

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université Hadj Lakhdar de Batna
Faculté des sciences de l'ingénieur
Département d'Informatique

Intégration des technologies de coopération et d'intelligence dans les environnements d'apprentissage à distance

Thèse

En vue de l'obtention du diplôme de
Doctorat en Sciences en Informatique

(Option: Intelligence Artificielle et Génie Logiciel)

Soutenue publiquement le 03/03/2007 par :

Saïd TALHI

Devant le jury composé de :

Président	N. Bouguechal Prof. Université de Batna
Examineurs	A. Benyettou Prof. Université d'Oran (USTO)
	S. Chikhi MC. Université de Constantine
	A. Zidani MC. Université de Batna
Rapporteur	M. Djoudi MC. Université de Poitiers (France)
Co-Rapporteur	M. Batouche Prof. Université de Constantine

Résumé

Les logiciels éducatifs réalisés dans les deux contextes de l'EAO et l'EIAO classiques sont tous conçus pour fonctionner en autonomie et en mono-usager. Si ces environnements faisaient généralement référence à un individu -l'apprenant- face à un ordinateur dans un lieu donné à un moment donné, il faut aujourd'hui, prendre en considération des dimensions nouvelles (temporelles, spatiales et sociales) qui viennent bouleverser cette situation.

En effet, les possibilités des technologies d'information et de communication apparues ces dernières années, notamment le réseau mondial Internet et le Web, et les nouveaux besoins d'apprentissage (apprentissage à distance, apprentissage tout au long de la vie, apprentissage coopératif assisté par ordinateur, etc.) modifient considérablement les caractéristiques des environnements d'apprentissage et les questions qui s'en dégagent. Un nouveau moyen d'appréhender ces environnements est alors de les considérer comme des environnements dans lesquels coopèrent des agents humains et des agents artificiels.

Les travaux de recherche que nous menons dans cette thèse participent à cette mutation. Ils contribuent à proposer de nouvelles situations d'apprentissage en prenant en compte les aspects Distance et Coopération entre apprenants et auteurs/tuteurs. La plate-forme Ibn Sina que nous développons dans ce contexte permet en effet de gérer un apprentissage coopératif à distance intégrant plusieurs fonctionnalités dont : l'interaction de groupe et l'édition coopérative de documents, l'assistance à la navigation, l'assistance à l'apprenant par tuteur humain et enfin l'assistance à l'apprenant par tuteur informatique et la réalisation coopérative de ces tuteurs qui fait l'objet de cette thèse. Ce sont ces facteurs d'assistance et de collaboration qui nous paraissent essentiels dans une plate-forme d'apprentissage à distance pour palier le problème de la sensation d'isolement constatée souvent chez l'apprenant en situation d'apprentissage à distance.

Cette thèse traite donc de la problématique de l'assistance par tuteur informatique en situation d'apprentissage à distance. Nous présentons à cet effet un système tuteur intelligent basé sur la technologie hypermédia. Construit sur la base d'une architecture client-serveur, le tuteur consiste en un canevas générique que les auteurs instancient à travers un environnement auteur coopératif. Grâce aux technologies hypermédias et systèmes à base de connaissances adoptés dans sa conception, ce système permet une adaptation dynamique de l'activité d'apprentissage, et ce, en se basant sur trois modèles : un modèle pédagogique, un modèle de documents et un modèle de l'apprenant.

Mots-clés : Systèmes tuteurs intelligents hypermédias, systèmes hypermédias adaptatifs, pédagogie par objectifs, objets pédagogiques, système auteur coopératif, travail coopératif assisté par ordinateur, collecticiel, groupware, architecture client/serveur, world wide web (WWW).

Remerciements

Toute thèse est le résultat de très nombreuses collaborations, tant au niveau scientifique, qu'au niveau humain. Il est très difficile de toutes les mentionner. Je suis persuadé que les personnes omises ne m'en voudront pas trop de les avoir oubliées dans l'énumération qui suit: ce n'est en rien délibéré. L'essentiel restera toujours d'avoir la chance d'être ensemble pour de nouveaux travaux en devenir. Je remercie en particulier:

Mon directeur de thèse Monsieur **Mahieddine DJOUDI**, Maître de conférence à l'Université de Poitiers (France), et mon co-directeur de thèse Monsieur **Mohamed BATOUCHE**, Professeur à l'Université Mentouri de Constantine, pour m'avoir dirigé tout au long de cette thèse. J'ai beaucoup bénéficié de leurs conseils, de la documentation qu'ils m'ont fournie et de leurs suggestions pertinentes pour la mise au point de ce travail.

Monsieur **Noureddine BOUGUECHAL**, Professeur à l'Université de Batna, qui m'a fait l'honneur de présider le jury de cette thèse malgré ses nombreuses obligations.

Monsieur **Abdelkader BENYETTOU**, Professeur à l'USTO d'Oran, qui m'a fait l'honneur de faire un long chemin pour venir évaluer cette thèse.

Monsieur **Salim CHIKHI**, Maître de Conférence à l'Université Mentouri de Constantine, qui a accepté de faire un long chemin pour venir évaluer mon travail.

Monsieur **Abdelmadjid ZIDANI**, Maître de conférence à l'Université de Batna, qui a accepté d'évaluer mon travail malgré ses nombreuses obligations.

Monsieur **Rachid Ahmed Ouamer**, Maître de conférence à l'Université de Tizi Ouzou, qui m'a fait l'honneur d'accepter de faire participer au jury pour évaluer cette thèse.

Toute personne ayant aidé de près ou de loin à la réalisation de ce travail.

Que cette thèse soit un témoignage de ma gratitude envers mes parents et tous les membres de ma famille qui m'ont apporté leur soutien moral tout au long de cette thèse !

Je ne saurais terminer ces remerciements sans un énorme merci à Charah qui, partageant ma vie, a supporté avec patience et abnégation mes nombreuses absences notamment durant la phase finale de rédaction.

Un « pardon » à ma petite Belkis Amira pour avoir failli à mon rôle de père pendant ces derniers mois.

A tous un sincère et chaleureux merci !

Cette thèse est dédiée :

A la mémoire de mon père Ahcène, sans son sacrifice je ne serai pas là où je suis ;

A ma mère Zouina, sans son sacrifice je ne serai pas là où je suis ;

A ma femme Salima ;

A ma petite Belkis-Amira ;

A toute ma famille ;

A tous mes amis et mes collègues.

*La nuit durait et le jour était long à préparer son entrée
Les étoiles se plaignaient de cette attente prolongée
Enfin le vent du matin chassa le sombre voile
Et les jardins d'alentour montèrent de suaves parfums
A l'Est rougissante et toute intimidée
L'aurore montra ses joues humides de rosée
Dans l'espace céleste la nuit se mit à fuir d'étoile en étoile
Et comme les feuilles d'un arbre celles-ci tombèrent une à une
Enfin le soleil apparut dans son éclat resplendissant
Et reculant devant le jour, la nuit s'évanouit*

Ibn Charaf

Table Des Matières

CHAPITRE 1	14
INTRODUCTION GENERALE	14
1.1 PRELIMINAIRES.....	14
1.2 CONTEXTE DE TRAVAIL	15
1.3 MOTIVATIONS	17
1.3.1 <i>Guidage de l'apprenant à distance par tuteurs intelligents</i>	17
1.3.2 <i>Réalisation coopérative de STI</i>	19
1.4 DOMAINES DE RECHERCHE	21
1.5 OBJECTIFS	22
1.5.1 <i>Indépendance de la matière d'enseignement</i>	22
1.5.2 <i>Modèle Pédagogique réutilisable</i>	22
1.5.3 <i>Indépendance de la plate-forme matérielle</i>	23
1.5.4 <i>Proposition d'une architecture pour le mode apprenant</i>	23
1.5.5 <i>Proposition d'une architecture pour le mode auteur</i>	24
1.6 PLAN DE LA THESE.....	25
CHAPITRE 2	26
SYSTEMES HYPERMEDIAS ET TUTEURS INTELLIGENTS POUR L'ENSEIGNEMENT A DISTANCE SUR LE WEB	26
2.1 INTRODUCTION	26
2.2 PROBLEMATIQUE DU GUIDAGE EN ENSEIGNEMENT A DISTANCE.....	28
2.2.1 <i>Etablissements d'enseignement à distance</i>	28
2.2.2 <i>Plates-formes d'enseignement à distance</i>	29
2.2.3 <i>Modalités de guidage de l'apprenant à distance</i>	31

2.3 SYSTEMES CLASSIQUES D'ENSEIGNEMENT MEDIATISE PAR ORDINATEUR	36
2.3.1 <i>Micro-mondes</i>	37
2.3.2 <i>Systèmes Tuteurs Intelligents</i>	39
2.4 SYSTEMES HYPERMEDIAS EDUCATIFS	50
2.4.1 <i>Hypertexte, Hypermédia, Multimédia</i>	50
2.4.2 <i>Hypermédias et Enseignement/Apprentissage</i>	54
2.4.3 <i>Hypermédias adaptatifs et enseignement/apprentissage</i>	59
2.4.4 <i>Hypermédias adaptatifs dynamiques</i>	62
2.5 INTEGRATION DES TECHNOLOGIES DES HYPERMEDIAS ET DES TUTEURS INTELLIGENTS	63
2.5.1 <i>Planification d'enchaînement</i>	64
2.5.2 <i>Analyse intelligente des solutions de l'apprenant</i>	64
2.5.3 <i>Support interactif de résolution de problèmes</i>	64
2.5.4 <i>Support adaptatif de collaboration</i>	65
2.6 EXEMPLES DE STI SUR LE WEB	65
2.6.1 <i>Le Système CALAT</i>	65
2.6.2 <i>Le système ADIS</i>	67
2.7 CONCLUSION	68
CHAPITRE 3	70
SYSTEMES AUTEURS – DE L'AUTONOMIE A L'OUVERTURE SUR LE WEB	70
3.1 INTRODUCTION	70
3.2 DEFINITIONS D'UN SYSTEME AUTEUR	71
3.3 OBJECTIFS D'UN SYSTEME AUTEUR.....	72
3.4 OUTILS DE DEVELOPPEMENT D'APPLICATIONS EDUCATIVES MULTIMEDIA.....	73
3.4.1 <i>Outils de création monomédia</i>	74
3.4.2 <i>Systèmes auteurs : outils d'intégration multimédia</i>	76
3.5 TAXINOMIE DES SYSTEMES AUTEURS.....	79
3.5.1 <i>La métaphore de la carte</i>	79
3.5.2 <i>La métaphore de la ligne du temps</i>	80
3.5.3 <i>Les diagrammes d'icônes</i>	81
3.5.4 <i>Les systèmes hypertextes améliorés</i>	81
3.5.5 <i>Systèmes généralistes et systèmes spécialisés</i>	82
3.5.6 <i>Portabilité</i>	82

