire ces deux structures par deux DTD (Document Type Definition) XML. La meilleure méthode pour construire des DTD consiste à réutiliser des DTD existantes (du domaine public, de son industrie, de son entreprise...), afin d'éviter de réinventer la roue. Nous avons choisi d'utiliser une DTD de documentation technique de Schlumberger en la simplifiant et en créant deux DTD à partir de ce noyau commun : une pour les *Descriptions de Processus* et une pour les *Tâches*. Des balises spécifiques à la documentation d'une interface utilisateur ont été ajoutées à ces DTD (par exemple pour identifier le nom d'un champ à remplir dans un formulaire). Cette phase de conception étant très importante, il est indispensable de se faire assister par un expert (un sous-traitant expert en documentation structurée par exemple).

La Figure 5 présente la structure de la DTD la plus simple, *Descriptions de Processus*. Elle correspond au document montré dans la Figure 2. Le document XML débute par un *status*, qui contient des méta-data (non montrées ici). Le contenu (*content*) est composé obligatoirement d'un titre (*title*), suivi de *overview*, puis de *need-to-know-first*, enfin de *tasks-to-perform*. La balise *title* ne contient pas de texte, car elle est elle-même composée d'un *title-body* qui contient le texte du titre, puis de 0 à n *alt-title* qui contiennent les textes des titres alternatifs. Chaque paragraphe est décrit par la structure *para*, qui est composée d'un mélange de texte, de listes non ordonnées, de listes ordonnées, de figures et de tableaux, suivie d'un mélange de notes et d'alertes.

