

SPECS, un système pour la création de contenus éducatifs

Par/ **EVERARDO Reyes Garcia**

Director, B.A. Animation and Digital Arts,
Communication Department, ITESM Campus Toluca- Mexico

Résumé

Dans cet article, nous présentons notre prototype de système nommé *SPECS* (*Structured Pedgogical Content System*), développé dans le cadre de notre travail de recherche au sein du Laboratoire Paragraphe de l'Université Paris VIII.

Le principal objectif de la réalisation du système *SPECS* a été de supporter les requis de base pour la création de documents pédagogiques numériques. Dans son état actuel, le système constitue une application informatique orienté Web qui est le produit d'une série d'évolutions et d'améliorations résultant principalement de son emploi dans un contexte éducatif réel [REY 05a, 05b et 06].

1. Introduction

SPECS est l'acronyme de *Structured Pedagogical Content System*. Il est un système qui s'inscrit dans le domaine des hypermédias pédagogiques permettant la création de documents pédagogiques numériques suivant un modèle prédéfini. Toutes les phases du développement de *SPECS* ont été encadrées au sein du Laboratoire Paragraphe de l'Université Paris VIII. Dans son état actuel, il s'agit d'un système hypermédia adapté pour un fonctionnement basé sur le Web, c'est-à-dire qu'il a été écrit à l'aide de technologies Web et que les documents produits avec lui respectent les standards W3C.

Bien que le développement de certains composants de *SPECS* ait été inspiré de l'édition hypermédia, notre système a la particularité de se centrer sur les structures et non sur le formatage. L'objectif est double. Premièrement, permettre aux utilisateurs d'élaborer des documents suivant un modèle préexistant depuis une interface Web, et deuxièmement, garder une persistance de la structure au niveau syntaxique, sémantique, et pragmatique à tous les niveaux de la création et de la consultation. Il peut être dit que, dans sa forme la plus simple, *SPECS* est un système auteur pour structurer, modéliser, gérer et représenter des documents selon à un modèle sémantique donné.

Nous précisons que lorsqu'on parle d'une orientation Web, on entend qu'il s'agit d'un système développé pour une utilisation à partir du *World Wide Web*, en profitant des technologies de réseau et de protocoles Internet. Ce choix technique répond à des raisons pratiques car les utilisateurs ciblés n'utilisent pas un même système d'exploitation informatique. Par contre, la grande majorité dispose d'un navigateur pour explorer le Web. Dans ce sens, le moteur de rendu d'un navigateur, par exemple Gecko dans Mozilla Firefox, fonctionne aussi bien sous plateforme Macintosh que Windows ou Linux.

Notre prototype a été développé avec des langages libres et populaires sur le Web afin de garantir de possibles améliorations par des tiers personnes et assurer une installation compatible avec la plupart des serveurs. Le code source est écrit en PHP et XML. Les données primaires sont également sauvegardées dans une base de données MySQL pour rendre possible la modification du contenu.

Pour le développement de la version actuelle de *SPECS*, nous avons pris en compte les expériences faites au cours de son évolution et surtout de son utilisation dans un contexte réel. La figure 1 montre l'interface correspondant à la version la plus récente, réalisée pour l'Université Paris VIII.



Figure 1. SPECS : Versions les plus récentes des interfaces de gestion et auteur

2. Cadres de fonctionnement

2. Architecture et composants

Les principaux composants de SPECS sont :

- une base de données : utilisée pour modifier et chercher des documents ;
- deux répertoires pédagogiques, un pour les documents créés et un autre pour les objets média. Ils sont utilisés pour stocker et assigner des URL aux documents et aux objets médias ;
- un répertoire de structures : une série de règles et de feuilles de styles pour l'affichage de documents ;
- une interface graphique d'utilisateur : elle fournit des contrôles et des boutons pour marquer les contenus, ainsi que pour ajouter des éléments complexes ;
- un générateur de documents XML : il récupère l'information saisie par les auteurs pour produire des fichiers XML selon un modèle sémantique ;
- une collection de services : un moteur de recherche, un moteur d'importation de documents, un moteur d'exportation de documents, une série de vues pour les documents

Les composants et l'architecture du système peuvent être visualisés de manière schématique dans la figure 2.