3.6 CRITERES DE SELECTION D'UN SYSTEME AUTEUR.....	83
3.6.1 Questions d'ordre logistique	83
3.6.2 Questions d'ordre fonctionnel.....	84
3.6.3 Questions d'ordre technique	84
3.7 REALISATION D'UN LOGICIEL EDUCATIF A PARTIR D'UN SYSTEME AUTEUR	85
3.8 SYSTEMES AUTEURS DE L'EAO CLASSIQUE	88
3.8.1 Préliminaires	88
3.8.2 Principaux systèmes auteurs sur gros systèmes	89
3.8.3 Principaux systèmes auteurs sur micro-ordinateurs.....	89
3.9 SYSTEMES AUTEURS DE TUTEURS INTELLIGENTS CLASSIQUES	90
3.9.1 Préliminaires	90
3.9.2 Exemples de systèmes auteurs	92
3.9.3 EDDI : Un environnement basé sur une expertise du domaine	95
3.10 LES SYSTEMES AUTEURS SUR LE WEB	96
3.10.1 Préliminaires	96
3.10.2 Aspects pédagogiques.....	98
3.10.3 Aspects de gestion des apprenants.	99
3.10.4 Aspects de standardisation	100
3.11 CONCLUSION	102
CHAPITRE 4	104
SPECIFICATION DES BESOINS ET MODELE CONCEPTUEL	104
4.1 INTRODUCTION	104
4.2 UNE PEDAGOGIE REUTILISABLE : LA PEDAGOGIE PAR OBJECTIFS	105
4.2.1 Qu'est-ce qu'un objectif pédagogique ?.....	105
<i>L'objectif opérationnel est exprimé en fonction de l'apprenant</i>	<i>107</i>
<i>L'objectif opérationnel précise les conditions de manifestation du comportement... 109</i>	<i>109</i>
4.2.2 Les taxonomies :	111
4.2.3 Les dix avantages de la pédagogie par objectifs selon [Mager 75].....	112
4.3 PEDAGOGIE PAR OBJECTIFS ET APPRENTISSAGE A DISTANCE	113
4.4 DEFINITION DES BESOINS FONCTIONNELS	116
4.4.1 Expression des besoins du côté de l'auteur.....	116
4.4.2 Expression des besoins du côté de l'apprenant	118

4.5 MODELE DE STI PRODUIT.....	121
4.6 L'EDITION COOPERATIVE DU STI.....	124
4.7 ARCHITECTURE CLIENT SERVEUR DU WEB	125
4.7.1 Serveurs	127
4.7.2 Navigateurs.....	127
4.8 ARCHITECTURE GENERALE DE L'ENVIRONNEMENT TALHITS	129
4.8.1 Le mode apprenant Hits	130
4.8.2 Le mode auteur CamHits.....	130
4.8.3 Le navigateur.....	131
4.9 CONCLUSION	132
CHAPITRE 5	134
MODE APPRENANT – ARCHITECTURE LOGICIELLE.....	134
5.1 INTRODUCTION	134
5.2 ARCHITECTURE DU STI HYPERMEDIA (HITS).....	134
5.2.1 Coopération inter-composants	137
5.2.2 Les connaissances relatives au domaine d'apprentissage	138
5.2.3 Les paramètres quantitatifs du module d'enseignement	161
5.2.4 Le module d'enseignement	162
5.2.5 Les règles de recherche des uahs	164
5.3 LES CONNAISSANCES RELATIVES A L'APPRENANT	165
5.3.1 Le modèle de l'apprenant.....	166
5.3.2 Les règles d'acquisition (RIAC).....	168
5.3.3 Les règles d'évaluation (RIEV).....	169
5.4 LES CONNAISSANCES TUTORIELLES	173
5.4.1 Les connaissances de négociation.....	174
5.4.2 Les connaissances de gestion du séquençement des uah	176
5.4.3 Génération automatique du curriculum.....	180
5.5 LE MODULE DE COMMUNICATION INTERPERSONNELLE.....	181
5.5.1 Messagerie électronique.....	181
5.5.2 Forum de discussion.....	182
5.6 DEROULEMENT D'UNE SESSION D'APPRENTISSAGE.....	183
5.7 CONCLUSION	186

CHAPITRE 6	188
MODE AUTEUR COOPERATIF – ARCHITECTURE LOGICIELLE.....	188
6.1 INTRODUCTION	188
6.2 TRAVAIL COOPERATIF ASSISTE PAR ORDINATEUR.....	189
6.2.1 <i>Notion de Collecticiel</i>	190
6.2.2 <i>Coopération vs Collaboration</i>	191
6.2.3 <i>Coopération synchrone vs Coopération asynchrone</i>	192
6.3 L'EDITION COOPERATIVE DE DOCUMENTS	193
6.3.1 <i>Le processus d'édition coopérative</i>	194
6.3.2 <i>Fonctionnalités de base dans un collecticiel d'édition coopérative</i>	196
6.4 MODE AUTEUR COOPERATIF CAMHITS	201
6.4.1 <i>Modélisation de CamHits</i>	201
6.4.2 <i>Architecture logicielle de CamHits</i>	204
6.4.3 <i>Le module d'acquisition des paramètres</i>	211
6.4.4 <i>Le Générateur des règles instances</i>	212
6.4.5 <i>L'outil de mise au point</i>	215
6.5 CONCLUSION	216
CHAPITRE 7	218
PROTOTYPAGE ET UTILISATION	218
7.1 INTRODUCTION	218
7.2 ARCHITECTURE CLASSIQUE CLIENT/SERVEUR WEB	219
7.3 ARCHITECTURE BASEE SUR LES LANGAGES DE SCRIPT PHP/ASP/JSP	220
7.4 CHOIX DU SERVEUR DE BASES DE DONNEES.....	221
7.5 CHOIX DU SERVEUR WEB	222
7.6 LE MODE AUTEUR.....	224
7.6.1 <i>Outils utilisés</i>	224
7.6.2 <i>Fonctionnalités de base du mode auteur CamHits</i>	226
7.7 LE MODE APPRENANT	232
7.7.1 <i>Outils utilisés</i>	232
7.7.2 <i>Fonctionnalités de base du mode apprenant Hits</i>	234
7.8 CONCLUSION	241

CONCLUSIONS ET PERSPECTIVES	242
CONCLUSIONS	242
PERSPECTIVES D'AMELIORATION DU SYSTEME.....	244
BIBLIOGRAPHIE	246
ANNEXES.....	258



Liste des figures et des tables

Liste des figures

FIGURE 2.1.	EXEMPLE DE MANIPULATION D'UN TRIANGLE AVEC LE MICRO-MONDE CABRI-GEOMETRE	39
FIGURE 2.2.	ARCHITECTURE GENERALE D'UN STI	41
FIGURE 2.3.	UN EXEMPLE DE PLAN PEDAGOGIQUE.....	48
FIGURE 2.4.	ARCHITECTURE DE CALAT	66
FIGURE 2.5.	PROCESSUS DE GENERATION DE PAGES DE CALAT	66
FIGURE 2.6.	ARCHITECTURE LOGICIELLE DU SYSTEME ADIS	67
FIGURE 3.1.	UN SON EDITE AVEC WAVEEDIT	75
FIGURE 3.2.	LA TABLE DE MONTAGE DE DIRECTOR (UNE PISTE CONTIENT AU PLUS UN MEDIA).....	80
FIGURE 3.3.	CONSTRUCTION D'UNE APPLICATION AVEC AUTHORWARE.....	81
FIGURE 3.4.	PROCESSUS DE REALISATION D'UN LOGICIEL EDUCATIF A PARTIR D'UN SYSTEME AUTEUR.....	87
FIGURE 4.1.	HIERARCHIE DES 3 NIVEAUX D'OBJECTIFS PEDAGOGIQUES D'UN BUT PEDAGOGIQUE	106
FIGURE 4.2.	MODELE CONCEPTUEL DU SYSTEME AUTEUR	120
FIGURE 4.3.	UNE PARTIE DU MODELE CONCEPTUEL DES DONNEES DE LA PLATE-FORME IBN SINA	123
FIGURE 4.4.	DIALOGUE ENTRE UN CLIENT ET UN SERVEUR SELON LE PROTOCOLE HTTP	126
FIGURE 4.5.	ARCHITECTURE GENERALE DU SYSTEME AUTEUR TALHITS	129
FIGURE 5.1.	CREATION D'UN STI PAR TALHITS	135
FIGURE 5.2.	ARCHITECTURE LOGICIELLE DE HITS	136
FIGURE 5.3.	COOPERATION PAR ENVOI DE MESSAGE ENTRE LES COMPOSANTS	138
FIGURE 5.4.	SPECIFICATION 1484. 12. 1. DU GROUPE LTSC DE L'IEEE	141
FIGURE 5.5.	REPRESENTATION DU CONCEPT UPE DANS LE MODELE MODAP	142
FIGURE 5.6.	UN EXEMPLE D'UAH SANS EVALUATION : UNE INTRODUCTION AU LANGAGE PHP.....	146
FIGURE 5.7.	UN EXEMPLE D'UAH EVALUABLE : INITIATION A LA PROGRAMMATION EN LANGAGE PHP	147
FIGURE 5.8.	UN EXEMPLE D'UNE UAH D'EVALUATION.....	148
FIGURE 5.9.	UN EXEMPLE D'UNE UAH DE REMEDIATION.....	148
FIGURE 5.10.	MESURE DE L'ATTEINTE DES OBJECTIFS D'UNE UAH.....	149
FIGURE 5.11.	LES TROIS NIVEAUX DU RESEAU DE PREREQUIS DES OPS.....	151
FIGURE 5.12.	RESEAU DE PREREQUIS DES CONCEPTS (REPC).....	161
FIGURE 5.13.	MODELE CONCEPTUEL DU MODULE D'ENSEIGNEMENT	163
FIGURE 5.14.	EXEMPLE D'EVOLUTION D'UN MODELE DE L'APPRENANT	168
FIGURE 5.15.	UN EXEMPLE D'ARBRE D'EVALUATION	171
FIGURE 5.16.	UN EXEMPLE DE STRUCTURE D'UN COURS.....	175
FIGURE 6.1.	CLASSIFICATION ESPACE-TEMPS	193
FIGURE 6.2.	COOPERATION SYNCHRONE.....	198
FIGURE 6.3.	INTERFACE DE L'EDITEUR PARTAGE CALLIOPE [MITCHELL 1996]	199
FIGURE 6.4.	COOPERATION ASYNCHRONE	200
FIGURE 6.5.	LES TROIS UNIVERS CONCEPTUELS DE CAMHITS.....	202
FIGURE 6.6.	ARCHITECTURE LOGICIELLE EN COUCHES DU MODE AUTEUR	205
FIGURE 6.7.	STRUCTURE DES CONTROLEURS DE DIALOGUE CD ET CPD	208
FIGURE 6.8.	DIALOGUE DE SAISIE DES PARAMETRES DU STI	211
FIGURE 6.9.	DIALOGUE AFFICHE A L'ACTIVATION DU GENERATEUR DES REGLES INSTANCES.....	213

