

Mémoire sur les Systèmes Auteurs

Nathalie Henry

Février 2004

Table des matières

1	Introduction	2
2	Catégories	3
2.1	Fonctionnalités communes	3
2.2	Les systèmes de séquençement et de planification	4
2.3	Les systèmes à stratégies multiples	4
2.4	Les systèmes de simulation	6
2.5	Les systèmes experts	7
2.6	Les systèmes à multiples représentations des connaissances	7
2.7	Les systèmes à but spécifiques	8
2.8	Les hypermédia adaptatifs	9
3	Le projet ARIADNE	10
3.1	Objectifs	10
3.2	Gestion des contenus pédagogiques	11
4	Les tendances de la recherche	13
4.1	Généricité et Spécificité	13
4.2	Modularité	13
4.3	Réutilisabilité et Interopérabilité	14
4.4	Facilité d'utilisation	14
4.5	Les outils méta-auteur	14
5	Conclusion	15

1 Introduction

Dans un cadre très large, un système auteur est un environnement permettant de créer un logiciel. Il offre diverses fonctionnalités guidant et soutenant l'auteur. Dans le domaine des environnements informatiques/interactifs d'apprentissage humain (EIAH), un système auteur est utilisé pour créer des EIAH.

Les EIAH étant des logiciels extrêmement complexes et diversifiés, les systèmes auteurs se limitent pour la plupart à créer des supports de cours numériques ou bien des tuteurs intelligents.

Systèmes de Tuteurs Intelligents. Un Système de Tuteur Intelligent (STI) est un système d'apprentissage assisté par ordinateur. Un modèle des connaissances représente ce qui est enseigné et des stratégies pédagogiques précisent comment l'enseigner. Ces tuteurs sont donc initialement prévus pour remplacer l'enseignant ou le formateur.

De nos jours, les STI ont un rôle de soutien pour l'enseignant. Ils constituent une aide pédagogique et ne sont pas voués à remplacer la personne humaine. Alors qu'une partie des STI est dédiée aux cours magistraux sur ordinateur (et notamment dans le cadre de la formation à distance), l'autre partie s'oriente plutôt vers les outils de simulation, utilisés en TP.

Un STI est composé de quatre éléments : un modèle de représentation des connaissances à enseigner, un modèle de l'apprenant, un modèle pédagogique et une interface avec l'apprenant. Un système auteur se chargera donc d'aider à la création de ces quatre composants qui seront plus ou moins développés selon le type du STI.

Objectifs. Le système auteur idéal devrait offrir la possibilité à un enseignant de créer simplement et efficacement un tuteur intelligent. Il serait une sorte de mentor intelligent et adaptatif, guidant la création du tuteur mais également proposant diverses aides et outils. Un enseignant sans connaissance préalable du système ou même de l'informatique devrait être apte à l'utiliser pour créer un tuteur intelligent de qualité.

Nous présenterons dans une première partie les différents types de systèmes auteurs catégorisés par T. Murray [Mur99] puis dans une seconde partie, le projet européen ARIADNE. Enfin nous tenterons de synthétiser les tendances actuelles de la recherche dans ce domaine.

2 Catégories

T. Murray classe les systèmes auteurs par rapport aux types de STI que ceux-ci permettent de créer. Il précise [Mur99] qu'il prend en compte des systèmes que leurs créateurs eux-mêmes ne décriraient pas comme STI, il s'agit de systèmes utilisant des techniques de représentation d'IA (réseaux sémantiques...) et d'autres incluant des modèles de contenus ou de stratégies pédagogiques.

Sept catégories ont été dégagées par T. Murray. Après avoir décrit les fonctionnalités globales à tous les systèmes auteurs, nous présenteront successivement chaque catégorie, en précisant les fonctionnalités caractéristiques des systèmes auteurs associés et un ou deux exemple(s) pertinent(s).

2.1 Fonctionnalités communes

Parmi deux douzaines de systèmes auteurs étudiés par Murray, certaines fonctionnalités communes émergent. Certaines autres semblent nécessaires à la robustesse d'un système auteur.

Fonctionnalités basiques. Les fonctionnalités telles que le *copier/coller*, le *trouver* et le *annuler* sont basiques mais doivent être implémentées dans un système auteur. La majorité les possède même si *annuler une action* peut être parfois un problème complexe.

Utiliser des paradigmes familiers. Un principe plus qu'une fonctionnalité mais néanmoins un principe essentiel est le fait d'utiliser des paradigmes familiers aux utilisateurs de logiciels. Effectivement, il s'agit d'avoir un outil ergonomique aidant à la création d'un STI. Il faut que les boutons et/ou menus soient aisément identifiables : autant reprendre les paradigmes classiques (windows...). Il est aussi important de garder des similarités avec les outils de création de contenus pédagogiques traditionnels.

WYSIWYG (What You See Is What You Get) ou le prototypage rapide. Comme dans les environnements de développement rapide (Delphi, Visual C++...), l'utilisateur doit pouvoir rapidement créer et tester son STI. Un rapide aller retour édition-test est souhaitable.

Conception graphique. L'utilisateur des systèmes auteurs, rappelons-le, est censé être un enseignant ou un formateur sans connaissances informatiques préalables. La conception graphique ou visuelle est donc incontournable. Le formalisme est représenté par des icônes ou dessins et il doit être aisément compréhensible et mémorisable. Une vision claire et intuitive de son travail doit être offerte à l'utilisateur.

Primitives haut-niveau. Il est très important d'offrir des primitives *haut-niveau* à un enseignant. Il s'agit par exemple d'avoir une primitive "conseil" ou "explication". Cela facilite l'utilisation du système mais aussi permet de construire rapidement un STI.

Primitives bas-niveau et personnalisation. Alors que les primitives haut-niveau sont incontournables, il est aussi intéressant d'avoir des primitives bas-niveau ou la possibilité minimale de paramétrer. L'outil doit permettre à un utilisateur expérimenté de créer exactement ce qu'il souhaite (écrire un script pour une macro, personnaliser des fonctionnalités). La personnalisation est aussi importante afin d'utiliser au mieux le système auteur et de l'adapter à l'utilisateur.

Fonctionnalités administratives. Enfin, certains systèmes présentent des fonctionnalités administratives permettant d'intégrer dans les STI des analyses statistiques, des analyses de niveaux, la génération de rapports de progression ... utiles pour une exploitation des résultats par les enseignants.