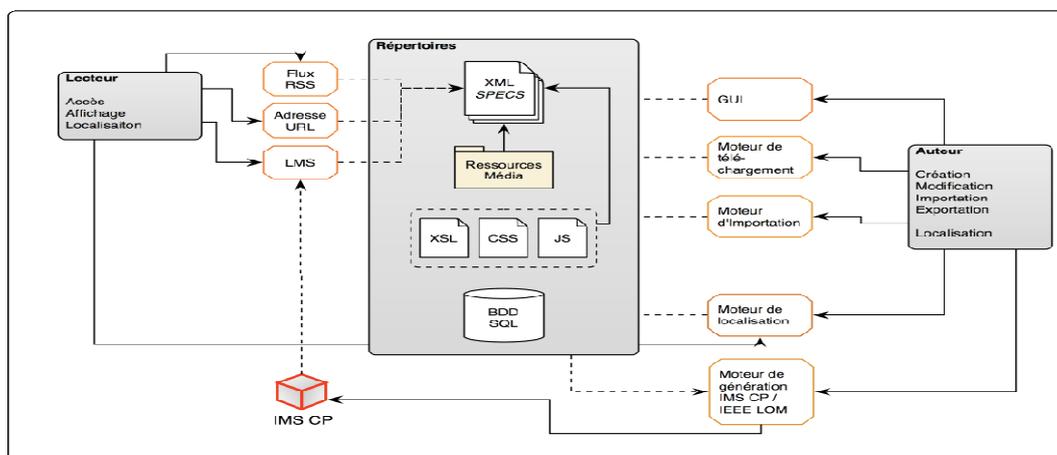


Figure 2. SPECS : architecture et composants

3. Fonctionnement

Deux types d'utilisateurs sont principalement envisagés pour l'utilisation du système : les auteurs et les lecteurs. Un auteur est la personne qui interagit avec le système visant pour la création ou la gestion de documents, ce peut être un enseignant, un administratif, ou tout acteur disposant d'un mot de passe pour accéder à l'interface d'auteur. Pour sa part, un lecteur est la personne qui interagit avec le système sans produire pas de nouveaux documents, il peut être un apprenant, un enseignant, un « *surfeur* » quelconque sur le Web, ou une machine.

Les interfaces d'utilisateur sont donc différentes pour les deux types d'acteurs. Les auteurs ont accès à une interface qui permet de réaliser des opérations qui incluent l'édition, la publication, l'effacement, la modification, la recherche, l'exportation ou l'importation de documents pédagogiques numériques et de ressources médias. Pour les lecteurs, l'interface est limitée à la visualisation (interactive ou non, c'est-à-dire en choisissant parmi diverses formes de représentation d'un document) et à la recherche de documents et de ressources. Un lecteur ne peut pas modifier un contenu.

Le mode d'opération du système au niveau des auteurs commence à partir du moment où l'on accède à l'interface auteur, qui est sécurisée par mot de passe. À l'intérieur, on dispose des panneaux et des boîtes à outils (figure 3). Lorsqu'il s'agit d'éditer, modifier ou réutiliser un document, l'auteur accède à une interface similaire à des formulaires Web, mais enrichie par plusieurs fonctionnalités interactives qui permettent de manipuler le balisage XML.



Figure 3. SPECS : balisage XML

Les documents sont ensuite générés dynamiquement et stockés sous format XML, accessibles depuis une adresse URL. Chaque document contient par défaut une référence d'affichage du type hypertexte navigable. Sur ce point, nous profitons de l'avantage des navigateurs modernes pour supporter nativement les technologies XML du côté client. Ceci permet la distribution et l'affichage de documents directement sous format XML, sans avoir besoin de transformations vers XHTML. De plus, nous avons ajouté deux couches hérités du DHTML : la navigation dans le contenu est contrôlée par le langage JavaScript et la charte graphique par une feuille de style CSS. Cette séparation permet de modifier la charte graphique d'une série de documents tout en gardant la même composition d'éléments de bloc et vice-versa.

Depuis une autre interface, le gestionnaire de documents et de médias, les auteurs ont accès à toutes les ressources et documents créés ou importés avec le système. Les utilisateurs ont également à leur disposition un moteur de recherche dans les métadonnées pour faciliter la localisation de documents et d'objets médias.