¹⁰ <http://www.arbortext.com>

¹¹ <http://www.novasoft.com>

¹² <http://www.balise.com/>

¹³ <http://www.adobe.com>

¹⁴ <http://www.visio.com>

¹⁵ <http://www.jasc.com/>

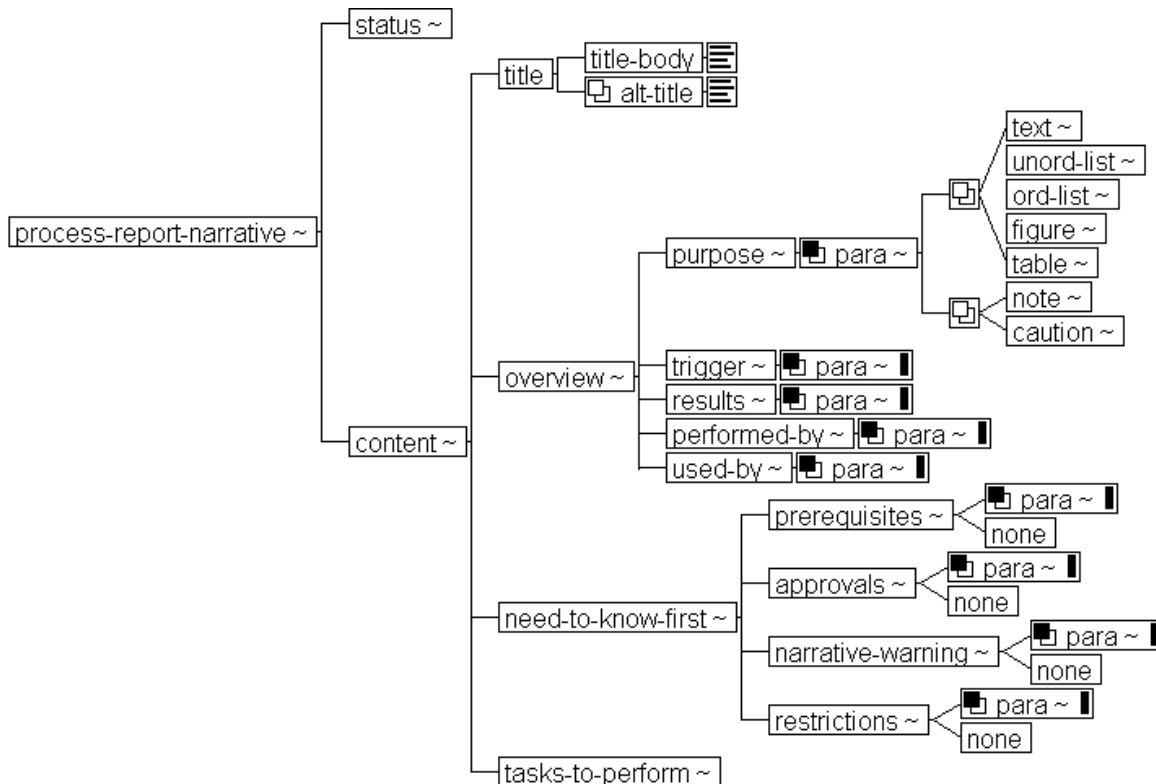


Figure 5 : Exemple de DTD → une Description de Processus

La deuxième étape consiste à convertir l'existant en XML, afin de pouvoir mettre à jour le contenu et produire les éditions électroniques. Les documents source sont des fichiers RTF contenant des codes spécifiques à WinHelp. Un programme de rétro-conversion, basé sur le middleware SGML-XML *Balise*, a été conçu spécifiquement pour transcoder les codes de mise en page de RTF (le gras, l'italique, les styles, les listes, les icônes...) en balises XML sémantiques. Il utilise des heuristiques permettant, dans un contexte donné, de fournir une structure cible. La Figure 6 illustre ce processus ; par exemple une liste à puce commençant par le mot « Purpose » avec au début l'icône « smallblue.bmp » et le mot « overview » devient une balise ouvrante « <purpose> » située dans la structure « <overview> ».

Ce travail de rétro-conversion est capital, car il n'est effectué qu'une seule fois. C'est un travail itératif qui nécessite beaucoup de doigté et de patience. L'administrateur lance un programme « batch » qui convertit un lot de documents et analyse les « logs ». Les documents convertis sans erreur sont enlevés de la liste. Chaque document

ayant produit une erreur est étudié ; soit le RFT source est modifié pour correspondre aux spécifications du programme de rétro-conversion et il repasse à la moulinette, soit le XML résultant est modifié et le fichier est parsé avec balise pour s'assurer qu'il respecte la DTD. Le programme de rétro-conversion a été amélioré au fur et à mesure des tests, ce qui nous a assuré la réalisation de notre objectif de conversion : 80 % des documents au minimum convertis sans intervention humaine.

De plus, les 2 000 images raster au format Bitmap ont été converties au format GIF et les 200 graphes de flux ont été convertis d'un format propriétaire à ForeHelp (SHEG) au format Image Map¹⁶ utilisé sur le Web.

¹⁶ Une image GIF représentant l'organigramme et un fichier HTML décrivant les coordonnées des zones actives avec des liens hypertextes.