Nous allons présenter successivement les sept catégories de T. Murray. Nous tenterons de dégager les fonctionnalités caractéristiques des systèmes auteurs de chaque catégorie. Cependant, cette liste ne sera en aucun cas exhaustive vu la diversité des systèmes au sein même d'une catégorie.

2.2 Les systèmes de séquençement et de planification

Définition. Cette catégorie de systèmes auteurs permet de créer les STI les plus basiques : les cours sur ordinateur traditionnels. L'outil permet de séquencer et organiser un cours magistral en plusieurs unités d'apprentissage (UA). Le résultat est similaire aux outils auteurs traditionnels, i.e. pour éditer les contenus pédagogiques, mais les STI créés possèdent un modèle des connaissances et un modèle de l'apprenant. Le parcours de l'étudiant est déterminé en fonction de ses performances. Ces systèmes auteurs sont particulièrement utilisés pour créer des outils d'enseignement de connaissances conceptuelles et épisodiques : il s'agit de lire et penser plutôt que d'apprendre en faisant. L'interactivité est par conséquent limitée.

Fonctionnalités de séquençement et d'organisation. Ces systèmes auteurs offrent deux fonctionnalités majeures : séquençement et organisation des UA. Il est possible de définir des objectifs pédagogiques et parfois aussi des conceptions communes (erreurs communes des étudiants). La principale aide à l'enseignant consiste à avoir une vue globale et hiérarchisée des connaissances liées au cours magistral.

Fonctionnalité de création du contenu. Certaines fonctionnalités sont offertes pour avoir des cours visuellement attrayants : insertion de graphiques, d'images ... Il s'agit de faciliter la création de cours numériques pour les enseignants et d'améliorer l'interface avec l'apprenant.

Exemple. Un exemple des outils de cette catégorie cité par T. Murray est DOCENT [DOC]. Cet outil a d'ailleurs engendré une suite de logiciels destinés à la formation sur ordinateur.

2.3 Les systèmes à stratégies multiples

Définition. Par rapport à la catégorie précédente, ces systèmes offrent la possibilité de développer le modèle pédagogique. La difficulté est de définir quelles stratégies pédagogiques seront utilisées, ce sujet suscite un grand débat entre les chercheurs. Faut-il se référer aux didacticiens, aux pédagogues, aux cognitiens, aux enseignants eux-mêmes ? Afin d'implémenter différentes stratégies, les outils auteurs de cette catégorie proposent un panel de fonctionnalités.

Primitives d'actions pédagogiques. La particularité majeure de ces systèmes est d'avoir un ensemble conséquent de primitives d'actions pédagogiques. L'enseignant a donc un panel de *fonctions* lui permettant de définir sa stratégie pédagogique personnelle.

Au macro-niveau, l'enseignant utilisera les fonctionnalités de séquençement et d'organisation identiques à celles de la catégorie précédente permettant de découper le cours en UA.

Au micro-niveau, l'enseignant disposera de fonctionnalités précises : à quel moment et comment donner une explication, un conseil, une analogie, un exemple..., quels types de questions et d'exercices proposer, quels types d'aide et de retour donner à l'étudiant sur ces exercices ...

Stratégies prédéfinies. Certains systèmes permettent de prédéfinir plusieurs stratégies pédagogiques et de les appliquer selon les performances de l'apprenant [Ain00]. Par exemple, il est intéressant de modifier l'aide offerte à l'étudiant selon son niveau et/ou son besoin.

Meta-stratégies. Un système auteur idéal aurait une fonction de mentor, nous l'avons dit auparavant. Il s'agirait dans ce cas de pouvoir conseiller à l'enseignant une stratégie pour un type de connaissance à enseigner particulière. Inutile de préciser que cette tâche est ardue, par conséquent, la notion de *meta-stratégie* est apparue [Mur03a]. Le principe est de définir quelle stratégie sera appliquée pour quel type de connaissance.

L'utilisateur crée alors ses propres meta-stratégies, puisqu'il trop complexe d'implémenter des meta-stratégies de base dans un système auteur. Par contre, une telle extension des possibilités du système entraîne sa complexification et notamment le niveau de connaissances et le travail de l'enseignant.

Exemple. REDEEM, système auteur réalisé par S. Ainsworth et son équipe [AMG⁺03],[Ain00] ne supporte pas la création du contenu mais il permet d'importer du matériel pédagogique existant et de le décrire de façon à concevoir un réseau sémantique du contenu du cours. L'enseignant réutilise donc les supports numériques créés avec d'autres systèmes auteurs (EAO traditionnels). L'enseignant paramètre ensuite ses stratégies pédagogiques en fonction des niveaux des étudiants.

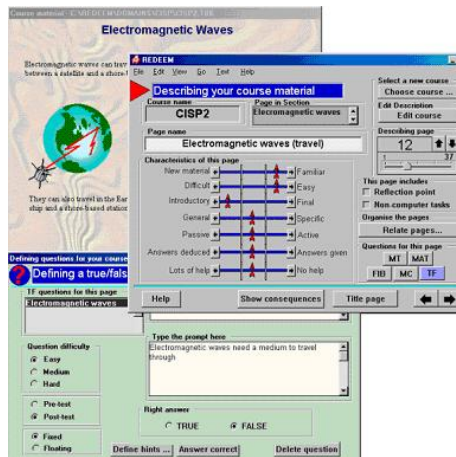


FIG. 1 – Description du matériel pédagogique

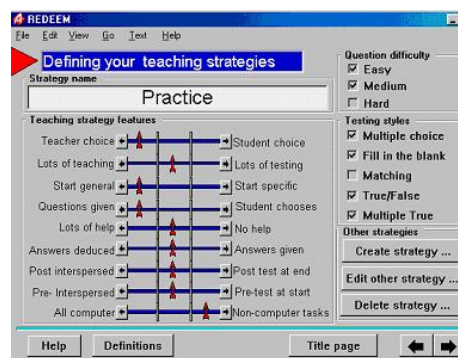


FIG. 2 – Création de stratégies pédagogiques

Eon, système auteur implémenté par T. Murray et son équipe [Mur03a], propose des meta-stratégies afin de choisir la stratégie pédagogique appropriée pour type de connaissance.