Afin de répondre aux normes e-learning lors de l'exportation des documents, *SPECS* incorpore un générateur de « paquets de contenus » conforme à la norme IMS CP ou SCORM CAM. Il est aussi possible de générer à la demande un troisième fichier selon le standard IEEE LOM, qui définit exclusivement les métadonnées pédagogiques. À travers ce mode d'exportation, les documents pédagogiques numériques s'insèrent dans une plateforme éducative en remplissant leur fonction de module et non seulement à titre de ressource indépendante. De cette manière, l'auteur peut construire un cours ou un scénario pédagogique selon divers modules qui pourront s'intégrer dans un ensemble logique géré par un LMS, par exemple.

La fonctionnalité d'importation n'est supportée qu'entre systèmes respectant un même modèle sémantique prédéfini. Pour ce faire, la technique AJAX a été une utilité pour insérer dynamiquement l'information dans la base de données et dans le répertoire de documents. Nous considérons que cette fonctionnalité trouve sa valeur lorsque, par exemple, diverses universités emploient le système. Ainsi, l'échange et la réutilisation de documents créés avec notre système, ou avec l'outil de préférence des auteurs pour créer des documents pédagogiques numériques (par exemple, *XMLSpy* sous Windows ou *Oxygen* sous Macintosh), deviennent aisés.

Quant au mode d'opération du système au niveau des lecteurs, *SPECS* propose une interface de recherche de documents et une liste de documents, créés ou importés, ordonnée thématiquement. L'accès aux documents est libre avec un navigateur Web conventionnel. Cependant, il est préférable de posséder une version récente de Mozilla Firefox, Internet Explorer, Safari, Camino ou Netscape car les documents sont livrés sous format XML et une grande partie de navigateurs de première génération ne supporte pas leur affichage.

À ce sujet, les documents contiennent par défaut une référence vers des règles XSL (un modèle d'assemblage) pour montrer le contenu sous une forme navigable. Il est envisageable, néanmoins, qu'au fur et à mesure que plusieurs structures ou modèles d'assemblage se développent, elles s'intègrent dans le répertoire de structures. À son tour, ces nouvelles structures sont proposées comme services depuis un affichage quelconque permettant ainsi que les lecteurs puissent changer non seulement la vue, mais aussi le type d'hypertextes dans lesquelles ils naviguent.

Étant donné que *SPECS* permet l'adressage URL de documents et que leur accès est libre, les manières de distribuer les documents sont variées : par référence de lien depuis un LMS ou un autre système de création ; par l'intégration de documents dans des super-structures (scénarios pédagogiques, *Walden's Paths*) ; par le découpage de documents dans des structures plus petites adaptées à une représentation interactive (mSpaces).

Une autre manière d'accéder aux documents repose sur la syndication du flux RSS intégré, qui communique les nouveaux documents créés à sans avoir besoin de revenir au site de manière régulière. C'est dans ce cas-là que le rôle de lecteur est réalisé par une machine. Un lecteur de fils RSS est couramment nommé « agrégateur » et sa fonction est de « parser » le document RSS afin de récupérer les derniers ajouts. Actuellement, plusieurs types d'agrégateurs existent sur le Web (par exemple, *Google Reader*) et plusieurs d'autres peuvent s'installer en local¹.

4. Le modèle de document

Comme il a été indiqué, un modèle de document est une structure logique permettant de donner les règles syntaxiques aux informations. Cette structure permet à un système d'information de comprendre que `<numéro_téléphonique>01 49 40 67 58</numéro_téléphonique>` représente bien un numéro de téléphone et non pas une quelconque suite de chiffres.

Nous avons vu aussi que le problème de créer un modèle de document relève du fait que le domaine de l'éducation est vaste et les approches variées. La principale contrainte est donc celle de déterminer si des structures communes existent à un niveau donné d'un scénario pédagogique et, par conséquent, d'éviter qu'un tel modèle puisse interférer avec les diverses techniques didactiques qu'un enseignant utilise pour le design d'objectifs pédagogiques. Autrement dit, il est nécessaire d'identifier les paratextes constituant un document pédagogique numérique afin de constituer un code pédagogique utilisable dans des contextes variés.