FIGURE 7.1.	ARCHITECTURE CLASSIQUE CLIENT-SERVEUR WEB	219
FIGURE 7.2.	ARCHITECTURE CLIENT SERVEUR UTILISANT DES LANGAGES DE SCRIPT	220
FIGURE 7.3.	GRAPHES D'IMPORTANCE D'UTILISATION DES SERVEURS WEB LES PLUS UTILISES	223
FIGURE 7.4.	ALGORITHME D'EXECUTION DE LA JSP	225
FIGURE 7.5.	BOITE DE DIALOGUE DU CHOIX DU MODE DE CONNEXION	226
FIGURE 7.6.	CREER UN NOUVEAU PROJET	227
FIGURE 7.7.	LA STRUCTURE LOGIQUE DU DOCUMENT	228
FIGURE 7.8.	ALLOCATION D'UN NŒUD	228
FIGURE 7.9.	L'EXPLORATEUR DE L'EDITEUR CAMHITS	229
FIGURE 7.10.	L'ETAT DE CONNEXION DES AUTEURS A UN MOMENT DONNE	229
FIGURE 7.11.	AFFECTATION DES ROLES AUX CO-AUTEURS	230
FIGURE 7.12.	LIBERATION D'UN NŒUD	230
FIGURE 7.13.	FENETRE D'EDITION DE TEXTE	231
FIGURE 7.14.	AJOUT D'UN MEDIA (IMAGE, SON, ETC.)	231
FIGURE 7.15.	PAGE D'ACCUEIL DU MODE APPRENANT	235
FIGURE 7.16.	APPRENTISSAGE LIBRE (MODE INFORMATION)	236
FIGURE 7.17.	DIALOGUE D'OUVERTURE D'UNE SESSION D'APPRENTISSAGE	236
FIGURE 7.18.	FORMULAIRE D'INSCRIPTION.....	237
FIGURE 7.19.	CHOIX DU POINT D'ENTREE DANS LE COURS.....	238
FIGURE 7.20.	PRETEST SUR LA PARTIE CHOISIE PAR L' APPRENANT	238
FIGURE 7.21.	PRESENTATION DE LA PREMIERE UAH DE LA PARTIE CHOISIE	239
FIGURE 7.22.	AFFICHAGE DES APPRENANTS EN LIGNE	240
FIGURE 7.23.	UTILISATION DE LA MESSAGERIE ELECTRONIQUE.....	240
FIGURE 7.24.	ACCES AU FORUM DE DISCUSSION	241

Liste des tables

TABLE 3.1.	QUELQUES EXEMPLES DE SYSTEMES AUTEURS AVEC LEUR NIVEAU DE PUISSANCE	85
TABLE 4.1.	TAXONOMIE DES OBJECTIFS PEDAGOGIQUES DE BLOOM	112
TABLE 4.2.	TAXONOMIE DES OBJECTIFS ET ACTIVITES PEDAGOGIQUES	112
TABLE 5.1.	METADONNEES D'UNE UNITE D' APPRENTISSAGE HYPERMEDIA	145
TABLE 5.2.	LES SIX NIVEAUX DE LA TAXONOMIE DE BLOOM ET LES VERBES D' ACTION ENVISAGES	152
TABLE 6.1.	FACTEURS CONCERNES PAR LA CONSCIENCE DU GROUPE	210



Chapitre 1

Introduction Générale

1.1 Préliminaires

Depuis l'apparition de l'informatique, le domaine de l'éducation - comme d'ailleurs beaucoup d'autres domaines - n'est pas resté insensible face à cette nouvelle technologie. Beaucoup de recherches se sont en effet effectuées, depuis les années soixante et jusqu'à ce jour, dans le but d'utiliser l'ordinateur pour accomplir une tâche aussi humaine qu'est l'enseignement (ou l'apprentissage).

Avant l'émergence des Nouvelles Technologies d'Information et de Communication (NTIC), deux grandes périodes de recherche ont marqué l'histoire de la médiatisation technologique de la pédagogie et ont contribué à l'élaboration d'un nombre assez appréciable de logiciels éducatifs plus ou moins réussis. Nous distinguons :

- L'ère de l'Enseignement Assisté par Ordinateur (EAO classique) [Bestougeff & Fargette 82] influencé par la psychologie béhavioriste [Cadet 98] et où furent les premières tentatives d'enseignement par des *didacticiels* construits d'une manière ad hoc.
- L'ère de l'Enseignement Intelligemment Assisté par Ordinateur (EIAO) [Sleeman & Brown 82] influencé par la psychologie cognitive [Cadet 98] et qui a exploité à bon escient les techniques de l'intelligence artificielle dans le but de réaliser des systèmes tuteurs et des environnements d'apprentissage plus intelligents que les didacticiels classiques et qui tiennent compte des capacités cognitives de l'apprenant. Notre contribution dans ce contexte fût à travers le système auteur de tuteurs intelligents Moalim, que nous avons réalisé dans le cadre de notre thèse de Magister [Talhi 96], [Talhi & al 96].

Cependant, la caractéristique essentielle des logiciels réalisés dans ces deux périodes – que ce soit pour les didacticiels classiques ou pour les Systèmes Tuteurs Intelligents (STI) [Wenger 87] [Nicaud & Vivet 88] – est le fait qu'ils soient tous conçus pour fonctionner en autonomie.

Si ces logiciels d'apprentissage faisaient généralement référence à un individu (l'apprenant) face à un ordinateur dans un lieu donné à un moment donné, il faut aujourd'hui, prendre en considération des dimensions nouvelles (temporelles, spatiales et sociales) qui viennent bouleverser cette situation. En effet, selon Baron [Baron 01], les possibilités technologiques des NTIC apparues ces dernières années, notamment le réseau mondial Internet et l'application « world wide web » qu'il supporte, et les nouveaux besoins de formation (formation à distance, formation tout au long de la vie, etc.) modifient considérablement les caractéristiques des environnements d'apprentissage et les questions qui s'en dégagent. Un nouveau moyen d'appréhender ces Environnements Informatiques pour l'Apprentissage Humain (EIAH) est alors de les considérer comme des environnements dans lesquels coopèrent des agents humains et des agents artificiels [Després 01].

Avec ces préoccupations, le domaine de l'enseignement médiatisé par ordinateur, à travers les deux nouveaux sigles que sont l'EAD (Enseignement A Distance) et l'ACAO (Apprentissage Coopératif Assisté par Ordinateur), a ouvert de nouvelles orientations de recherche qui tentent d'apporter des éléments de réponses à des questionnements récents.

Les travaux de recherche que nous menons dans cette thèse participent à cette mutation. Ils contribuent à proposer de nouvelles situations d'apprentissage en EAD en prenant en compte les aspects « Intelligence » et « Coopération » entre les différents agents liés aux EIAH. Nous tentons spécialement contribuer à résoudre le problème de la « sensation d'isolement » constatée souvent chez l'apprenant en situation d'apprentissage à distance, et ce, en lui apportant le soutien qui lui est indispensable pour conserver sa motivation et son progrès dans l'apprentissage.

Dans les sections qui suivent nous décrivons en premier lieu le contexte dans lequel est réalisé ce travail puis nous présentons les deux éléments de problématique qui ont motivé notre recherche à savoir l'« intelligence » et la « coopération » à intégrer dans les environnements informatiques pour l'apprentissage à distance. Après quoi, nous fixons les objectifs de notre recherche et nous terminons par une présentation du plan de cette thèse.

1.2 Contexte de travail

Le travail décrit dans cette thèse s'inscrit dans un cadre de recherche plus large : la plateforme d'enseignement collaboratif à distance Ibn Sina, dédiée aux universités algériennes qui font face à des effectifs de plus en plus grands et donc sont en quête de nouvelles méthodes et

stratégies pour assurer leur tâche pédagogique. Notre travail consiste au développement d'un système auteur coopératif de STI pour cette plate-forme.