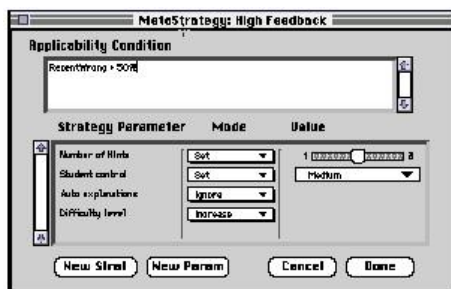


FIG. 3 – Editeur de méta-stratégie

Chaque meta-stratégie a une condition d'application CA et des paramètres P.

Si : CA = "RecentWrong" (variable du modèle de l'apprenant) supérieure à 50% - i.e. *l'étudiant se trompe une fois sur deux* -

Alors : P = nombre de conseils+5, nombre d'exemples+2, ...

2.4 Les systèmes de simulation

Définition. Les systèmes auteurs de cette catégorie permettent de construire un tuteur de simulation, ce qui constitue la difficulté centrale. L'interface est exclusivement graphique pour permettre à l'apprenant de *manipuler* les outils de simulation. En général, des stratégies pédagogiques sont prédéfinies (par niveau) et le modèle de l'apprenant est simple : majoritairement traces de l'utilisation du STI et éventuellement une base des erreurs communes.

Primitives graphiques et objets graphiques. L'interaction entre le système et l'apprenant se fait par la manipulation d'objets graphiques représentant des composants ou des outils. Un panel de fonctionnalités est donc dédié à la création de ces objets graphiques. Les systèmes offrent en général un ensemble de primitives graphiques ainsi que la possibilité de charger des objets et/ou graphiques personnels[Mun03].

Stratégies pédagogiques prédéfinies. Le modèle pédagogique est en général prédéfini dans ce type de systèmes auteurs. Cependant les utilisateurs peuvent contrôler certains paramètres. Il existe dans la plupart des cas deux ou trois niveaux (débutant, expert...) correspondant à des stratégies différentes. Le modèle de l'apprenant, lui, est créé de façon totalement automatique et n'est pas paramétrable.

Outils d'administration - primitives haut-niveau. Diagnostiquer l'apprenant n'est pas chose aisée [Tow03], plusieurs fonctionnalités sont proposées pour aider l'enseignant à récupérer les traces utiles. Plus généralement des primitives de haut niveau sont implémentées. Il est également possible dans certains systèmes [Mun03] de donner la possibilité à un apprenant d'enregistrer une séance et/ou d'en revoir une partie.

Exemples. RIDES est un système auteur créé par A. Munro et son équipe [Mun03]. Il se base essentiellement sur la manipulation d'objets graphiques. La création du tuteur est très guidée et les stratégies pédagogiques prédéfinies. Un point très intéressant pour l'auteur est le fait de pouvoir alterner très facilement édition-test. Il est même possible de modifier des comportements d'objets *on-the-fly*, pendant l'exécution. Une version 3D VIVIDS est en cours de développement.

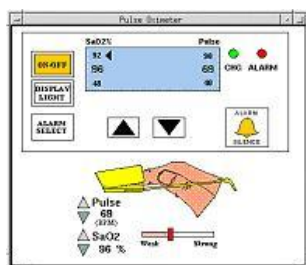


FIG. 4 – Interface créée pour un STI de diagnostic médical

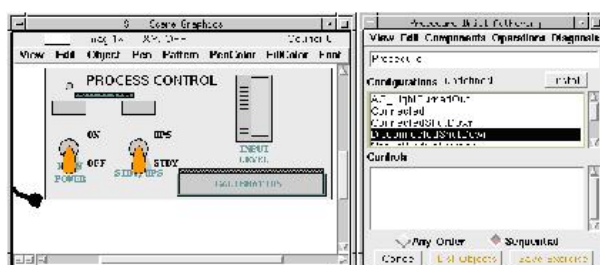


FIG. 5 – Edition d'une procédure de dépannage

SimQuest, créé par W. Joolingen et son équipe [JJ03], est également un système auteur permettant de créer des STI de simulation. Il a été créé environ 20 environnements avec cet outil. L'accent a été particulièrement porté sur l'aide et le soutien apporté à l'utilisateur. Chaque étape lui est présentée et chaque fonctionnalité est décrite. Il y a également un SimQuest Wizard guidant l'enseignant et lui permettant de choisir entre plusieurs didactiques.

2.5 Les systèmes experts

Définition. La construction d'un système expert est très complexe, qu'il s'agisse de le créer manuellement ou automatiquement. Il est en tout cas très difficile de demander à un enseignant non familiarisé avec ce type de système d'utiliser un système auteur pour en créer un. Cependant, quelques systèmes auteurs ont relevé le défi.

Fonctions basiques. La majorité des systèmes auteurs ayant pour but de construire des modèles experts n'offrent pas de fonctions très "intelligentes", elles ne vérifient pas la cohérence de la base de règles. Ce sont de simples fonctions d'édition.

Création de la base de règle automatique. Une fonctionnalité surprenante de Demonstr8 [Ble03] est de construire la base de règles à partir d'exemples fournis par l'enseignant. Le principe est le suivant : après avoir construit l'interface, l'enseignant utilise le STI plusieurs fois en fournissant les entrées et les sorties correspondantes. Le système déduit à partir de ces exemples une base de règles.

Exemples. L'objectif de Demonstr8 [Ble03] est de construire la base de règles par l'exemple, nous l'avons dit. Ce système est au stade de prototype mais un exemple de tuteur extrêmement simple a déjà été créé. Il s'agit du tuteur apprenant la soustraction et l'addition.

L'auteur construit tout d'abord l'interface avec l'apprenant. Une fois les éléments mis en place, il crée un problème puis en cliquant sur "Recording" enregistre sa propre résolution du problème. A partir de plusieurs exemples, Demonstr8 construit la base de règles.