Nous avons pris en compte des mécanismes existants tels que :

- l'étude de modèles communs à un curriculum pédagogique, qui consiste à dériver une structure fondamentale d'après les différents grains modulaires (modules, unités, leçons, activités, exercices) ;

¹ Une liste d'agrégateurs RSS se trouve à l'adresse : <http://www.newsonfeeds.com/faq/aggregators>

- l'emploi de la méthode MISA, qui consiste à appliquer des méthodes de l'ingénierie pédagogique pour décrire les blocs d'unités articulant un scénario pédagogique ;
- l'utilisation d'ontologies pédagogiques, qui se divisent couramment en ontologies d'architecture de systèmes d'apprentissage, ontologies de l'instruction et ontologies de domaines ;
- l'emploi de méthodes d'ingénierie de documents, qui analyse les documents existants (numériques ou non) pour décrire une structure fonctionnelle sous format de schéma XML.

La version préliminaire d'un schéma de document pédagogique numérique comprend deux parties principales :

- *métadonnées* : les métadonnées sont des données qui décrivent d'autres données. L'intérêt de structurer des métadonnées était de mieux distinguer l'information pédagogique de l'information administrative visant à donner un contexte clair d'utilisation pour chaque document. Nous avons adopté un format simplifié du standard IEEE LOM pour cette partie ;
- *contenu pédagogique* : le contenu pédagogique est le noyau dur d'un document. Il s'agit des règles syntaxiques et sémantiques appartenant au contenu pédagogique. Cette partie est d'importance majeure car en elle réside la sémantisation cohérente de documents. Nous nous sommes inspirés principalement du modèle LMML pour la définition de sections et sous-sections ainsi que du modèle EML pour la définition de noms de nœuds.

Le modèle de document élémentaire que nous avons développé respecte une approche modulaire d'enseignement conforme à l'Afnor. La démarche suivie nous a conduit à décomposer les unités d'apprentissage, chacune constituée d'un module contenant des séquences pédagogiques. À leur tour, les séquences comprennent des chapitres, des activités et des exercices. Ainsi, il peut être dit qu'un document constitue essentiellement un module.

À son degré le plus atomique, les éléments du document définissent quelques paramètres associés au style de textes, comme para, gras, italique, liste, item, ou rouge. Pour les médias, nous avons associé à une balise générique media des classes pour chaque type d'objet média, y compris les liens hypertextuels.

Dans la pratique, nous avons implémenté ce modèle de document au sein du campus numérique COMOR de l'Université Paris V. La version de *SPECS* installée acceptait une séparation de toutes les composantes d'un DPN et définissait comme règle un maximum de trois séquences pour chaque document. La raison de cette séparation s'explique par un désir de standardisation au niveau de la structuration des documents et au niveau de leur représentation postérieure. En effet, le fait d'avoir trois séquences par module délimite et prévoit un type de navigation simple ainsi que la possibilité de fixer la quantité maximale de sous-parties d'un module tout en sachant qu'à l'intérieur de chaque séquence, il existe d'autres éléments récursifs comme les chapitres, les activités ou les exercices dont le nombre est indéfini.

Bien que nous ayons fait le design d'un modèle de document à partir des méthodes propres à l'enseignement, il est clair qu'un modèle ne peut constituer une totalité et ne peut pas être appliqué à tous les contextes. Nous croyons qu'un modèle, comme un code ou une structure, est formulé pour répondre à des besoins de communication dans la mesure où il facilite et rend évidentes les connexions à l'intérieur d'un système de relations pédagogiques.

Dans ce sens, un document pédagogique numérique permet de percevoir effectivement des relations entre des éléments d'une structure pédagogique. L'avantage de prédéfinir les éléments selon un contexte est donc de prévoir que ces relations seront présentées aux apprenants sous plusieurs formes issues de domaines de l'hypermédia tout en rendant évident les paratextes d'un document. Tout comme le titre d'une section aide à distinguer sa situation par rapport à

l'ensemble d'un document, il est important de montrer les autres paratextes compris par un module pédagogique.

Mais nous ajouterons que la prédéfinition des éléments d'un document pédagogique numérique a d'autres implications tout aussi essentielles. Premièrement, les éléments sont des signes exprimés sous la forme de balises dénotant le type de contenu qu'elles enveloppent. De plus, ces balises sont des composants des objets techniques hypermédias dans le sens où elles remplissent les fonctions de nœuds et de liens. Enfin, elles permettent de faire évoluer l'objet technique hypermédia dans la mesure où leur utilisation est ciblée sur un usage sémantique en laissant de côté le formatage. Ceci permet d'envisager que les utilisateurs, au fur et à mesure qu'ils utiliseront un code pédagogique, pourront détecter des inconsistances ou des ambiguïtés, et proposer de nouvelles versions, chaque fois plus précises, d'un contexte d'application.