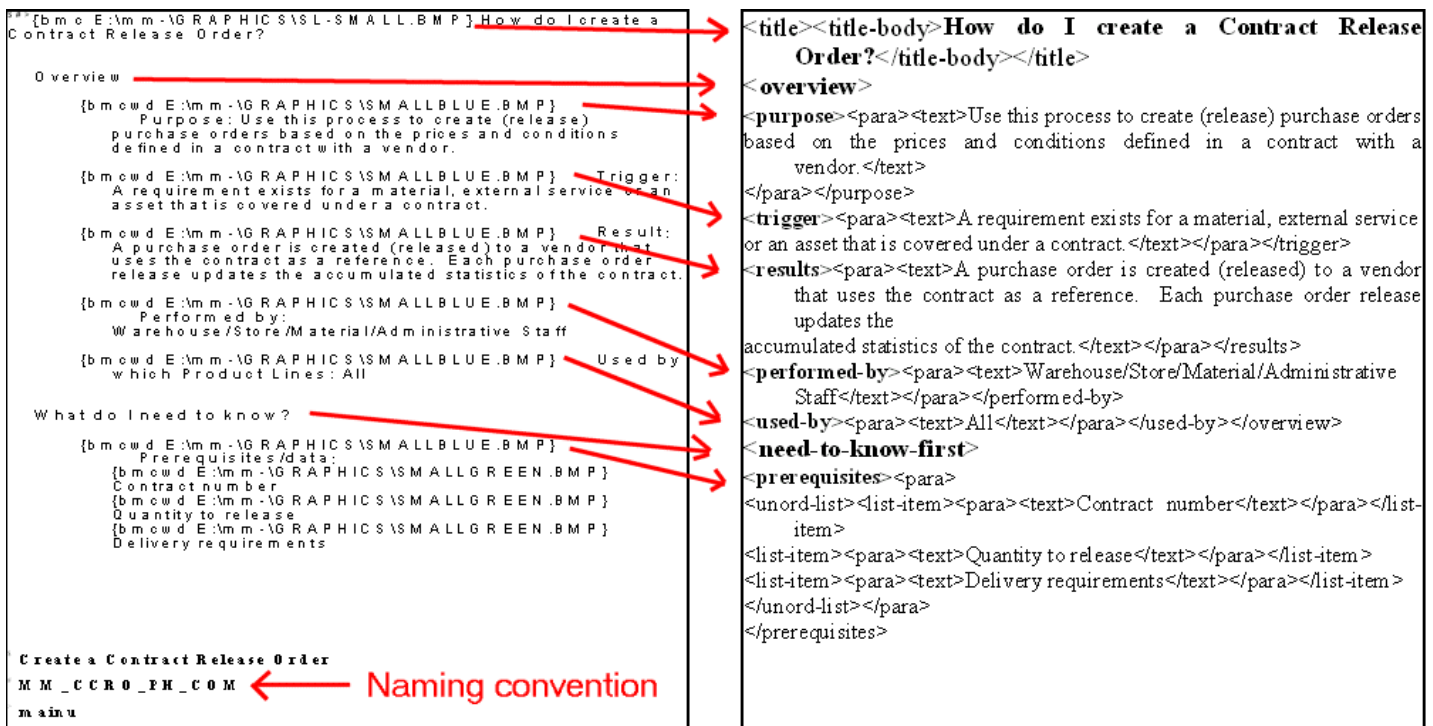


Figure 6 : Exemple de conversion RTF → XML pour une description de Processus

Processus éditorial

Une fois les documents convertis en XML, la mise à jour de la documentation a pu commencer. Le processus éditorial est réalisé par 12 rédacteurs (les formateurs) situés dans le même étage. Ils éditent les documents sous *Adept Editor*, qui est un éditeur structuré permettant une saisie contrôlée par la DTD. Des macros lancées à partir d'un menu spécifique automatisent la création de structures complexes. Ils peuvent visualiser le résultat de leur travail en HTML, tel que le verrait l'utilisateur final de la documentation, grâce à un convertisseur à la volée XML vers HTML.

L'édition dans *Adept Editor* est illustrée par la Figure 7 et le résultat en HTML est présenté Figure 2. Les titres en gras dans la Figure 7 sont générés automatiquement par l'éditeur à partir d'une feuille de style. Par exemple, le mot « Overview » s'affiche dans la fenêtre de droite en Arial gras de taille 15 points avec un espace en dessous, car il correspond à la balise <overview> affichée dans la vue arborescente de la fenêtre de gauche.

Tous les documents sont gérés par la base de données documentaire *NovaManage*, qui contrôle les accès, attribue les numéros de version et garantit l'unicité des documents. Lorsqu'un document (XML ou graphes de flux) est rentré dans la

base après édition, ses attributs sont automatiquement mis à jour à partir des méta-data du document. Cette mise à jour automatique est réalisée via *Balise*, qui analyse grammaticalement et syntaxiquement (*parse*, en anglais) les documents entrés par rapport à leur DTD, extrait les méta-data, et alimente les champs de la base. Les méta-data sont mises à jour dans le document XML sous *Adept Editor* et dans les graphes de flux avec un formulaire sous Visual Basic.