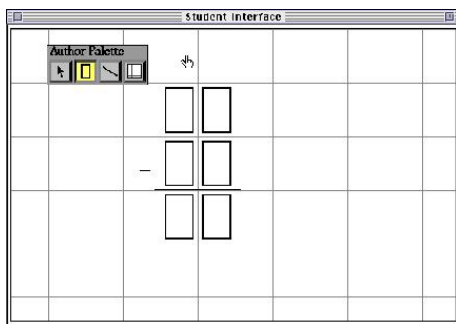


FIG. 6 – Création de l'interface avec l'apprenant

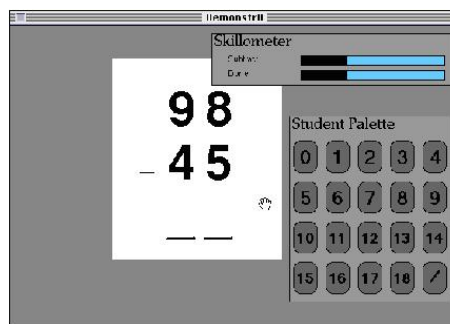


FIG. 7 – Tuteur de soustraction terminé et prêt à l'emploi

2.6 Les systèmes à multiples représentations des connaissances

Définition. L'accent est mis sur la représentation des connaissances. Le système auteur associera les stratégies pédagogiques et le modèle de l'apprenant correspondants. Les systèmes auteurs de cette catégorie sont génériques...

Construction du réseau sémantique. Un réseau sémantique est un ensemble de triplets de la forme (sujet, attribut, objet)[HHW⁺03]. L'auteur crée ce réseau sémantique grâce à diverses fonctionnalités offertes. L'auteur peut ajouter des images, des ressources diverses et des descriptions. Ces informations seront réécrites dans des *templates* pour afficher des questions ou des explications.

Construction automatique du tuteur. Comme nous l'avons évoqué auparavant, le tuteur est construit pratiquement automatiquement une fois que le réseau sémantique est construit. Par exemple, dans Iris [AFFC⁺03], le modèle de l'apprenant, le modèle pédagogique et même le diagnostic de l'apprenant sont des processus génériques et prédéfinis.

Exemples. XAIDA [HHW⁺03], est un système auteur se basant sur un réseau sémantique. il comporte deux modules essentiels : Develop pour construire le système (réseau et interface) et Deliver le module exécutable interagissant avec l'apprenant.

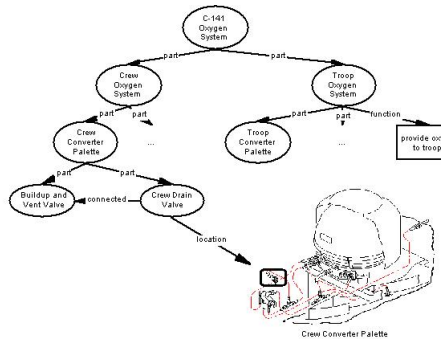


FIG. 8 – Exemple de réseau sémantique

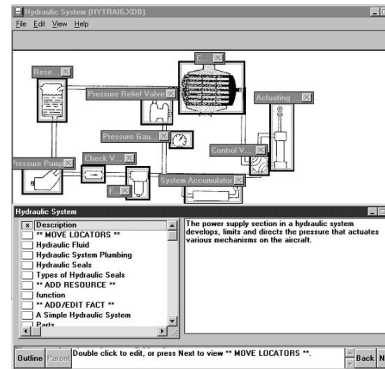


FIG. 9 – Exemple d'interface produite

IRIS [AFFC⁺03] est un système auteur qui guide particulièrement l'enseignant. L'intérêt principal est que l'architecture du tuteur est générée automatiquement à partir d'informations sémantiques sur le domaine, d'informations sur la nature du tuteur désiré et de l'utilisateur final. Reste à évaluer la qualité du tuteur produit ...

2.7 Les systèmes à but spécifiques

Définition. Les systèmes auteurs de cette catégorie sont très spécifiques. Ils servent à créer des STI pour enseigner une tâche ou simuler un logiciel particulier. Le fait de restreindre le cadre de ces systèmes permet de mieux guider l'utilisateur, de créer automatiquement certaines fonctionnalités et donc d'être plus efficaces.

Fonctionnalités spécifiques. L'intérêt de tels systèmes auteurs est d'avoir un cadre restreint. Ainsi des fonctionnalités spécifiques peuvent être mises en place. Par exemple des créations automatiques de modèles de conversations téléphonique dans le cas de LAT [SDMB03]. Des modes de visualisation particuliers peuvent être également offerts à l'utilisateur. Plutôt que de faire une visualisation générique, on optera pour deux ou trois visualisations particulières et pertinentes.

Fonctionnalités de réutilisation. L'objectif de ces systèmes auteurs est un gain en temps et coût de développement. Ils possèdent donc des fonctionnalités de réutilisation de modules préalablement construits. Il s'agit bien sûr de tuteurs construits avec le même système ou de partie de contenu réutilisé dans le même tuteur.

Exemples. LAT, [SDMB03], est un système auteur spécifique servant à créer des STI pour apprendre à utiliser LEAP, un logiciel de telemarketing. Un utilisateur de LEAP répond à ses clients

tout en manipulant la base de données du logiciel. Le système auteur permet de générer des cas d'utilisation fictifs, i.e. des appels téléphonique clients. L'apprenant dispose d'actions définies par l'auteur et enregistre ses réponses. L'auteur crée simplement et efficacement ses tuteurs, il est très guidé et peut réutiliser certains modules (introduction de la conversation client par exemple).

2.8 Les hypermédia adaptatifs

Définition. Les hypermédia adaptatifs sont un peu à l'écart des STI. Il s'agit en fait de contenus destinés à être utilisés online. Le principe est d'utiliser des hypermédia (outils du web) avec des techniques d'IA pour ce qui est de la navigation dans l'hyperespace. C'est pour cette raison que ces hypermédia sont qualifiés d'"intelligents" ou d'"adaptatifs".

Fonctionnalités de conception de l'hyperespace. P. Brusilovsky décrit ce que doivent être les fonctionnalités pour concevoir un hyperespace[Bru03]. Un document hypermédia contient beaucoup de liens et de ressources. Il est nécessaire d'avoir un puissant outil de structuration de l'espace pour ne pas avoir à faire face à une désorientation de l'apprenant.

T. Murray, dans Metalinks [Mur03b], force cette structuration en guidant fortement l'utilisateur. Des contraintes concernant le format des pages sont par exemple nécessaires pour avoir un parcours fluide dans le meta-livre : chaque page comporte une introduction et une conclusion ou une transition.

Fonctionnalités de navigation dans l'hyperespace. Les fonctionnalités de navigation dans l'hyperespace constituent le coeur "intelligent" du système auteur. Les liens sont alors annotés et/ou cachés suivant le parcours de l'apprenant dans l'hyperespace. Le modèle de l'apprenant permet de déterminer la navigation.