Ainsi, une deuxième phase d'implémentation des documents pédagogiques numériques sera atteinte avec la mise à disposition d'un système hypermédia à architecture ouverte supportant la révision et la modification des éléments en tant que service.

Notons à présent une dernière application qui découle de la séparation d'éléments et qui est la transformation d'un document dans d'autres formats. À travers la définition du modèle fondamental, nous réalisons des tests concernant le changement du format d'un document pédagogique numérique. Les transformations de base employées dans nos travaux incluent XML vers PDF, XML vers RDF et XML vers SMIL. Nous ajoutons que ces transformations se produisent par la génération de documents à partir de l'extraction sémantique des éléments et non par des transformations XSLT comme le ferait le *servlet Cocoon*². Par exemple, dans le cas d'une transformation XML vers SMIL, un auteur dispose d'un assistant qui analyse la composition d'un document existant et propose une interface pour adapter le contenu à l'espace de la présentation. Étant donné que SMIL est un langage pour définir des présentations multimédias et que l'objectif est que celles-ci se génèrent à partir d'un document existant, le problème réside dans la quantité du texte et de médias, qui ne peut être la même dans les deux situations. Pour cela, un auteur modifie une instance d'un document et il génère un nouveau document sans transformer le document original.

5. Représentations de documents

La partie de représentation de documents est orientée vers une utilisation par les lecteurs. Il s'agit de l'affichage d'un document distinguant ses éléments constitutifs depuis un navigateur Web conventionnel. Notre objectif était d'offrir une navigation hypermédia simple et conviviale.

Nous avons programmé une feuille de style à l'aide du langage XSL (*Extensible Stylesheet Language*), qui est la recommandation W3C pour le traitement natif des documents XML. Le support de XSL par les navigateurs est souvent négligé ou peu développé, pourtant les bénéfices sont majeurs. D'un côté, il permet de structurer un document en éléments de bloc HTML, constituant le squelette du design et, d'un autre côté, d'ajouter des classes de style aux éléments du squelette qui peuvent s'intégrer à l'aide de CSS (*Cascade Style Sheets*), qui est la recommandation W3C spécifique aux choix stylistiques.

Notre feuille de style standard permet un affichage de type hypertexte navigable [SAL 05], elle est intégrée par défaut dans tous les documents et présente les fonctionnalités suivantes :

- navigation à base d'onglets ;
- séparation de l'information : administrative, organisationnelle, et pédagogique.
- génération automatique de liens hypertextuels à l'intérieur du document pour chaque chapitre, activité et exercice dans chaque séquence du document ;
- génération de liens vers des ressources médias ;
- génération de liens vers des sites Web externes ;

² *Cocoon* est un *servlet* utilisé en parallèle avec le serveur Web Tomcat. Associée à Tomcat, ce *servlet* fournit un environnement de publication à base de documents XML/XSLT.

- génération des métadonnées Dublin Core ;
- sélection de plusieurs vues et styles ;

La figure 4 (a, b, c) montre deux documents du même type de modèle avec une feuille de style adaptée à l'institution éducative.