Les graphes de flux subissent un traitement spécial, car il n'y a pas de facilité de visualiser des documents vectoriels dans un navigateur Web. Ils sont conçus sous *Visio*, des liens hypertextes sont posés sur les objets graphiques et les fichiers sont sauvegardés dans la base dans le format Visio 5. Chaque fichier Visio est aussi sauvegardé sous le format Image Map. Le rédacteur lance ensuite un programme Visual Basic qui demande via un formulaire de saisir les méta-data, puis compresse au format ZIP l'image GIF, le fichier HTML avec les liens et un petit fichier XML avec les méta-data. Le graphe de flux au format ZIP est sauvegardé dans la base. Le ZIP permet de ne gérer qu'un seul fichier dans la base au lieu de trois.

Un glossaire central est administré par la base de données. Il contient des termes, des définitions, et des applicabilités pour les définitions.

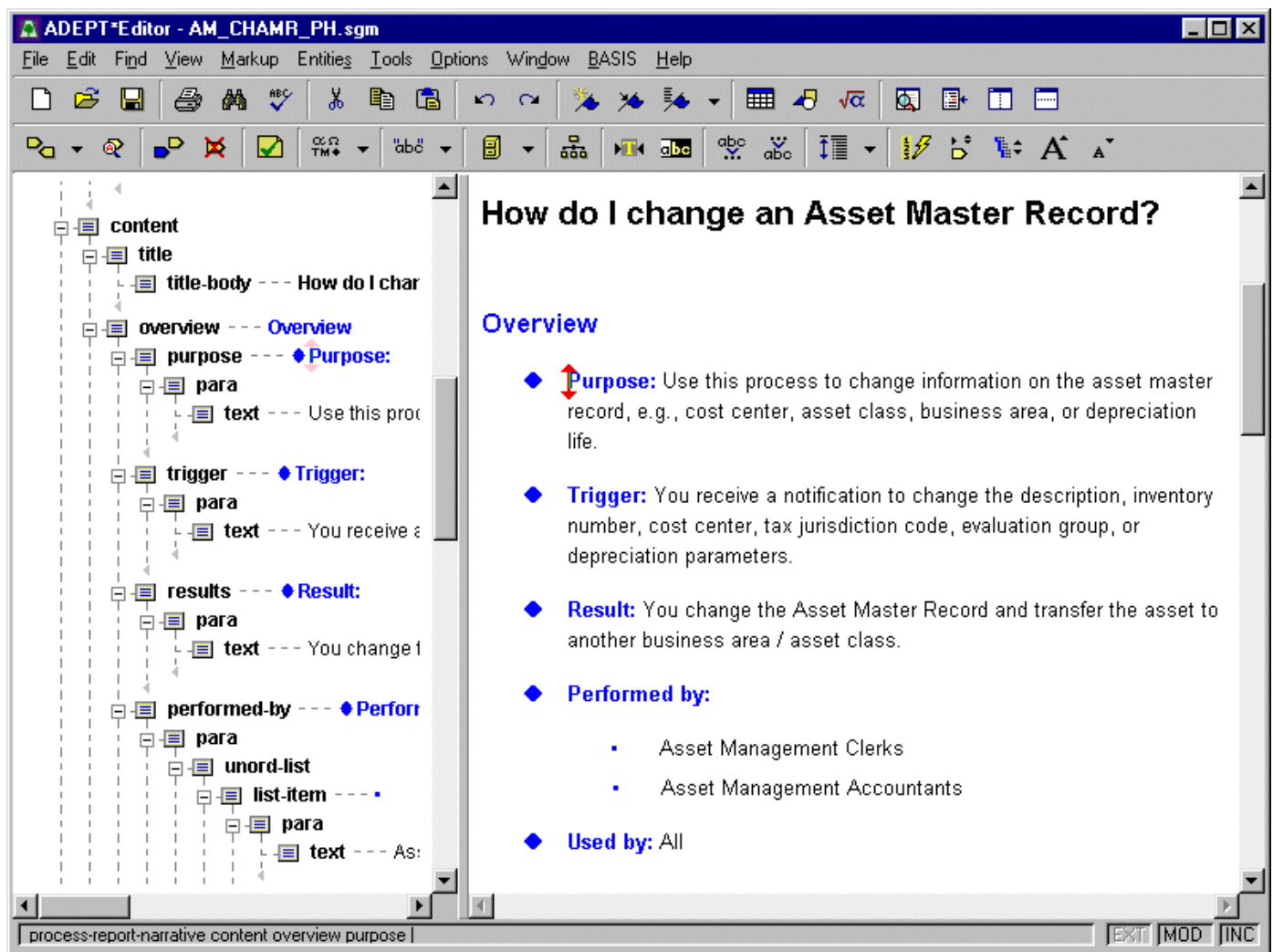


Figure 7 : Édition d'une description de Processus dans Adept Editor

Le contrôle qualité des documents est réalisé par différents acteurs :

- *Adept Editor* : la conformité aux DTD.
- L'administrateur : le respect des procédures et des règles de rédaction.
- Les experts sectoriels du projet BASIS : le contenu.
- La base de données et le programme de publication : validation des liens hypertextes entre documents (XML, captures d'écran et graphes de flux).

L'édition en XML sous *Adept Editor* est un succès :

- Les formateurs se sont rapidement habitués à XML et à un environnement d'édition structuré qu'ils ne connaissaient pas.
- Les DTD leur ont permis de se concentrer sur le contenu et non sur la mise en page.

- Les DTD et le contrôle qualité permettent d'assurer une documentation consistante.
- La visualisation en HTML permet aux rédacteurs de voir en direct ce que l'utilisateur va voir.
- La base de données documentaire est facile à utiliser.