Outils de vérification de la cohérence. Plusieurs chercheurs abordent le sujet de la vérification de cohérence [MV01], [Bru00]. Il est important de vérifier la validité du cours à chaque étape. Certains auteurs proposent même l'utilisation d'un système expert pour vérifier cette cohérence.

Réutilisation d'outils existants. La notion de réutilisation d'outils existants est abordé dans plusieurs articles. Dans le cas de WEAR[MV01], il s'agit de réutiliser des pages HTML préalablement réalisées avec des outils du marché. Nous verrons que d'autres systèmes auteurs (projet ARIADNE) utilisent également des outils existants comme XML.

Exemples Metalinks, [Mur03b] est un hypermédia adaptatif réalisé par l'équipe de T. Murray. D'autres exemples ont été développés au cours des dernières années : Interbook ou EadGen[NN02] par exemple.



FIG. 10 – Exemple de page générée avec Metalinks



FIG. 11 – Exemple de page générée avec EadGen

Cette dernière catégorie de systèmes auteurs a fait apparaître la notion de formation à distance mais également la notion de réutilisabilité. Un projet d’envergure concernant particulièrement le Web-Based Learning s’est mis en place en Europe. Il s’agit du projet que nous allons présenter dans la section suivante.

3 Le projet ARIADNE

Le projet européen ARIADNE [ARIa] a pour but d’améliorer la qualité de l’apprentissage grâce au développement d’objets pédagogiques, d’outils et de méthodologies qui permettent une approche ”partage et réutilisation” pour la formation et l’éducation [ARIb].

3.1 Objectifs

Sensibiliser et Mobiliser ARIADNE a pour mission de sensibiliser et de mobiliser la communauté de chercheurs du domaine. Il est important de prendre conscience du fait que le partage et la réutilisation est un principe fondamental. Alors que chaque équipe développe des outils particuliers avec des technologies propres, il est nécessaire d’uniformiser ces recherches.

Pour que le projet fonctionne, il est nécessaire que le plus grand nombre de scientifiques du domaine y adhère. La mobilisation et l’effort commun est source de réussite. De plus, le projet étant européen, il est important que chaque membre de l’Europe soit représenté pour garantir la prise en compte des différences culturelles incluant langue, approche pédagogique et législation.

Réaliser Le champ de recherche d’ARIADNE concerne les meta-données, les outils pédagogiques et l’interopérabilité en général. Dans le cadre de ce projet, un ensemble d’outils ”noyaux” (*core tools*) est en cours de développement. Ayant pour finalité d’être adoptés par la communauté, ces outils se doivent d’être efficaces, stables et faciles d’utilisation. Ils utiliseront éventuellement des standards existants. Le cadre de ce projet est à l’échelle d’une organisation et non de quelques chercheurs isolés, certaines contraintes sont donc à prendre en compte particulièrement pour la numérisation des contenus pédagogiques [BCC⁺02].

Il est tout d’abord nécessaire de numériser en grande quantité, rapidement et à moindre coût afin de susciter l’usage. Les productions devront être stables et perennes. Il s’agit de rentabiliser l’effort humain et financier et de minimiser les bouleversements dans les pratiques.

De plus, il est très important de tenir compte du multi-usage qui peut être fait des objets pédagogiques. Effectivement, les données pédagogiques traditionnelles sont exploitées différemment selon les enseignants, il est important de conserver cette flexibilité pour permettre de passer sans heurts du système traditionnel au système numérique.

Améliorer Il est évident qu’un tel projet n’aboutit pas en une étape. Une démarche d’amélioration continue doit être prise en compte [BCC⁺02]. Les modifications des auteurs ou des technologies doivent être prises en compte aisément et l’intégration de nouveaux outils ou mises à jour doit se réaliser au fur et à mesure du processus d’uniformisation.

Il est nécessaire que le projet débute sur une base saine et claire mais il est aussi important qu’il puisse remettre certaines parties en doute, visant à l’amélioration du projet.

Concrètement, ARIADNE s’intéresse à la problématique de gestion des contenus pédagogiques (Learning Content Management Systems). Nous aborderons les besoins et standards concernant cette problématique dans la section suivante.

3.2 Gestion des contenus pédagogiques

Besoins. Il est nécessaire de prendre en compte quatre étapes dans la gestion des contenus [BCC⁺02].

La première étape est la modélisation. Celle-ci doit évidemment s'adapter à l'existant mais être déclinable et fiable. Il doit également être possible de modifier le modèle sans perdre le contenu modélisé pour faire face aux mise à jour.

La seconde étape est la production à proprement parler. Le problème du format est incontournable, il doit être possible de résister aux évolutions technologiques. Les contenus doivent également dépendre faiblement des supports et usages afin de permettre une hétérogénéité large et leur flexibilité. Il est aussi intéressant de permettre une méthode d'évaluation de leur qualité.

La troisième étape est l'édition. Celle-ci doit se faire rapidement, sur des supports multiples permettant la gestion des contenus multimédia. Ces supports devront également prendre en compte l'interactivité avec l'apprenant qui ne doit pas seulement être passif devant l'objet pédagogique mais bien acteur ou auteur.

Enfin, la dernière étape concerne la gestion. Il doit être évidemment possible de réutiliser les contenus mais aussi de les recombinaison éventuellement et de les contextualiser le cas échéant (permettre la spécialisation de certains contenus).

Standards. Plusieurs standards peuvent répondre à ces besoins. La LOM[v6.a], par exemple, norme de l'IEEE [IEE] uniformise l'indexation des contenus pédagogiques. L'AICC [AIC] développe des techniques d'échange de données et de contenus, IMS [IMS] travaille sur des spécifications techniques pour l'interopérabilité entre systèmes tandis que SCORM[SCO] tente d'intégrer tous les travaux sur le domaine.

LOM. LOM est une norme mise en place par l'IEEE afin de standardiser les contenus pédagogiques. LOM contient les spécifications syntaxique et sémantiques des méta données sur les objets pédagogiques. La norme définit les attributs requis pour pouvoir décrire au mieux un objet ou contenu pédagogique.

Les objectifs de la norme sont les suivants : permettre la caractérisation des objets pédagogiques et par conséquent leur utilisation et leur échange. Cette norme permettra aussi d'identifier et de vérifier la validité des objets pédagogiques circulants. Un des objectifs à terme est d'avoir un standard permettant à toute la communauté de coopérer et de manipuler les mêmes objets.