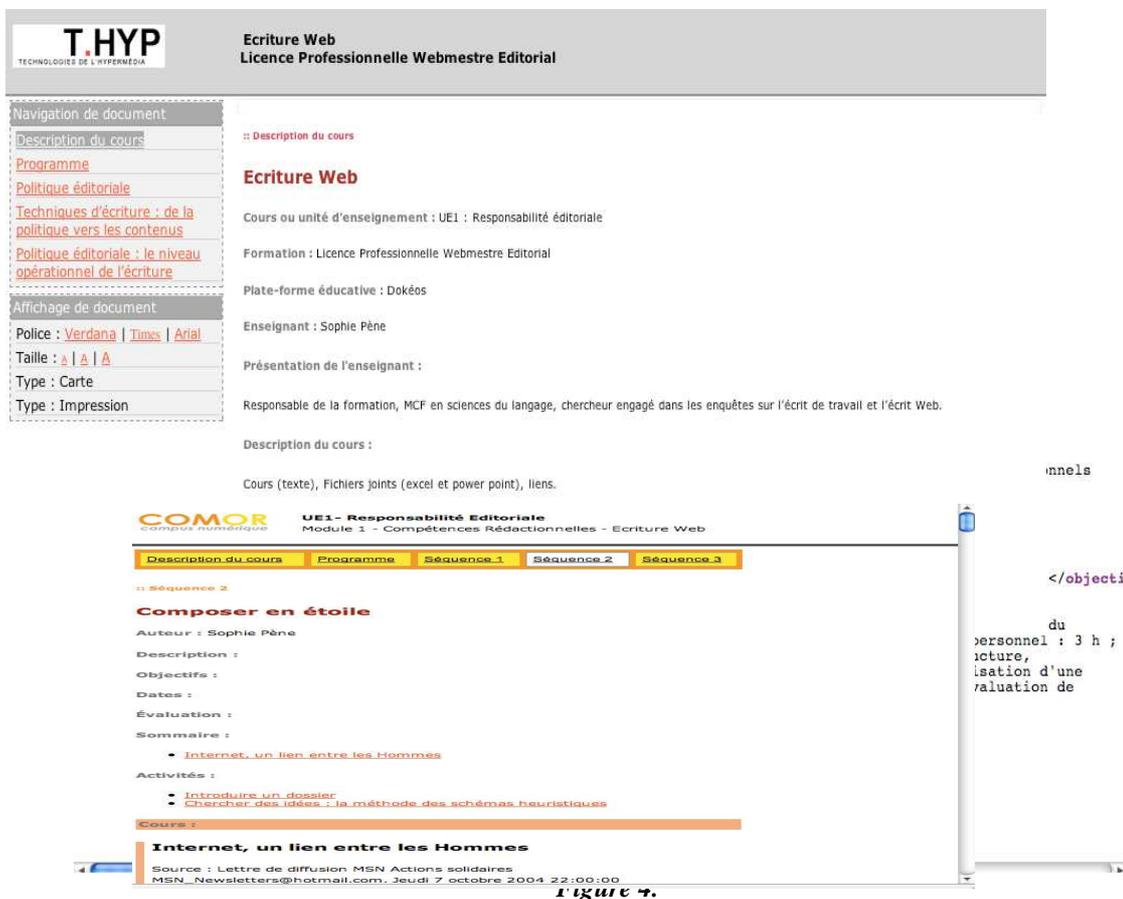


Figure 5. SPECS : affichage de documents

Les classes de style des éléments de bloc se fait par une feuille de style supplémentaire écrit avec le langage CSS. Le style de base s'est inspiré des travaux sur l'utilisabilité, la remédiation et l'ergonomie du Web afin de faciliter la lecture à l'écran.

En suivant les travaux de Norman [NOR 88, 03], Nielsen [NIE 90], Kahn [KAH 98] et André [AND 04], notamment, nous avons opté pour une police Verdana à une taille de 12 points avec un interligné double pour le *texte en volume* (l'ensemble d'un corpus textuel) et de 14 points pour les titres de séquences et de module, le *texte en isolation*. Quant à l'affichage d'un document pour l'impression sur papier, nous avons choisi une police du type Times.

Bien entendu, une implémentation idéale du design de l'information en utilisant des feuilles de style doit supporter la modification et surtout l'ajout d'autres feuilles de style en tant que services. De cette manière, on peut apprécier l'existence d'une galerie non seulement de différentes vues mais aussi de différents domaines des hypermédias. Dans ce contexte, SPECS permet l'importation de nouveaux styles programmés en XSL+CSS et la modification de la feuille de style standard. Récursivement, lorsqu'un nouveau style est créé, il est disponible aux lecteurs. Il s'ajoute à un menu en haut de la page de chaque vue d'un document.

Nous considérons que c'est justement à l'aide d'un design de la représentation de l'information que l'on peut envisager des fonctions secondaires ouvertes des objets techniques hypermédias.

6. L'interface graphique d'auteur

Un composant de l'espace d'auteur qu'il convient d'étudier plus en détail est l'interface graphique d'édition de documents. Cette interface est incluse dans un environnement d'espace de travail sur le Web qui s'inscrit dans une approche collaborative dans la mesure où tous les auteurs peuvent utiliser et réutiliser des documents, des parties de documents, ainsi que les objets médias disponibles dans le répertoire de médias [HOR 00].