Il a été cependant indispensable de mettre en place un « help-desk » dès le début du processus de mise à jour de la documentation afin d'assister les rédacteurs, de leur permettre d'exploiter au mieux les outils, de rédiger selon les règles et surtout de maîtriser les deux DTD. Sans ce support de haut niveau, le projet n'aurait pas été un succès. En effet, il est difficile au début de passer d'une habitude de rédaction sans contrainte sous Word à une rédaction dans un environnement structuré et contrôlé. Par exemple, dans la Figure 7, le rédacteur ne crée pas un item dans une liste à puce avec « Purpose » en gras et un texte après, mais il crée

une structure appelée <purpose>, puis il insère une structure <para> qui contient la balise <text> qui contient le texte lui-même.

Le système éditorial actuel a quelques inconvénients. Cette saisie contrôlée a pour inconvénient de figer la structure. Il nous est apparu nécessaire de la faire évoluer à l'usage. L'autre inconvénient important est la gestion des graphes de flux sous deux formats (Visio et Image Map zippé). La normalisation récente des dessins vectoriels et raster au format Web CGM¹⁷ permettra de résoudre ce problème, en autorisant une sauvegarde dans un format unique, CGM (*Computer Graphic Metafile*).

Processus de publication

Chaque semaine, le contenu de la base est publié automatiquement sur l'Intranet de Schlumberger. Les derniers documents mis à jours sont exportés de la base (XML, GIF et Image Map), puis un programme en Balise transforme les fichiers XML en pages HTML (entre 1 et 20 pages HTML par document XML selon la complexité de la tâche à documenter). Le programme d'export construit aussi une trentaine d'index, une dizaine de table des matières et place les liens hypertextes entre les documents en suivant la nomenclature¹⁸.

Plus de 3 000 pages HTML sont produites à partir des 800 fichiers XML. Un compilateur HTML-Help permet de vérifier s'il n'y a pas de liens hypertextes cassés. Des fonctions JavaScript permettent d'afficher des « pop-up » et du Dynamic HTML, quelque soit la marque du navigateur (version 4 et au-dessus).

Un pré-traitement permet de créer une structure de fichiers permettant de court-circuiter l'aide en ligne standard du logiciel SAP, pour insérer la nôtre à la place lors d'un appel à l'aide (menu *Help*) depuis l'interface cliente SAP sur le poste de travail de l'utilisateur. L'aide en ligne de SAP est néanmoins disponible depuis la nôtre.