LOM définit les meta-données associée aux objets pédagogiques par neuf caractéristiques [v6.b].

1. Caractéristique générale : Dans cette catégories sont regroupées les informations générales telles que l'identifiant, le titre, le langage, les mots-clés , la description...
2. Cycle de vie : Cette catégorie permet de décrire le cycle de vie de l'objet, son statut(numéro de version) et ceux qui ont affecté l'objet.
3. Meta-Meta-Données : Cette catégorie permet de décrire qui, où et quand on été saisie les présentes méta-données.
4. Technique : Cette catégorie les techniques requises pour l'utilisation de l'objet pédagogique : est-ce une application java ? un fichier mpeg ? quelle est sa taille ?

5. Pédagogie : Cette catégorie précise les caractéristiques pédagogiques clés de cette ressource : son contexte, son taux d'interactivité, le rôle présumé de l'utilisateur final, ...
6. Droits : Cette catégorie permet de préciser quels sont les droits de la ressource, les conditions d'accès et de modification.
7. Relation : Cette catégorie décrit les liens que l'objet peut avoir avec d'autres objets pédagogiques. (Plusieurs instances "relation" sont réalisables).
8. Annotation : Cette catégorie fournit des informations l'usage pédagogique de la ressource, qui a fait cette annotation et quand. (Evidemment, plusieurs annotations sont possibles).
9. Classification : Enfin, cette catégorie précise où se trouve la ressource dans un système de classement particulier. (Là aussi il peut y avoir plusieurs classifications).

Après avoir défini un standard concernant les objets ou ressources pédagogiques, il reste à concevoir les outils pour créer ces objets ou ressources.

Création des contenus. Un système auteur d'objets pédagogiques est proposé par J-P. David et son équipe [DGF02]. Il s'agit d'une technologie basée sur XML et XSLT, standards W3C [W3C].

L'enseignant remplit un document XML par l'intermédiaire d'une interface simple et efficace : GenDoc [Dav]. Les Document Type Definition définissent l'ensemble des balises à utiliser. Ces balises sont transparentes pour l'utilisateur qui n'as pas à les écrire mais simplement à cliquer sur un arbre des éléments XML.

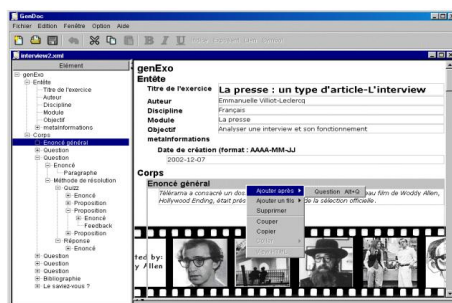


FIG. 12 – Capture d'écran de l'interface GenDoc

L'équipe de J-P. David a également développé un outil pour réaliser un scénario, ou une séance de travail en utilisant des objets pédagogiques : AnalyDoc [Dav].

Concernant la production "industrialisée" de supports pédagogiques numériques, la chaîne éditoriale SCENARI [SCE] dans le cadre du projet CHAPERON[CHA] a été développée. Cette chaîne éditoriale est finalisée et peut répondre à un certain nombre de besoins explicités plus haut.

Pour le point particulier concernant l'édition de contenus multimédia temps réel, des outils auteurs sont proposés succinctement[BCR02]. Il s'agit d'outils permettant d'encoder et transmettre des données pour un coût très faible. Un outil de videoconference a également été développé.

4 Les tendances de la recherche

Evoquons les grandes tendances de la recherche sur les systèmes auteurs.

4.1 Généricité et Spécificité

Les outils auteurs génériques offrent des possibilités très larges, en étant en premier lieu non liés à un contexte particulier. L'utilisateur est laissé libre et un ensemble conséquent de fonctionnalités lui est offert. La contrepartie est que l'outil n'est pas forcément adapté à la création de certains tuteurs et certaines tâches ne sont pas automatisées. De plus, le niveau de connaissances pour utiliser le système n'est pas négligeable. Cependant, la recherche s'oriente vers le système auteur mentor guidant l'utilisateur.

Les adeptes du spécifique démontrent qu'un système auteur spécifique est efficace. Effectivement, l'utilisateur est guidé, les tâches et fonctionnalités proposées sont adaptées. Par contre, le coût de développement d'un tel système est-il avantageux par rapport au développement manuel d'un tuteur ?

Plusieurs idées émergent sur ces sujets.

Murray propose avec Eon [Mur99] un double niveau de système auteur. Eon est un outils générique, qui, paramétré par un expert devient un outil auteur spécifique prêt à être utilisé par un enseignant ou un formateur. Nous évoquerons cette notion de *meta-authoring* dans une section suivante.

La seconde idée intéressante concerne les systèmes auteurs basés sur les composants [RBW03] [AR1a]. Effectivement, beaucoup de systèmes auteurs ont été développés dans des équipes différentes pour des besoins différents [BCC⁺02]. Chacun est plus ou moins efficace pour la création d'une partie ou d'un composant de STI. L'idée est alors de combiner le meilleur de chaque système, en créant des outils auteurs modulaires.

4.2 Modularité

En programmation, cette notion est courante. Effectivement, il est intéressant de programmer des briques logicielles indépendantes et de pouvoir les réutiliser, ou en changer certaines. L'intérêt pour un système auteur est évident. Un module pour chaque composant du STI permettrait de faire évoluer le système.

En allant plus loin, il pourrait être encore plus intéressant de combiner des modules ou composants de divers systèmes auteurs. Le problème, dans ce cas, serait d'uniformiser, d'adopter une norme ou un standard commun pour le développement des outils auteurs.

Le premier pas a été effectué avec le projet ARIADNE. Effectivement, certains chercheurs se sont concentré sur des briques de systèmes auteurs, comme par exemple les auteurs de contenus multimédia [BCR02].

Uniformiser les objets pédagogiques permet une interopérabilité des systèmes et une réutilisabilité des travaux antérieurs, c'est un travail ardu entamé par plusieurs organismes tels que l'IEEE [IEE], le CEN/ISSS[CEN], l'AICC[AIC] et ADL (avec SCORM [SCO])...