L'espace d'édition est ainsi le composant qui permet aux auteurs de rédiger et de modéliser le contenu de documents pédagogiques numériques. Il est divisé en sous-sections correspondant aux parties d'un module : les informations génériques du cours ou métadonnées, les informations pédagogiques et les séquences. Pour l'espace de saisie du texte, nous avons opté pour l'emploi de formulaires Web enrichis par des fonctionnalités interactives pour la modélisation du contenu. Ainsi, un auteur sélectionne les parties de texte à modéliser et en cliquant sur le bouton correspondant à la balise sémantique, le système les insère.

Le design de l'espace d'édition a été conçu dans le but de fournir une interface graphique simple aux auteurs. Dans cette optique, l'analyse de systèmes du chapitre précédent et une méthode de design théorique comme la théorie du management et de la conversation de Winograd et Flores [WIN 86], a constitué la base de notre approche.

Le principal problème que nous avons rencontré a concerné la manipulation de balises et leur représentation. Nous avons vu que dans le cadre de systèmes d'édition fondés sur le formatage, les interfaces d'utilisateur proposent deux solutions : un éditeur WYSIWYG ou des techniques de masquage de balises. Mais si notre intérêt est de promouvoir un usage évolutif de la sémantisation d'éléments d'un document pédagogique numérique et de séparer le fond de la forme, ces deux solutions semblent insuffisantes.

De ce point de vue, nous considérons qu'est importante une certaine évidence des signes linguistiques représentant le nom des balises sémantiques. Autrement dit, il nous semble essentiel d'avoir affaire avec les balises elles-mêmes plutôt qu'avec une autre représentation. Cette entreprise peut paraître déconcertante aux auteurs lors d'une première utilisation. Néanmoins sa pratique appartient à ce que Winograd et Flores classifient comme ruptures ou « *breakdowns* » : « Par *breakdowns* nous entendons les moments d'interruption de notre « être au monde » habituel, standard et confortable » [WIN 86 : 76].

En effet, l'idée de *breakdown* appliquée au design de logiciels explicite que nous, en tant qu'utilisateurs, développons notre rapport avec un outil informatique par de constantes modifications et adaptations de notre comportement avec les composants du *hardware* et *software*. Un exemple peut être l'identification de touches sur un clavier d'ordinateur. Un utilisateur débutant regarde constamment la position de la touche A, par exemple, mais après une certaine période d'utilisation, la pression sur cette touche devient une opération intériorisée qui se réalise presque par instinct, peut-être jusqu'au moment où ce même utilisateur emploie un clavier américain sur lequel la touche Q prend la place de la touche A. De manière analogue, les utilisateurs étant en rapport avec une notation à base de symboles linguistiques et mathématiques pourront s'adapter au fur et à mesure qu'ils développent l'habitude de travailler avec elles et c'est dans ce sens que la notation sémantique aide également à réaliser cette fin.

Dans ce contexte, il est évident que le processus d'édition de documents est sujet au changement, du formatage à la modélisation. Pendant le processus de modélisation, l'auteur insère du contenu et ajoute ensuite les balises correspondantes, au lieu d'insérer du contenu et de lui donner un style. Ce processus est valable aussi bien pour le traitement de textes que pour les médias. En effet, *SPECS* dispose d'une fonctionnalité semi-automatique d'ajout d'objets médias à partir du répertoire de médias ou de l'adresse URL d'une ressource sur le Web [BOU 02].

La version actuelle de notre espace d'édition est le résultat d'un processus de « design participatif » (*participatory design* [WIN 96]) avec le personnel du Campus numérique COMOR de l'Université Paris V. Dans ce sens et en raison du fait que notre système a été installé et utilisé par des auteurs à cette université, nous avons proposé plusieurs maquettes comme solution à l'espace d'édition. Les auteurs utilisaient ces maquettes et discutaient ensuite de leur « utilisabilité ». Le résultat des discussions apportait des corrections à l'interface jusqu'au moment où celle-ci répondait de manière satisfaisant aux attentes.

7. Conclusion

La réalisation pratique de notre approche des « documents pédagogiques numériques » a été mise en place à travers le système *SPECS* qui a la particularité de constituer une application pour leur production dans environnement hypermédia étendu comme le Web.