La Figure 8 illustre le processus de publication. Lors de la transformation du XML en HTML, toutes les informations éparpillées dans différentes bases sont fusionnées dans les documents HTML. Les liens vers l'aide en ligne de SAP sont ajoutés en fonction de l'applicabilité du document XML. Les définitions du glossaire sont rajoutées dans les pa-

ges HTML, aux endroits correspondant à une balise identifiant un appel au glossaire dans le document XML source, pour être affichés sous la forme de pop-up.

La production de manuels en PDF passe par un outil de composition : FrameMaker + SGML. Cet outil utilise en entrée les fichiers issus de l'export et construit (via un programme Balise) plusieurs manuels.

La traduction du contenu de l'aide est envisagée. La langue de rédaction est actuellement l'anglais. Les traductions se feront en français et en espagnol, probablement semi-automatiquement via un outil de traduction à mémoire, du type Trados¹⁹. La gestion de la documentation sous forme modulaire dans une base de données documentaire se justifie pleinement dans ce cas, car il est possible, grâce aux mécanismes de lien inter-documents, d'identifier les documents qui ont été modifiés dans le langage source et de repérer les seuls documents dans les langues cibles qui sont à traduire.

Leçons apprises durant le projet

La préparation d'un projet de ce type est très importante. Il est indispensable de rassembler de la documentation, de se renseigner sur XML, de consulter des experts (internes ou externes), de se renseigner sur les fonctionnalités des logiciels de gestions documentaires, de posséder les bons outils (parseur XML, éditeur XML, visualisateur graphique de DTD) et surtout de planifier le projet.

La phase de rédaction des spécifications est capitale. La création des DTD ne peut se faire qu'après analyse fine de la structure des documents et de leur complexité. La spécification des outils de rétro-conversion est importante, car le résultat final en dépend et donc, la quantité de corrections manuelles à effectuer. Il est indispensable de se rendre dans des salons pour se renseigner sur les logiciels de gestions documentaires et identifier les bons candidats. Lors de la phase de spécification de l'application, une bonne analyse des processus en cours est indispensable.

Il est primordial d'avoir de bonnes relations avec votre sous-traitant. Il faut être prêt à discuter de points techniques avec lui, sinon il faut faire venir un expert pour vous assister (interne ou externe).

¹⁷ <http://www.cgmopen.org/>

¹⁸ Tous les documents sont identifiés par une nomenclature.