4.3 Réutilisabilité et Interopérabilité

Ces deux idées sont fondamentales pour l'utilisateur. Les outils auteurs, comme leurs noms l'indiquent, ne sont "que" des outils pour l'enseignant ou le formateur. Ils doivent permettre un gain de temps, de coût et de difficulté. Le fait d'uniformiser le format des contenus pédagogiques permet à l'enseignant de garder ses précédents travaux de numérisation. La création se fera une fois pour toutes et ces objets seront utilisables dans plusieurs systèmes.

L'interopérabilité est également une notion importante mais beaucoup plus difficile à mettre en place du fait que les équipes de recherche travaillent dans des laboratoires séparés. IMS et ADL(SCORM) travaillent sur ce sujet. Une interopérabilité entre systèmes demande un effort commun : qu'il s'agisse d'adopter une norme ou de créer des outils de conversion pour intégrer des produits d'autres systèmes [PER].

A un niveau moindre, la collaboration peut aussi s'envisager entre créateurs de STI sur un seul et même système auteur. Certaines recherches s'orientent vers des systèmes auteurs utilisables à distance [TDZ01]. Il s'agit d'avoir une plateforme qui permet à plusieurs auteurs de créer un STI et ce, de façon synchrone ou asynchrone. Le développement concerne principalement des outils ou fonctionnalités supplémentaires de communication et de collaboration pour des utilisateurs distants.

Faciliter la vie de l'enseignant est un des objectifs prioritaires. Les systèmes auteurs en plus d'être réutilisables et interopérables doivent être simples d'utilisation.

4.4 Facilité d'utilisation

Un outil utilisable efficacement à la première prise en main et par n'importe quel enseignant [BCC⁺02] : voici un objectif difficile à atteindre, certains le considèrent même comme inatteignable pour ce qui est des outils génériques [Mur03a].

Le principe premier pour faciliter l'utilisation est d'opter pour une interface graphique et utiliser des paradigmes familiers (windows). L'ergonomie de l'interface est donc un point essentiel sur la prise en main mais pas uniquement.

Il est nécessaire de faciliter le passage du manuel au logiciel, en utilisant une métaphore familière et intuitive (storyboard, langage de l'éducation...). Offrir dans un premier temps un nombre limitées de fonctionnalités pour guider l'utilisateur puis lui offrir des fonctionnalités supplémentaires. Il est aussi possible de guider séquentiellement la création.

Cependant, il est aussi important soulignent certains chercheurs [Mur99] de permettre à un utilisateur expérimenté de s'appropriier l'outil. En offrant, par exemple, des raccourcis claviers, une liberté de création (approche objectifs globaux-écrans apprenants ou l'inverse), des fonctionnalités programmables (macros, scripts), la facilité d'utilisation au quotidien pour un utilisateur familier du logiciel est prise en compte.

Pour faciliter l'utilisation des systèmes, l'accent est aussi porté sur les aides ou compagnons apportés à l'utilisateur. Pourquoi ne pas envisager un STI pour l'apprentissage du système auteur ?!

4.5 Les outils méta-auteur

Une idée portée par plusieurs chercheurs [Mur03a], [BM02], concerne les outils méta-auteurs : créateurs ou paramétreurs d'outils auteurs.

L'idée de Murray est un système auteur très générique qui permettrait à un utilisateur expert de créer une ontologie correspondant aux besoins d'un enseignant ou formateur. Une ontologie de domaine est un ensemble de mot de vocabulaire concernant un domaine. Une fois l'ontologie créée spécifiquement pour un enseignant et le système auteur paramétré par l'expert, le système serait utilisable par l'enseignant lui-même. Ainsi, l'enseignant dispose d'un outils adapté à ses besoins.

Mizoguchi et son équipe travaillent depuis plusieurs années sur les ontologies [Miz92]. Ils proposent [MB00] d'utiliser une ontologie spécifique comme standard dans le domaine. Ainsi, une étape serait franchie vers l'uniformisation et l'interopérabilité des systèmes auteurs. L'intérêt d'une ontologie est principalement d'explicitier les concepts généralement sous-jacents : il s'agit donc d'un meta-modèle.

Mizoguchi fait référence à XML comme une ontologie niveau 1 [MB00]. Il cite le premier pas de l'IEEE LTSC Committee avec LOM et affirme que si l'établissement d'un standard demande un effort commun considérable, il devrait être récompensé.

5 Conclusion

Les systèmes auteurs : est-ce une réalité ou une fiction ?

Certains systèmes auteurs sont commercialisés et utilisés, surtout dans le cadre de la formation à distance ou de la formation en entreprise. Le projet ARIADNE vise à rendre les systèmes auteurs interopérables et de ce fait, utilisés par un grand nombre de personnes.

Ce projet est toute fois à l'état de développement. Certaines universités européennes diffusent des cours en ligne utilisant les technologies ARIADNE mais le principe n'est pas encore au point.

Concernant l'évaluation des systèmes auteurs et surtout des tuteurs ou cours qu'ils permettent de créer, aucune méthodologie n'a été réellement spécifiée. Il est donc très difficile d'avoir une idée des performances des divers systèmes. Cependant, certains projets ambitieux (ARIADNE mais aussi Demonstr8 et Eon) semblent prometteurs.

Pour conclure, l'utilisation des outils auteurs par les enseignants pour créer des cours de tous les jours n'est pas encore pour demain. Cependant, la formation à distance et la standardisation des contenus et méthodes pédagogiques est en voie de développement.