En nous situant dans un contexte contemporain de l'éducation assisté par les TIC et afin de répondre à certaines des limites posées par les plateformes d'enseignement à distance [HUM 04], nous avons développé l'outil *SPECS* comme un prototype de système d'auteur ayant comme objectif de tester la pertinence d'une approche centrée sur les structures. Il constitue ainsi une alternative à la création de contenus, alternative qui n'est pas fondée sur des principes d'édition basée sur le formatage, mais plutôt sur des stratégies de repérage des éléments constituant un code pédagogique et sur la possibilité d'inventer diverses manières de les représenter en accord avec les codes hypermédiés.

La phase d'implémentation de notre système *SPECS* a été conduite dans un cadre d'utilisation réelle au sein des universités Paris V et VIII. Son utilisation implique de la part des auteurs la modélisation de contenus à partir d'un modèle pédagogique de documents et, d'autre part, la représentation sous format hypertextuel pour les apprenants. En effet, nous avons pris en compte des techniques pour exploiter les différents composants d'un document pédagogique numérique sous des formes issues des domaines des hypermédiés, soit comme éléments indépendants, soit comme ressources imbriquées dans des structures plus larges. « Dès l'instant où la structuration physique des documents reflète un effort de structuration logique de la part de l'auteur et si celui-ci a conscience d'intégrer ses documents au sein d'un ensemble beaucoup plus vaste dont il peut tirer profit au moyen des références croisées, alors l'hypertextualisation automatique révèle qu'un document linéaire, de type technique, possède intrinsèquement des potentialités hypertextuelles » [BAL 96 : 143].

Références

- [AND 04] André, J. et A. Paccoud (2004). « Écrire pour l'écran », in Bélisle, C. (ed.) *La lecture numérique : réalités, enjeux et perspectives*. Villeurbanne : Presses de l'ENSSIB. pp. 105-136.
- [BAL 96] Balpe, J-P., A. Lelu, F. Papy et I. Saleh (1996). *Techniques avancées pour l'hypertexte*. Paris : Hermès.
- [BOU 02] Bouhaï, N. (2002). *Lire, écrire et partager le savoir sur le Web : problème et solutions*. Thèse de doctorat en sciences de l'information et de la communication. Université de Paris VIII.
- [HOR 00] Horton, W. (2000). *Designing Web-based training*. New York : Wiley.
- [HUM 04] Hummel, H., J. Manderveld, C. Tattersall et R. Koper (2004). « Educational modelling language and learning design: new opportunities for instructional re-usability and personalized learning », in *International Journal of Learning Technology*. 1(1). pp. 111-126.
- [KAH 98] Kahn, P. et K. Lenk (1998). « Principles of typography for user interface design », in *Interactions*. 5 (6). Novembre. pp. 15-29.

- [NIE 90] Nielsen, J. (1990). *Hypertexts & hypermedia*. San Diego, CA, États-Unis : Academic Press.
- [NOR 88] Norman, D. (1988). *The psychology of everyday things*. New York : Basic Books.
- [NOR 03] Norman, D. (2003). *Emotional design: why we love (or hate) everyday things*. New York : Basic Books.
- [REY 05a] Reyes, E. et I. Saleh (2004). « Using multimedia in e-learning, a new role for teachers? », in Courtiat, J-P., C. Davarakis, T. Villemur (ed.) *Technology Enhanced Learning: IFIP TC3 Technology Enhanced Learning Workshop (TeL' 04)*. Boston/New York : Springer-Verlag. pp. 183-184.
- [REY 05b] Reyes, E. et I. Saleh (2005). « Modeling learning with hypermedia: taking advantage of digital documents », in *Proceedings of the 16th. Annual Conference on Innovation in Education for Electrical and Information Engineering (EAEEIE'05)*. Lappeenranta, Finlande : Lappeenranta University of Technology. pp. 37-42.
- [REY 06] Reyes, E. (2006). « Meta-writing: re-structuring Web content from a semantic perspective », in *Paris-Utrecht MiniSymposium (PUMS 2)*. Mai 19. Université de Paris VIII.
- [SAL 05] Saleh, I., A. Mkadmi et E. Reyes (2005). « De l'hypertexte à l'hypermédia », in Saleh, I. (ed.) *Les hypermédias : conception et réalisation*. Paris : Hermès Lavoisier, pp. 17-60.
- [WIN 86] Winograd, T. et F. Flores (1986). *Understanding computers and cognition: a new foundation for design*. New York : Addison-Wesley.