Les enseignements s'effectuant en grande partie en présentiel actuellement, doivent être repensés sous une autre forme « mixte », c'est à dire en partie sur campus en présentiel et en partie à distance via l'Internet. La plate-forme Ibn Sina a pour ambition de satisfaire la deuxième forme, en permettant un enseignement/apprentissage plus ouvert et sans contraintes spatio-temporelles.

La plate-forme Ibn Sina permet de gérer un apprentissage coopératif à distance via l'Internet, intégrant plusieurs aspects :

- La communication et l'édition partagée de documents [Zidani & al 02] ;
- L'assistance à la navigation [Herrouz & al 02], [Ouchen & al 02] ;
- L'évaluation de la plateforme qui fait l'objet d'une thèse de Doctorat par notre collègue Zidat Samir [Zidat & Djoudi 06] et ;
- L'intégration des techniques des STI dans l'environnement d'apprentissage et la conception coopérative de ces STI qui font l'objet de cette thèse.

Le système auteur que nous proposons est baptisé TALHITS (Teaching And Learning by Hypermedia Intelligent Tutoring Systems : c'est à dire Enseigner et Apprendre par Systèmes Tuteurs Intelligents Hypermédias) consiste en deux environnements : un « mode auteur coopératif » CamHits [Talhi & al 06] dédié aux enseignants auteurs pour produire des STI Hypermédias, et « un mode apprenant » Hits [Talhi & al 05] permettant de guider l'apprenant dans son apprentissage..

Parmi les premiers besoins formulés pour la plate-forme Ibn Sina, il est recommandé que le système auteur soit, d'un côté, totalement indépendant de la matière à enseigner pour satisfaire le maximum de spécialités possibles, et de l'autre côté, doit-il être indépendant de la plate-forme matérielle et facile à utiliser par les deux acteurs auxquels il est destiné à savoir : l'enseignant auteur d'un côté et l'apprenant distant de l'autre côté. Du côté de l'apprenant en particulier, deux fonctionnalités sont exigées dans l'environnement à savoir : « l'interactivité » pour palier le phénomène de la « passivité » de l'apprenant distant et « l'adaptabilité » pour palier les deux difficultés généralement rencontrées dans les systèmes hypermédias que sont la « désorientation » et la « surcharge cognitive»

1.3 Motivations

Le travail de recherche que nous menons dans cette thèse est justifié par les deux motivations suivantes : guidage intelligent de l'apprenant à distance et réalisation coopérative de STI par les enseignants.

1.3.1 Guidage de l'apprenant à distance par tuteurs intelligents

Contrairement à la situation de formation en présentiel où l'apprenant est en contact direct avec le formateur, l'apprenant en situation de formation à distance est par définition éloigné géographiquement du formateur. Il est donc tout à fait légitime qu'il se sente plus confiant dans la première situation que dans la deuxième. Ceci est dû au fait qu'en présentiel, il est constamment guidé et assisté par le tuteur humain, alors que dans la situation à distance, il se sent livré à lui-même et la moindre difficulté rencontrée sur une tâche peut entraîner le blocage de son activité d'apprentissage dans son ensemble.

L'absence de soutien à l'apprenant dans les dispositifs de l'EAD entraîne généralement le découragement et l'abandon de nombreux étudiants. C'est le cas dans de nombreuses universités américaines et le CNED en France, où le taux de réussite annoncé, entre 1991 et 1996, est seulement de 30% [Perriault 96].

Afin de ne pas laisser un apprenant distant isolé et livré à lui-même, il convient alors de lui apporter le soutien et le guidage nécessaire pendant son activité d'apprentissage, c'est le seul garant pour maintenir sa motivation et sa persévérance. Si les dispositifs classiques de l'EAD reposant essentiellement sur le courrier postal ne répondaient guère à ce besoin de soutien, aujourd'hui grâce aux NTIC (internet, intranet, web, collecticiels, etc.) ceci est rendu possible que ce soit en mode synchrone ou en mode asynchrone.

Ces dernières années, l'EAD n'est plus l'apanage d'institutions spécialisées. De plus en plus d'universités dans le monde utilisent des plates-formes informatiques pour diffuser leurs cours à des étudiants distants. Certaines de ces plates-formes sont développées par des organismes privés et d'autres sont issues de travaux de recherche. Nous trouvons dans la littérature, deux études comparatives des principales plates-formes commercialisées (WebCT, VirtualU, LearningSpace, Librarian, TopClass, Pleiad, Classleader, etc.) réalisées, l'une par l'observatoire des ressources pour la formation ORAVEP [Oravep 00] et l'autre par l'association Préau [Préau 00]. Després [Després 01] remarque que ces plates-formes reposent presque toutes sur un modèle classique d'enseignement et d'encadrement pédagogique en EAD, puisque les activités d'apprentissage

qu'ils proposent se réduisent le plus souvent à consulter des cours classiques sous différents formats (html, pdf, ps, etc) et à répondre à des QCMs ; les questions pédagogiques étant souvent placées au second plan. Il fait remarquer également que ces plates-formes apportent peu de fonctionnalités de soutien à l'apprenant mais cherchent, en contrepartie, à favoriser la communication asynchrone apprenant-apprenant et apprenant-tuteur pour qu'il y est entraide mutuelle. Les fonctionnalités offertes par ces plates-formes, ont plus vocation à aider l'apprenant à organiser son parcours d'apprentissage qu'à le soutenir dans la réalisation de ses activités d'apprentissage.

Le vrai soutien, selon lui, c'est cette intervention très réactive d'un tuteur humain ou d'un système informatique intelligent qui guide et suit de très près l'apprenant à distance pour qu'il atteigne ses objectifs pendant son activité d'apprentissage. Ceci dans le but de ne pas laisser cet apprenant seul en situation d'échec ou de blocage et donc conserver sa motivation. Les travaux que Després [Després 01] a entrepris dans ce cadre concernent le soutien en mode synchrone par un tuteur humain à qui il a offert un ensemble d'outils pour qu'il puisse percevoir l'activité des apprenants et intervenir au moment opportun auprès d'eux pour les guider.

Conscients du problème de soutien aux apprenants dans les environnements informatiques d'enseignement à distance, et compte tenu de la pauvreté des systèmes existants concernant cet aspect, nous avons prévu deux modes de soutien dans la plate-forme Ibn Sina comme nous l'avons déjà évoqué plus haut : le soutien par « tuteur humain » et le soutien par tuteur informatique, le premier faisant partie d'un travail de thèse à part, le dernier étant l'objet de cette thèse. Trois considérations nous ont conduit à considérer ces deux types de tutorat :

- Le premier facteur est la diversité des disciplines enseignées dans les universités algériennes. Certaines disciplines nécessitent par exemple un *mode d'Apprentissage Par l'Action* (APA) et il nous a semblé impossible de suivre totalement ce genre d'activités par un tuteur informatique. Mieux vaut un soutien humain car le plus apte à réellement s'adapter à la démarche de l'apprenant. Certaines autres disciplines nécessitent (ou plutôt se contentent) d'un *mode d'Apprentissage Par l'Instruction* (API) ou un *mode d'Apprentissage par l'Exploration* (APE) et donc ne nécessitent pas vraiment la présence d'un tuteur humain. Un système informatique peut aisément s'occuper de la planification et la transmission des connaissances à l'apprenant et/ou bien l'aider à organiser son exploration, puis mesurer par la suite l'atteinte des objectifs visés.

- Le deuxième facteur est social. Les relations sociales entre enseignants et apprenants et la nature sociale de l'être humain en général ne peuvent être remplacées par aucun système informatique. La chaleur humaine est irremplaçable et elle agit comme un facteur essentiel favorisant la persévérance. Lise Desmarais [Desmarais 00] affirme cette réalité en mettant en évidence le rôle primordial joué par le tuteur humain pour la réussite des apprenants à distance et en particulier ceux se trouvant dans une situation critique proche de l'abandon.
- Le troisième facteur, lui, est la contrainte temporelle qu'exige le tutorat humain en mode synchrone. En effet, nous ne pouvons imaginer la disponibilité du tuteur humain tout au long de l'activité d'apprentissage des apprenants. Un apprenant qui se trouve par exemple en situation d'apprentissage en dehors des plages horaires de la disponibilité en ligne du tuteur humain peut s'il le désire être guidé et suivi par le tuteur informatique.

Cependant, il faut remarquer que la réalisation de ces tuteurs informatiques intelligents n'est pas chose aisée pour les enseignants formateurs qui, dans leur majorité, ne possèdent pas de compétences en informatique. Dans le but de leur faciliter la tâche et les décharger de toute activité de programmation, nous avons conçu pour eux un STI hypermédia générique et un environnement auteur coopératif qui n'exige d'eux que d'être experts dans leur matière à enseigner (contenu et pédagogie) pour instancier le canevas de STI disponible. Dans ce qui suit, nous allons décrire la problématique des « systèmes auteurs coopératifs » dédiés à la production de ces STI hypermédias.

1.3.2 Réalisation coopérative de STI

Malgré les efforts consentis ces trois dernières décennies, la réalisation de STI demeure une tâche difficile à entreprendre. Pour réussir, cette tâche exige souvent la collaboration entre équipes pluridisciplinaires constituées de pédagogues, psychologues, experts de domaines et informaticiens. La difficulté de constitution de ces équipes en pratique, a néanmoins eu un effet réducteur quant à la productivité dans ce domaine. En vue d'améliorer la productivité et permettre à une communauté plus large de s'impliquer, des systèmes auteurs sont apparus pour assister les développeurs et ont permis de réaliser plus ou moins aisément des STI. La tâche est alors réduite aux experts de domaines à faire couler des connaissances dans un canevas de STI générique prédéterminé par le système. Certains de ces systèmes auteurs sont discutés dans [Murray 99], cependant, ils ont tous été conçus pour fonctionner en mono-usager.