¹⁹ <http://www.trados.com/>

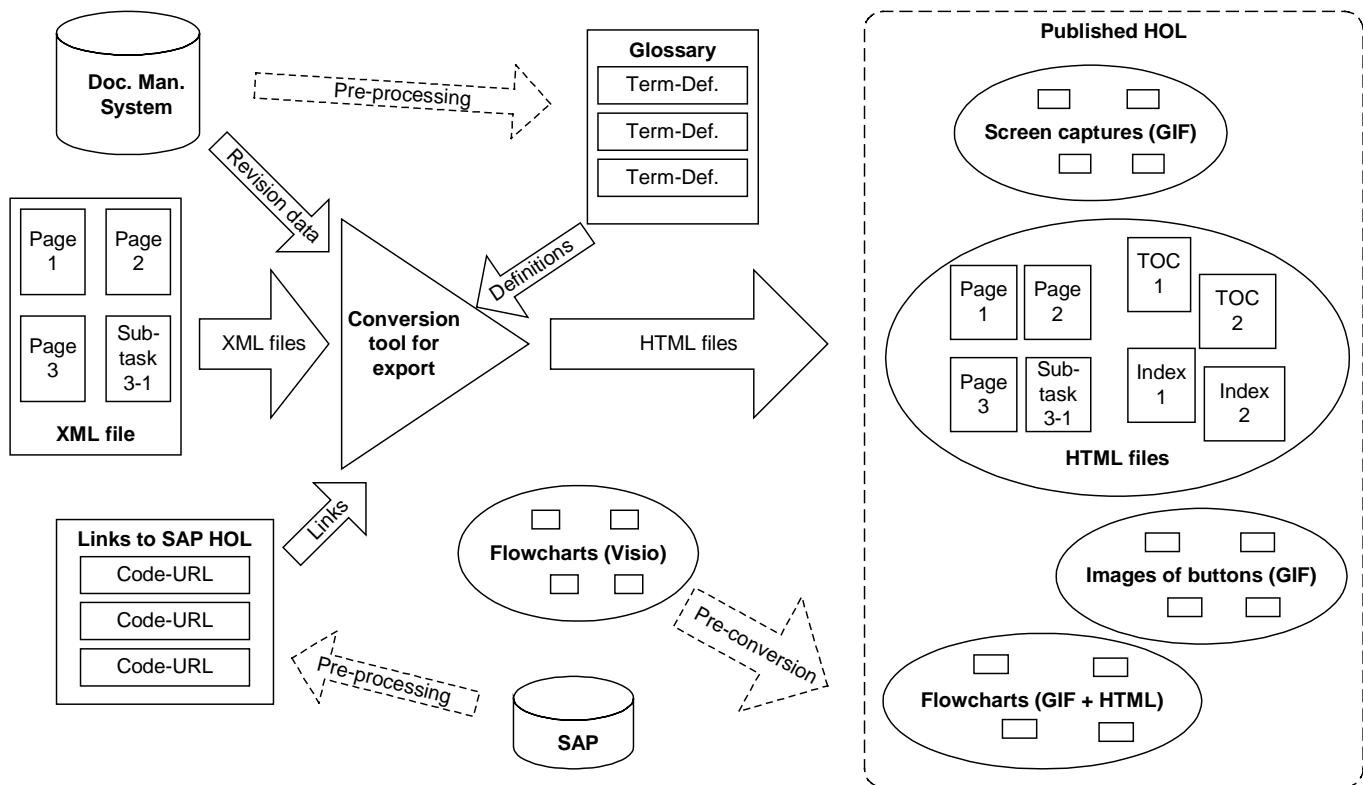


Figure 8 : Processus de publication

Il faut tester le plus possible les livraisons, en partant du principe que le sous-traitant ne teste jamais assez par manque de temps. Le sous-traitant a souvent des bonnes idées, issues de son expérience du terrain ; il faut l'écouter. Enfin, il faut bien spécifier les livrables, en particulier la documentation des programmes.

Avant le lancement officiel de l'application, il faut se forcer à tester et encore tester, car on a toujours oublié des cas particuliers. L'installation des logiciels sur les postes des utilisateurs n'est pas un détail à négliger. Notre installation occupait environ 120 Mo de disque et prenait environ ½ journée par poste, à cause des environnements tous différents de chaque utilisateur. La formation des rédacteurs ne doit pas être sous-estimée ; deux jours peuvent être nécessaires pour former à l'utilisation des DTD. La documentation utilisateur n'est pas à négliger non plus, même si votre « help-desk » est à côté des rédacteurs.

Après le lancement, la mise en œuvre d'un « help-desk » pour les rédacteurs est indispensable. Vous devez être aussi prêt à mettre les mains dans l'application pour effectuer des réglages. Il vous faut aussi passer beaucoup de temps au début pour effectuer des contrôles qualité sur les do-

cuments rédigés et pour organiser des réunions de mise au point du style de rédaction. Enfin, il est temps d'enregistrer les petits problèmes, qui feront l'objet de mises à jour de la part du sous-traitant ou d'évolutions futures.

Conclusion

Les principes mis en œuvre dans ce système éditorial XML sont applicables à tout système éditorial destiné à produire de la documentation technique (documentation de maintenance, documentation utilisateur, aide en ligne...). Il est ainsi possible de réutiliser les modules de la documentation modulaire dans plusieurs documents livrés (par exemple, une même opération de démontage-remontage dans un manuel de maintenance et dans un manuel d'exploitation). Il est aussi possible de filtrer l'information afin de fournir plusieurs versions, l'une étant un sous ensemble de l'autre (par exemple les documents qualité pour les ingénieurs de l'entreprise et les documents qualité pour les clients de l'entreprise).

Louis-Pierre GUILLAUME (88)
E-mail : guillaume1@slb.com