Références

- [AFFC⁺03] A. Arruarte, B. Ferrero, I. Fernandez-Castro, M. Urretavizcaya, A. Alvarez, and J. Greer. The iris authoring tool. In T. Murray, S. Blessing, and S. Ainsworth, editors, *Authoring Tools for Advanced Technology Learning Environments*. Kluwer Academic Publisher, 2003.
- [AIC] AICC. Available on : <http://www.aicc.org>.
- [Ain00] S. Ainsworth. Redeem : Its authoring tools and human teaching. In *the ITS2000 workshop on Modelling Human Teaching Tactics*, 2000.
- [AMG⁺03] S. Ainsworth, N. Major, S. Grimshaw, M. Hayes, J. Underwood, B. Williams, and D. Wood. Redeem : Simple intelligent tutoring systems from usable tools. In T. Murray, S. Blessing, and S. Ainsworth, editors, *Authoring Tools for Advanced Technology Learning Environments*. Kluwer Academic Publisher, 2003.
- [ARiA] ARIADNE. Available on : <http://www.ariadne-eu.org/>.
- [ARiB] ARIADNE. Ariadne strategy paper. Available on : <http://www.ariadne-eu.org/en/about/general/Ariadne%20Strategy%20Paper.html>.
- [BCC⁺02] B. Bachimont, I. Cailleau, S. Crozat, M. Majada, and S. Spinelli. Outils auteurs : approche industrielle versus approche artisanale. In *Proceedings of the TICE2002 conference, session ARIADNE*, 2002.
- [BCR02] G. Bailo, M. Chirico, and M. Raggio. Authoring tools for real time creation and distribution of embedded multimedia content. In *Proceedings of the TICE2002 conference, session ARIADNE*, 2002.
- [Ble03] S. Blessing. A programming by demonstration authoring tool for model-tracing tutors. In T. Murray, S. Blessing, and S. Ainsworth, editors, *Authoring Tools for Advanced Technology Learning Environments*. Kluwer Academic Publisher, 2003.
- [BM02] J. Bourdeau and R. Mizoguchi. Collaborative ontological engineering of instructional design knowledge for an its authoring environment. In *Proceedings of the AIED conference*, 2002.
- [Bru00] P. Brusilovsky. Course sequencing for static courses? applying its techniques in large-scale web-based education. In *Proceedings of the ITS2000 conference*, 2000.
- [Bru03] P. Brusilovsky. Developing adaptive educational hypermedia systems : From design models to authoring tools. In T. Murray, S. Blessing, and S. Ainsworth, editors, *Authoring Tools for Advanced Technology Learning Environments*. Kluwer Academic Publisher, 2003.
- [CEN] CEN/ISSS. Available on : <http://www.cenorm.be/cenorm/businessdomains/businessdomains/informatio>
- [CHA] CHAPERON. Available on : <http://www.ina.fr/recherche/projets/encours/chaperon.fr.html>.
- [Dav] JP. David. Gendoc et analydoc. Available on : <http://a.flament.free.fr/data/presentationH2PTM.ppt>.
- [DGF02] JP. David, C. Guilloux, and A. Flament. A learning objects generator with xml-xslt technology. In *Proceedings of the TICE2002 conference, session ARIADNE*, 2002.
- [DOC] DOCENT. Available on : <http://www.docent.com>.
- [HHW⁺03] H. Half, P. Hsieh, B. Wenzel, T. Chudanov, M. Dirnberger, E. Gibson, and C. Redfield. Requiem for a development system : Reflections on knowledge-based, generative instruction. In T. Murray, S. Blessing, and S. Ainsworth, editors, *Authoring Tools for Advanced Technology Learning Environments*. Kluwer Academic Publisher, 2003.

- [IEE] IEEE. Available on : <http://www.ieee.org>.
- [IMS] IMS. Available on : <http://www.imsglobal.org>.
- [JJ03] W. Joolingen and T. Jong. Simquest : Authoring educational simulations. In T. Murray, S. Blessing, and S. Ainsworth, editors, *Authoring Tools for Advanced Technology Learning Environments*. Kluwer Academic Publisher, 2003.
- [MB00] R. Mizoguchi and J. Bourdeau. Using ontological engineering to overcome common ai-ed problems. *International Journal of Artificial Intelligence in Education*, 11 :107–121, 2000.
- [Miz92] R. Mizoguchi. Task ontology and its use in a task analysis interview system : Two level mediating representation in multis. In *Proceedings of the JKAW92 conference*, 1992.
- [Mun03] A. Munro. Authoring simulation-centered learning environments with rides and vividis. In T. Murray, S. Blessing, and S. Ainsworth, editors, *Authoring Tools for Advanced Technology Learning Environments*. Kluwer Academic Publisher, 2003.
- [Mur99] T. Murray. Authoring intelligent tutoring systems : an analysis of the state of the art. In *Proceedings of the AIED conference*, 1999.
- [Mur03a] T. Murray. Eon : Authoring tools for content, instructional strategy, student model and interface design. In T. Murray, S. Blessing, and S. Ainsworth, editors, *Authoring Tools for Advanced Technology Learning Environments*. Kluwer Academic Publisher, 2003.
- [Mur03b] T. Murray. Metalinks : Authoring and affordances for conceptual end narrative flow in adaptative hyperbooks. In *Proceedings of the AIED conference*, 2003.
- [MV01] M. Moundridou and M. Virvou. Authrogin and delivering adaptative web-based textbooks using wear. In *Proceedings of the AIED conference*, 2001.
- [NN02] M. Nanard and J. Nanard. Eadgen, un environnement ouvert de production en ligne basé sur xml et xslt. In *Proceedings of the TICE2002 conference*, 2002.
- [PER] PERLEA. module tourbillon. Available on : <http://bat710.univ-lyon1.fr/sdaubias/p-perlea/>.
- [RBW03] S. Ritter, S. Blessing, and L. Wheeler. Authoring tools for component-based learning environments. In T. Murray, S. Blessing, and S. Ainsworth, editors, *Authoring Tools for Advanced Technology Learning Environments*. Kluwer Academic Publisher, 2003.
- [SCE] SCENARI. Available on : <http://wwwspul.utc.fr/ics/index.html>.
- [SCO] SCORM. Available on : <http://www.adlnet.org/index.cfm?fuseaction=AboutSCORM>.
- [SDMB03] R. Sparks, S. Dooley, L. Meiskey, and R. Blumenthal. The leap authoring tool : Supporting complex courseware authoring through reuse, rapid prototyping and interactive visualizations. In T. Murray, S. Blessing, and S. Ainsworth, editors, *Authoring Tools for Advanced Technology Learning Environments*. Kluwer Academic Publisher, 2003.
- [TDZ01] S. Tahli, M. Djoudi, and A. Zidani. Un système auteur de tuteurs intelligents. In *EIAO*, 2001.
- [Tow03] D. Towne. Automates knowledge acquisition for intelligent support of diagnostic reasoning. In T. Murray, S. Blessing, and S. Ainsworth, editors, *Authoring Tools for Advanced Technology Learning Environments*. Kluwer Academic Publisher, 2003.
- [v6.a] LOM v6.3. Available on : <http://ltsc.ieee.org/wg12/index.html>.
- [v6.b] LOM v6.3. Description de la norme. Available on : <http://www.univ-lemans.fr/acorb/LOM.pdf>.
- [W3C] W3C. Available on : <http://www.w3c.org>.