De nos jours, nous pensons que la disponibilité de ces systèmes ne résout pas totalement le problème. En effet, avec la complexité et l'interdépendance des sciences actuelles, la difficulté de construction d'un STI par un auteur autonome reste toujours à l'ordre du jour. Formuler et structurer les connaissances du domaine et les connaissances pédagogiques pour les faire introduire dans le système n'est pas une chose aisée. Ainsi donc et même du point de vue « saisie » et « structuration des connaissances » seulement, cette construction nécessite souvent la coopération de divers experts du domaine en question. Etant évidemment géographiquement éloignés dans la plupart du temps, il est par conséquent nécessaire de mettre à leur disposition des supports coopératifs leurs permettant de communiquer et coordonner leurs activités.

Aujourd'hui, grâce à l'intégration des NTIC, notamment le réseau mondial Internet avec son application web (www) et les collecticiels, des rencontres virtuelles à large échelle sont rendues possibles [Cockburn & Greenberg 93] [Reynard & al 98]. Plusieurs travaux dans ce sens ont déjà porté sur des domaines tels que l'édition coopérative de documents sur le web [Decouchant & al 95] [Pacull & al 94] et la conception coopérative d'objets ou de mondes virtuels, etc. Le point commun entre tous ces systèmes est qu'ils permettent à plusieurs participants de travailler ensemble de manière synchrone ou asynchrone pour réaliser une tâche commune.

D'autant plus que la technologie le permet donc et est expérimentée avec succès dans d'autres domaines, ceci nous conduit à penser qu'il serait souhaitable que les systèmes auteurs de STI du futur (spécialement leur mode auteur) intègrent la dimension coopérative à travers un réseau informatique local ou public. C'est dans cette optique que nous proposons notre système auteur coopératif de STI hypermédias et que nous avons baptisé TalHits.

Plusieurs travaux de recherche ont déjà porté sur la réalisation de systèmes auteurs de STI mono-usagers. Outre notre expérience avec le système Moalim cité plus haut, T. Murray [Murray 99] en a cité plus de deux douzaines dans sa synthèse la plus récente. Parmi ces systèmes nous pouvons citer : Cream-tools [Nkambou & al 96], Eon [Murray 98], Iris [Arruarte & al 97], Training-Express [Clancey & Joarger 88], etc. L'auteur en a fait une classification en sept catégories relativement au type de STI qu'ils produisent, à savoir : *enchaînement et planification de curriculum, stratégies tutorielles, simulation et entraînement, système expert en domaine, types de connaissances multiples, système à but spécial et, hypermédia intelligent et adaptatif.*

TALHITS, le système que nous proposons dans cette thèse et qui offre comme « originalité » *des fonctionnalités de coopération aux auteurs*, génère des STI hypermédias qui se situent dans la première et septième catégorie de cette classification : c'est à dire (*enchaînement et planification de*

curriculum) et (*hypermédia intelligent et adaptatif*). Les systèmes auteurs de cette catégorie structurent généralement la matière à enseigner sous forme d'un réseau d'Unités d'Apprentissage (UA) à granularités variables et où chaque UA possède certains objectifs pédagogiques.

Les UA sont liés entre elles par des liens de type « prérequis », « partie de », « défini par », « expliqué par », etc. Quoique ces systèmes n'utilisent pas de représentation explicite des connaissances du domaine, ils investissent cependant l'intelligence au niveau de l'enchaînement des UA, la manipulation des liens hypertextes et l'adaptation du cursus par l'utilisation d'un modèle de l'apprenant. Les UA à présenter à l'apprenant sont alors déterminées dynamiquement en se basant sur les performances de ce dernier, les objectifs pédagogiques de la leçon et les relations qui existent entre les différentes UA.

Le système coopératif TalHits est l'un des systèmes qui, dans le souci de faciliter la tâche aux auteurs, ne demande que d'instancier les UA. Les auteurs devront ensuite introduire les réseaux de prérequis et les paramètres nécessaires au fonctionnement du STI. Nous devons fournir, à travers ce collecticiel, un espace de travail commun à l'ensemble des auteurs impliqués dans la construction coopérative d'un STI. Cet espace doit permettre, à travers une interface graphique interactive, de leur apporter suffisamment de connaissances sur l'état courant du STI partagé ainsi que sur leurs contributions mutuelles.

1.4 Domaines de recherche

De ce qui précède, nous résumons que l'objectif de notre travail est de concevoir deux environnements : l'un, CamHits qui consiste en un certain nombre d'outils dédiés aux auteurs, et l'autre, Hits qui consiste en un canevas de STI hypermédia générique que les auteursinstancient à travers le mode auteur pour obtenir des STI dédiés à différents domaines d'apprentissage.

Le canevas de STI « Hits » est conçu sur la base de deux paradigmes conjugués à savoir celui des *hypermédi*as et celui des *systèmes tuteurs intelligents*. Le mode auteur CamHits, lui, repose également sur deux paradigmes que sont celui d' *hypermédia* et celui de *coopération*.

Dû à sa nature pluridisciplinaire, la conception du système nous a mené à fouiller dans trois domaines différents de recherche dans le but de comprendre les concepts et les méthodologies utilisées dans chaque domaine et en tirer profit de ce qui peut être utile et adaptable pour notre système. Ces trois domaines de recherche sont :

- Premièrement, celui de l' « éducation et la pédagogie en général ainsi que les technologies de leur médiatisation par ordinateur » (théories pédagogiques d'apprentissage, enseignement assisté par ordinateur (EAO), enseignement intelligemment assisté par ordinateur (EIAO), intelligence artificielle, environnements interactifs pour l'apprentissage humain (EIAH), enseignement à distance (EAD), apprentissage coopératif assisté par ordinateur (ACAO), etc.) ;
- Deuxièmement, celui des « systèmes d'information basés sur le web » avec toute la technologie qui gravite autour d'eux (Internet, Intranet, Web, Modèle Client-Serveur, Serveur Web, Navigateur, HTML, Xml, Java, Applet, CGI, etc.) et ;
- Troisièmement, celui du « Travail Coopératif Assisté par Ordinateur » (TCAO) et plus spécialement les applications d'édition coopérative de documents structurés.

1.5 Objectifs

L'objet de cette thèse est de concevoir un système auteur coopératif de STI hypermédias qui soit facile à utiliser par les utilisateurs potentiels que sont les enseignants et étudiants universitaires. Dans ce qui suit, nous discutons les principaux objectifs que nous visons dans cette thèse.

1.5.1 Indépendance de la matière d'enseignement

Un des objectifs que nous visons est que notre système soit totalement indépendant de la matière à enseigner pour pouvoir l'utiliser dans l'enseignement à distance de différentes spécialités. Ceci nous mène à étudier les techniques de structuration des connaissances, réutilisables pour différentes matières, pour concevoir la « *base de connaissances du domaine* » qui est l'élément clé dans tout STI. Un des objectifs de notre recherche est d'explorer comment les représentations standards de métadonnées d'objets pédagogiques, en cours de normalisation par certains organismes, peuvent-elles être utilisées intelligemment pour assister l'apprenant en situation d'apprentissage à distance.

1.5.2 Modèle Pédagogique réutilisable

L'approche pédagogique, indépendamment de la matière à enseigner, est aussi un point clé de notre recherche. En effet, les différentes matières ne s'enseignent pas nécessairement de la même manière dans la pratique. On n'enseigne pas, par exemple, les mathématiques comme le dessin industriel, la physique ou encore comme la musique. Chaque matière nécessite ses propres

approches pédagogiques (pédagogie par objectif, pédagogie de projet, etc.) et ses propres tactiques pour aborder l'enseignement (cours magistral, exercices, projet, simulation, expériences, écoute et répétition, visite sur terrain, etc.) [Casse 90]. En plus, il y a différentes théories d'apprentissage appliquées dans l'enseignement médiatisé par ordinateur du genre constructivisme, objectivisme, béhaviorisme, cognitivisme, etc. Les STI réalisés cités dans la littérature apportent en général des solutions fortement dépendantes de leur domaine d'application respectif et rares sont ceux qui ont réussi à définir une approche réutilisable pour différents domaines. Notre objectif est d'explorer ces différentes approches et théories pour tirer les concepts pédagogiques clés sur lesquels nous construisons notre modèle de STI.

1.5.3 Indépendance de la plate-forme matérielle

L'Internet est réputé comme étant le réseau des réseaux. Ses nœuds sont généralement formés de machines hétérogènes supportant différents types de matériels et de logiciels (systèmes d'exploitation, applications, navigateur web, etc.). Pour des raisons de portabilité, nous fixons donc comme objectif pour notre système d'être indépendant de la plate-forme sur laquelle il sera exécuté.

1.5.4 Proposition d'une architecture pour le mode apprenant

Pour construire un STI Hypermédia sur le web, il existe plusieurs modèles et architectures envisageables, en voici juste quatre exemples :

1. *Applet Java* : Le STI réside entièrement dans une applet Java. Cette applet pourra être téléchargée par un apprenant en visitant l'URL spécifique, puis exécutée dans le poste client (poste de l'apprenant) ;

2. *Architecture CGI* : Toutes les fonctionnalités du STI résident sur le serveur (dans un script CGI écrit par exemple en C, Perl, C++, etc.) et l'interaction de l'apprenant avec le programme CGI se fait via un navigateur web standard (comme MS Internet Explorer ou Netscape Navigator) qui envoie la requête de l'apprenant au serveur web pour lancer le programme CGI. Ce dernier exécute ce qui est demandé puis génère une page HTML qu'il retourne au serveur qui l'achemine à son tour au navigateur ;

3. *Architecture PHP/ASP/JSP* : Le module PHP (ou ASP ou JSP) est intégré dans le serveur web lui-même. Lors d'une requête provenant du navigateur, le serveur exécute le code PHP (ou ASP ou JSP) faisant partie d'une page HTML et les résultats sont intégrés au flot HTML qu'il renvoie au navigateur.

4. *Architecture à base d'agents* : Le STI est conçu selon une architecture à base d'un agent unique ou bien à base de plusieurs agents distribués entre les postes clients et serveurs et qui coopèrent entre eux pour accomplir la tâche d'assistance à l'apprenant.

Chacune de ces approches possède des avantages et des inconvénients. Notre objectif est d'étudier et de comparer ces différentes architectures en vue de bénéficier des points forts de chacune d'elles et s'inspirer dans la conception de notre tuteur générique Hits.

1.5.5 Proposition d'une architecture pour le mode auteur

S'interroger sur l'Assistance par Ordinateur au Travail Coopératif de groupe, c'est se poser d'abord la question de la manière dont les groupes travaillent pour réaliser une tâche commune, et se demander ensuite en quoi les NTIC peuvent apporter une contribution à l'efficacité de cette coopération.

Le TCAO est un domaine particulièrement récent qui ne date que du milieu des années 80. Des auteurs comme [Bannon & Schmidt 91] l'ont défini comme étant « une tentative pour comprendre la nature et les caractéristiques du travail coopératif avec comme objectif la conception d'une technologie informatique adéquate ». Cette définition révèle bien l'aspect pluridisciplinaire de ce domaine. Les recherches effectuées montrent en effet, qu'en plus des domaines informatiques variés (réseaux et télécommunications, intelligence artificielle, systèmes distribués, interfaces hommes-machines), les domaines des sciences humaines (psychologie, sociologie, linguistique) sont aussi d'une grande importance pour la conception de collecticiels (ou groupwares). Avec cette multitude de domaines impliqués, il n'est donc pas étonnant de trouver la littérature repue de modèles de coopération différents sur les plans : communicationnels, organisationnels et architecturaux.

Dans le domaine de l'éducation, un champs de recherche s'est détaché du TCAO ces dernières années (l'ACAO) dans le but de concentrer les efforts sur les modalités de coopération en enseignement et apprentissage. Dans ce cadre, beaucoup de prototypes d'environnements coopératifs d'apprentissages ont vu le jour. Cependant, la plupart de ces environnements centrent leur intérêt sur la tâche d'apprentissage elle-même et donc sur les apprenants; nous n'avons pas encore trouvé de travaux qui s'occupent de la production coopérative de connaissances d'apprentissage.

En ce qui nous concerne, la tâche commune à réaliser coopérativement est centrée sur l'enseignant et consiste en la « *production de contenus pédagogiques hypermédia* ». Notre objectif est donc

d'explorer et de comprendre les différentes méthodologies et concepts utilisés dans les domaines du TCAO et de l'ACAO en vue de proposer un modèle de coopération et une architecture logicielle rassemblant tous les outils et mécanismes nécessaires permettant la construction coopérative de STI.

1.6 Plan de la thèse

Cette thèse se compose de sept chapitres, qui, après cette présentation générale de notre problématique, s'articulent de la façon suivante :

Les deux premiers chapitres, après celui de l'introduction générale, situent le cadre théorique de notre travail. Nous examinons au chapitre 2 les différentes approches utilisées dans la construction des STI. Nous partons de ceux réalisés dans le cadre de plates-formes autonomes pour aller vers ceux qui sont réalisés pour l'usage sur le web. Ce chapitre nous permet également de mettre en place la terminologie utilisée et de discuter de quelques notions utiles pour la suite de l'exposé.

Dans le chapitre 3, nous présentons un bref bilan sur les systèmes auteurs de tuteurs intelligents en montrant les différentes architectures décrites dans la littérature. Nous partons des systèmes auteurs mono-usagers pour aller vers ceux qui sont ouverts sur le web.

Les quatre derniers chapitres décrivent notre système coopératif TalHits. Le chapitre 4 est consacré à la présentation générale du système. Nous y décrivons d'abord l'approche pédagogique sur laquelle il repose, puis nous définissons ses besoins fonctionnels. Nous décrivons par la suite l'architecture logicielle de TalHits.

Dans le chapitre 5, nous décrivons d'une manière plus détaillée le fonctionnement du mode apprenant Hits. Nous montrons les différentes représentations de connaissances choisies ainsi que les différents modes de raisonnement adoptés. Le chapitre 6 est consacré au mode auteur coopératif CamHits. Son architecture logicielle ainsi que le modèle de coopération choisi seront expliqués. Le chapitre 7 expose le prototypage et les modalités d'utilisation du prototype implémenté. Enfin, les conclusions générales et les perspectives d'amélioration de notre système composent la partie ultime de cette thèse.



Chapitre 2

Systemes hypermédias et tuteurs intelligents pour l'enseignement à distance sur le web

2.1 Introduction

A l'âge de la société de l'information, les nations développées ont pris conscience du fait que leur capital le plus précieux est le capital humain. Il ne suffit plus aujourd'hui d'avoir accès à des richesses naturelles abondantes pour créer et entretenir la prospérité. Une partie de plus en plus significative de la valeur ajoutée est le fruit direct de l'intelligence et de l'ingéniosité humaines.

Pour de nombreuses activités dans des domaines très variés, l'avantage compétitif réside non plus exclusivement dans la possession de chaînes de production performantes ou dans l'accès privilégié aux ressources naturelles, mais bien dans la capacité à mobiliser rapidement les *compétences* nécessaires pour faire face avec la plus grande efficacité aux problèmes de toute nature qui surgissent au fil du temps.

La création et le développement de *compétences* est l'objectif premier de tout *système de formation*, qu'il soit de formation initiale ou de formation continue. On doit donc considérer que la qualité des systèmes de formation est un facteur critique pour le développement des sociétés modernes.

Alors qu'aucune entreprise ne peut espérer survivre sans l'entretien périodique de ses machines de production, force est de constater que l'entretien des compétences du personnel est pour autant considéré comme un élément clé pour son développement et sa réussite. Les grandes sociétés ont, en effet, découvert depuis longtemps la nécessité de former leur personnel de façon permanente et ont donc toujours favorisé la formation continue de leur personnel.

Si jadis, le succès des offres de formation dans ce contexte est relativement modeste, aujourd'hui, le développement des technologies de l'information et de la communication (TIC) a donné lieu à l'espoir qu'il serait possible de pallier certaines difficultés rencontrées auparavant (du genre : manque de disponibilité de tuteurs ou contrainte de lieu et/ou de temps, etc.), et ce, en

exploitant les potentiels de l'informatique et des télécommunications pour mettre en place des dispositifs de formation mieux adaptés aux besoins.

Selon un article de la revue *Fortune* (Mai 1999), 92 % des grandes entreprises américaines implémentent actuellement des systèmes de formation en ligne pour leur personnel. Une des raisons invoquées pour ce choix est le potentiel de diminution significative du coût de la formation du personnel dans l'entreprise: une étude réalisée en 1994 fait apparaître que le coût de la formation peut être divisé de moitié par rapport à une formation en présentiel. Non seulement l'entreprise fait l'économie des frais de déplacement, mais il apparaît des études de cas que la durée de la formation peut être réduite (jusqu'à 40 %) par une formule d'auto-apprentissage soutenue par les technologies.

Par ailleurs, un article de *Business Week* (9 août 1999) fait état de l'existence de formations, voire même de programmes entiers, offerts en ligne par des universités réputées telles que Stanford, Columbia, NYU, Wharton, etc. En outre, plusieurs universités privées (comme Phoenix Online, Jones International, UNext, University Access, eCollege, ZDU, etc.) ont été créées spécifiquement dans le but de répondre aux attentes du marché de la formation en ligne.

Pour ce qui est de l'Europe, un sondage effectué en 1998 par l'Institut de l'Audiovisuel et des Télécommunications en Europe, auprès d'une centaine d'institutions éducatives a permis de recenser 74% d'intérêt pour les télé-services en général et la *formation en ligne* en particulier. Pionnière en matière d'enseignement à distance, l'OPEN UNIVERSITY de Londres a exploité l'Internet et s'est transformé en une multinationale de la formation supérieure, traduisant ses cours en différentes langues et ouvrant des antennes, notamment, dans la plupart des pays européens où elle entre directement en compétition avec les institutions similaires. En France également, le Centre National d'Enseignement à Distance (CNED) est au rendez-vous et il s'est fixé pour objectif de convertir tous les supports de cours imprimés en supports numériques, pour être transportés plus facilement à l'autre bout du monde [Charrier 01].

En Algérie, il faut bien constater que l'on ne dispose pas de chiffres et d'évaluations (tout au moins rendues publiques) sur l'utilisation des TIC en formation à distance. Cependant, l'Université de la Formation Continue (UFC), créé plus ou moins récemment en 1990, se propose ces dernières années de faire de l'enseignement à distance son mode privilégié d'intervention conformément à son statut, ainsi que d'intégrer les TIC dans son fonctionnement. Elle vient même de signer un accord de participation avec l'université virtuelle euro-méditerranéenne «Avicenne».

S'appuyant sur un réseau de 46 centres qui s'étend sur l'ensemble du pays, l'UFC a vocation à se transformer en université virtuelle nationale. Elle projette de constituer un pôle de compétences assurant, grâce au développement de la recherche, la promotion des technologies dans l'ensemble de l'enseignement supérieur Algérien [Khaldi 01].

Cependant, nous pensons que l'UFC seule ne peut espérer satisfaire toutes les demandes de la formation continue au niveau national. Les autres universités doivent réagir elles aussi et proposer des formations à distances à leur étudiants. Cette thèse tente de fournir une étude d'un système auteur leur permettant d'envisager ce type de formation à l'image de beaucoup d'universités du monde. Avant d'aborder la thématique des systèmes tuteurs intelligents sur le web, nous allons dans ce qui suit nous intéresser d'abord à la problématique du guidage et du soutien à apporter à l'apprenant en enseignement à distance et montrer par l'occasion la nécessité de disposer de tuteurs intelligents pour assurer ces deux besoins.

2.2 Problématique du guidage en enseignement à distance

Afin de maintenir la motivation de l'apprenant à distance et éviter qu'il soit seul en situation de blocage ou d'échec, l'apprenant doit pouvoir être guidé et soutenu à tout instant pendant la réalisation de son activité d'apprentissage. Il convient alors de nous interroger sur les modes de guidage et de soutien à fournir à l'apprenant dans un contexte de formation à distance. Commençons alors par explorer les pratiques en apprentissage à distance, d'abord à travers les établissements d'EAD, puis à travers les plates-formes de télé-formation les plus répandues pour identifier les différents types de soutien apportés aux apprenants, qu'elles proposent.

2.2.1 Etablissements d'enseignement à distance

Les établissements telles que le Centre National d'Enseignement à Distance (CNED) en France, l'Open University en Grande-Bretagne ou la Télé-Université au Québec ont été créés pour répondre à des besoins de formation visant des publics peu mobiles ou disposant d'emplois du temps fortement contraints.

L'Open University est une université indépendante et autonome fondée en 1969 en Angleterre. Elle compte aujourd'hui plus de 200.000 étudiants. Bâtie sur le modèle de l'université ouverte, comme son nom l'indique, elle assure à ses étudiants une formation continue mais aussi initiale, et délivre ainsi des diplômes de l'enseignement supérieur. D'autres universités se sont inspirées de ce modèle telles que l'Open Universitât en Hollande, la Fern Universitât en

Allemagne, l'Universidad Nacional de Education a Distancia en Espagne ou encore l'Universita Abierta au Portugal.

Le CNED en France a été fondé en 1939 pour la formation initiale des enfants avant de s'ouvrir à la formation des adultes en 1945. Il compte aujourd'hui plus de 400.000 étudiants et propose de la formation continue mais aussi initiale pour les cycles élémentaires, secondaires et l'enseignement supérieur.

Ces institutions reposent généralement sur un modèle d'enseignement et d'encadrement pédagogique très proche de celui de « l'enseignement par correspondance » apparu dans les années vingt. Dans une logique de transmission de la connaissance, les cours et les devoirs sont envoyés par la poste aux étudiants qui doivent les étudier seuls. Ces établissements mettent l'accent sur la conception, la mise en forme, la médiatisation et la diffusion des cours. Ils cherchent ainsi à modifier l'enseignement classique lors de la mise à distance de façon à ce que l'interaction directe ne s'établisse plus entre l'enseignant et l'étudiant mais entre le champ des connaissances et l'étudiant [Rigault-Riccardi 82]. L'encadrement se limite alors le plus souvent aux retours sur corrections et à d'éventuels échanges avec des tuteurs humains joignables par téléphone ou par e-mail. L'interaction entre tuteurs et apprenants est donc généralement lente et limitée.

Le modèle d'enseignement adopté par ces établissements répond plus à une logique de transmission des connaissances que de construction des connaissances. La mise en place d'activités d'apprentissage guidées et soutenues instantanément est totalement ignorée. De plus, les modes d'encadrement sont très limités et peu réactifs et ne répondent pas au besoin des apprenants engagés dans des activités d'apprentissage.

La formation à distance, avec ce modèle, requiert donc motivation et implication de la part des étudiants pour parer à la solitude. Les abandons sont par conséquent nombreux. Ces abandons sont souvent beaucoup plus liés à un découragement difficile à surmonter dans la solitude du parcours de formation qu'à des difficultés intellectuelles ou qu'à un niveau d'instruction insuffisant.

2.2.2 Plates-formes d'enseignement à distance

Avec l'avènement de l'Internet et du web, l'enseignement à distance n'est plus le privilège des grandes institutions. Même les universités de taille modeste peuvent désormais assurer une

formation à distance, grâce à des plates-formes de télé-enseignement qui fleurissent régulièrement à travers le monde.

Ces plates-formes permettent de gérer les inscriptions et les problèmes administratifs, de diffuser les cours et de mettre en place un encadrement pédagogique. Elles s'adressent aussi bien à des entreprises qu'à des universités. A l'heure actuelle, de plus en plus d'universités utilisent ces plates-formes pour diffuser leurs cours à des étudiants distants.

Dans la littérature, nous trouvons deux études comparatives réalisées sur les principales plates-formes commercialisées (comme WebCT, Virtual-U, Learning Space, Librarian, etc.): celle de l'ORAVEP [Oravep 00] et celle du Préau [Préau 00].

Nous constatons à travers ces études, comme [Desprès 01], que les préoccupations des éditeurs de plates-formes sont souvent techniques et consistent à rechercher des solutions permettant d'intégrer le plus grand nombre de standards de documents présents sur le marché et d'offrir des outils de création et de gestion de ces documents. Les questions pédagogiques sont souvent placées au second plan.

Si ces plates-formes permettent de démocratiser sans conteste la formation à distance en permettant des formations en masse à des coûts relativement faibles, elles reposent cependant sur un modèle classique d'enseignement et d'encadrement pédagogique. Les activités d'apprentissage se réduisent le plus souvent à consulter un cours et à répondre à des questions simples. La réalisation d'activités plus engageantes, fondées sur l'apprentissage par l'action (learning by doing), comme des travaux pratiques ou des exercices nécessitant la manipulation d'objets, n'est pas envisagée. Ces plates-formes se situent plutôt dans une logique de transmission de la connaissance.

Nous remarquons également que ces plates-formes n'apportent que peu de fonctionnalités de guidage et de soutien à l'étudiant. Certaines de ces fonctionnalités permettent toutefois à l'étudiant de gérer son parcours d'apprentissage. Suivant les plates-formes, l'étudiant peut consulter une vue sur son propre parcours sous forme d'indicateurs sur son état d'avancement (Class-Leader, Learning-Space, Pleiad, TopClass et WebCT), consulter des indicateurs d'auto-évaluation (Class-Leader, Learning-Space, TopClass, VirtualU et WebCT) ou encore comparer ses indicateurs personnels avec ceux des autres membres de son groupe (Class-Leader, VirtualU et WebCT).

Ces plates-formes cherchent aussi souvent à favoriser la communication entre étudiants pour qu'il y est entraide mutuelle. Si ces fonctionnalités permettent effectivement à l'étudiant de se situer par rapport à un parcours prévu ou par rapport aux autres étudiants, elles ne lui offrent cependant pas de soutien lors de la réalisation de l'activité d'apprentissage.

Le simple ajout de fonctionnalités de communication entre pairs dans le but de faire coopérer les étudiants, n'est pas non plus suffisant. Des travaux montrent en effet qu'il faut concevoir des activités d'apprentissage elles-mêmes coopératives et créant des inter-dépendances pour qu'il puisse y avoir coopération [George 01]. Aussi, faut-il prévoir initialement des fonctionnalités et des conditions favorables à l'émergence de groupes d'étudiants organisés et coopérants réellement [Hotte & Contamines 99].

Dans ces plates-formes, les tuteurs humains disposent également d'outils de communication, le plus souvent asynchrones, mais ils ne disposent pas d'outils spécifiquement conçus pour le tutorat à distance. Des indicateurs du type de ceux fournis aux étudiants leur permettent de prendre connaissance des parties du cours abordées par les étudiants mais ils n'ont aucune information sur la réalisation de leurs activités.

Ces fonctionnalités ont donc plus vocation à aider les apprenants à organiser leur parcours d'apprentissage qu'à les guider dans la réalisation de leurs activités d'apprentissage.

En outre, même si ces fonctionnalités peuvent résoudre les problèmes de certains apprenants, ce mode d'aide ne semble pas adapté à tous les individus car il présuppose souvent, et parfois à tort, de leurs capacités d'autonomie. Or, comme le fait remarquer Monique Linard [Linard 00], l'autonomie est faible ou inexistante chez une grande majorité d'étudiants et il conviendrait mieux de les aider à développer leur capacité de travail en autonomie que de présupposer de son existence pour mettre en place des formations à distance :

«L'autonomie n'est pas une simple qualité mais un mode de conduite intégrée (une méta-conduite) et, pour la plupart des individus, cette conduite ne faisant pas naturellement partie de leur répertoire, elle doit être apprise» [Linard 00].

2.2.3 Modalités de guidage de l'apprenant à distance

Partant des constats négatifs que nous venons d'établir, nous cherchons à apporter une solution au problème de guidage et de soutien à fournir à l'apprenant à distance, pour mener à bien son activité d'apprentissage. Trois solutions nous semblent éligibles : l'une consiste à faire intervenir le tutorat humain lui-même, l'autre consiste à intégrer un guidage totalement

informatisé par des systèmes tuteurs intelligents. Entre ces deux solutions extrêmes peut exister évidemment une troisième solution permettant de partager les rôles entre le tuteur humain et le tuteur informatique.

2.2.3.1 Guidage par tuteur humain

Dans cette situation, le tuteur humain est l'interlocuteur des apprenants, c'est lui qui répond à leurs sollicitations et les guide dans la réalisation de leurs activités d'apprentissage. Il semble que le soutien humain reste aujourd'hui encore le moyen le plus complet et aussi celui qui est le plus apte à réellement s'adapter à la démarche de l'apprenant, à prendre en compte une situation particulière (problème en cours de résolution, difficulté rencontrée, niveau de compétence de l'apprenant, caractéristiques cognitives, sociales et affectives, etc.). Monique Linard adhère à ce point de vue, considérant que tous les avantages que peuvent apporter les technologies pour la mise à distance d'activités d'apprentissage interactives ne sont pas suffisants sans le soutien d'un enseignant.

«Sauf pour les individus experts dans un domaine, qui ont par définition dépassé ce stade et sont devenus autonomes, l'acte d'apprendre est toujours à réapprendre et il faudra toujours un fort accompagnement humain à l'instrumentation technique» [Linard 00].

D'un point de vue théorique, c'est aussi l'avis de Vygotski [Vygotski 85] qui, dans l'approche socio-constructiviste de l'apprentissage, a mis en évidence l'intérêt des interactions sociales sur le développement cognitif. En s'intéressant au développement des fonctions cognitives chez l'apprenant, Vygotski a démontré que l'évaluation du niveau de développement cognitif ne donne pas le même résultat suivant que cette évaluation est réalisée d'après des problèmes résolus de façon autonome ou avec l'aide d'un adulte. Lorsqu'un apprenant résout seul un problème, il n'active que ses fonctions cognitives déjà formées et arrivées à maturité. Or, dans le même temps, d'autres fonctions sont en train de se construire, encore au stade de maturation et a priori non exploitables par l'apprenant lors de la résolution d'un problème de façon autonome. Ces fonctions peuvent néanmoins être utilisées par l'apprenant s'il est aidé par un expert qui tentera de le mettre sur la voie ou de lui apporter des débuts de solutions. L'apprenant sera alors «tiré en avant » pour résoudre des problèmes plus complexes.

Par ailleurs, pour en revenir au contexte de l'EAD, la question des relations sociales entre apprenants ou entre enseignants et apprenants y est revisitée car la distance physique crée une distance sociale qui accentue le phénomène d'isolement. Dans une étude sur la persévérance et les

taux d'abandon en EAD, Lise Desmarais a mis en évidence le rôle primordial joué par le tuteur humain pour la réussite des étudiants et en particulier ceux se trouvant dans une situation critique proche de l'abandon [Desmarais 00]. Elle explique qu'une relation étroite peut s'établir entre l'enseignant et l'apprenant distant, et agir comme un facteur essentiel favorisant la persévérance.

Toutefois, le tuteur à distance ne peut directement observer les apprenants comme il le fait en présence. Il n'a donc pas connaissance de la façon dont les apprenants réalisent leur activité ni des difficultés qu'ils rencontrent. Des travaux, dont celui de Christophe Després [Després 00], sont donc entrepris dans le but de concevoir des outils informatiques permettant au tuteur de percevoir l'activité des apprenants à distance et intervenir au moment opportun auprès d'eux pour les guider.

Cependant, ce suivi à distance par le tuteur humain en mode synchrone crée une situation particulière. Le tuteur doit être disponible durant toute la durée d'une activité inscrite dans le temps avec un début et une fin. Deux difficultés sont donc à surmonter par le tuteur: celui de la sur-sollicitation et celui de la contrainte temporelle.

a) phénomène de «sur-sollicitation» :

L'enseignant peut se trouver submergé par la charge d'activité que constitue le suivi synchrone des apprenants, il devient alors difficile pour lui de répartir son temps entre les différents apprenants et d'avoir la certitude de pouvoir intervenir au moment le plus opportun.

b) contrainte temporelle

Pour que le tuteur humain soit disponible tout au long de l'activité, il est nécessaire de définir des plages horaires pendant lesquelles les apprenants pourront réaliser leurs activités avec l'assurance de pouvoir être soutenus par le tuteur à tout moment. Cette nouvelle situation pose donc une contrainte forte en termes d'organisation et de coordination des différents acteurs (tuteur et apprenants). Ceci va donc à l'encontre d'un des atouts majeurs de l'EAD souvent mis en avant : la souplesse d'organisation.

Nous considérons donc que ce mode de guidage humain ne peut pas être envisagé comme une solution suffisante à elle seule. Ceci nous emmène à penser au guidage par systèmes tuteurs intelligents pour pallier ces deux problèmes. Ces tuteurs artificiels qui ont donné large satisfaction en autonomie peuvent également être déployés sur le web pour se substituer au tuteur humain, surtout pour permettre un mode d'apprentissage asynchrone sans contrainte temporelle.

2.2.3.2 Guidage par systèmes tuteurs intelligents

La deuxième solution de guidage et de soutien aux apprenants consiste donc à utiliser des STI. L'objectif initial des recherches menées sur les STI était de créer des logiciels capables de se substituer à l'enseignant humain. Ces STI assurent la conduite pédagogique de la session d'apprentissage, en s'adaptant aux objectifs et aux capacités intellectuelles de l'apprenant.

Les recherches effectuées dans le domaine depuis les années 70 ont pu délimiter les composants que doit inclure tout système prétendant être un STI [Wenger 87], [Nicaud & Vivet 88] : l'expertise du domaine (quoi enseigner ?), l'expertise pédagogique (comment enseigner ?), les connaissances sur l'apprenant (à qui enseigner ?) et l'interface de communication avec l'apprenant (sous quelle apparence communiquer ?). Si un consensus s'est forgé autour de ces principes, les systèmes diffèrent cependant, d'une part, par l'importance qu'ils accordent à chacune de ces tâches et, d'autre part, par les techniques d'intelligence artificielle (IA) mises en œuvre pour leur réalisation.

La principale contribution des STI relativement aux didacticiels classiques qui les ont précédés est la possibilité de modéliser l'expertise. Grâce à cette expertise, le système devient capable de résoudre lui-même les problèmes qu'il pose à l'apprenant. Le système est donc compétent dans le domaine à enseigner. Cette expertise modélisée permet au système de conduire des interactions qui ne pourraient pas l'être si le système travaillait avec des solutions préenregistrées. Aussi un STI est conçu de manière à ce qu'il génère lui-même une leçon adaptée à chaque apprenant, et ce, en se basant sur une représentation interne des connaissances de cet apprenant.

Si la majorité des plates-formes d'EAD existantes se situe dans une logique de transmission de la connaissance sous forme de documents hypertextes statiques via internet comme nous l'avons soulevé dans la section 1.2.2, beaucoup de recherches sont menées actuellement en vue de marier les techniques des STI avec celles des hypermédias pour créer des STI Hypermédias (ou Systèmes Hypermédias Adaptatifs [Brusilovsky 98]) qui sont plus intelligents et qui offrent plus d'interactivité et de guidage/adaptabilité pour l'apprenant.

Quelques systèmes de laboratoire sont issus de ces travaux, nous citons en particulier ELM-ART [Brusilovsky & al 96], CALAT [Nakabayashi 97], MANIC [Stern 97], etc.

Les technologies de guidage/adaptation concernent surtout des problèmes du genre : planification de tâches, analyse intelligente des solutions de l'apprenant, support interactif de résolution de problèmes, présentation adaptative, support de navigation adaptatif, etc.

[Brusilovsky 98]. La reconnaissance d'approches pédagogiques nouvelles et l'intérêt pour une pédagogie active sont également à inscrire au tableau des perspectives nouvellement ouvertes.

L'une des principales difficultés rencontrées est de parvenir à construire, en machine, un modèle des connaissances de l'apprenant que le STI pourra exploiter pour s'adapter à l'apprenant et lui apporter tout le soutien nécessaire. Plusieurs principes de modélisation ont été proposés mais aucun ne s'est réellement imposé. Nous décrivons quelques uns de ces principes au paragraphe 2.3.

De plus, les recherches menées tendent à montrer que l'activité de l'apprenant n'est pas uniquement liée à ses représentations mentales, mais qu'elle est située socialement et physiquement. L'apprenant n'élabore pas des plans d'action immuables et préalables à l'activité, il les produit, les évalue et les modifie en cours d'action [Nardi 96].

Beaucoup de recherches donc doivent encore être menées pour améliorer les STI sur le web au niveau du contrôle, de la flexibilité, de la modélisation efficace de l'apprenant et de la représentation du domaine. Tant que ces difficultés n'auront pas été totalement surmontées, les STI ne pourront se substituer entièrement à l'enseignant humain. Nous considérons donc, comme beaucoup d'autres chercheurs, qu'à l'heure actuelle, les STI ne peuvent pas être envisagés comme une solution suffisante pour soutenir pleinement des apprenants à distance. Une coopération entre le tuteur humain et le tuteur artificiel pourrait pallier ce problème comme nous le verrons dans la section suivante.

2.2.3.3 Guidage par coopération entre tuteur humain et tuteur artificiel

Afin de limiter les phénomènes de sur-sollicitation et la contrainte temporelle inhérents au mode tutorat humain d'un côté, et pallier aux difficultés techniques de concevoir des STI pouvant se substituer pleinement au tuteurs humains d'un autre côté, l'environnement informatique doit avoir la capacité de faire coopérer les deux tuteurs humain et artificiel pour prendre en charge le soutien et guidage des apprenants à distance. Par conséquent, dans une logique de système partenaire de l'enseignant [Balacheff & al. 98], il faut intégrer à l'environnement informatique, en plus des outils de suivi de l'activité et d'intervention auprès des apprenants, des assistants intelligents prenant en charge certaines difficultés rencontrées par les apprenants.

Les systèmes conseillers sont particulièrement adaptés pour réaliser ce type d'assistance localement. Dans les systèmes conseillers les conditions de déclenchement de l'assistance et le

choix des conseils à apporter sont déterminés par l'analyse des actions réalisées par l'apprenant à l'interface du système [Paquette & Tchounikine 02].

Cela suppose que le système dispose d'un modèle de la tâche qui lui permette de suivre l'activité de l'apprenant et de la comparer avec le modèle des solutions possibles pour réaliser la tâche. Si l'apprenant s'écarte d'un des cheminements corrects, le système est en mesure de l'assister grâce à des conseils associés au modèle de la tâche. La question du contrôle de l'interaction tutorielle à laquelle les STI cherchent à répondre est ignorée dans les systèmes conseillers mais leur intérêt est qu'ils sont considérablement plus simples à mettre en oeuvre que les STI car ils nécessitent un travail moindre au niveau du recueil de l'expertise du domaine et de la modélisation de l'apprenant.

En effet, pour réaliser un système conseiller, on cherche à modéliser une tâche particulière, celle pour laquelle on souhaite assister l'apprenant, mais ce travail est moins coûteux que la modélisation d'un domaine de connaissances tout entier. De plus, l'assistance apportée est directement fonction des actions de l'apprenant et non des connaissances que le système lui attribue.

Ainsi donc le système informatique partenaire du tuteur humain peut coopérer avec ce dernier de façon directe en lui fournissant un ensemble d'informations sur le déroulement des activités d'apprentissage à distance ainsi que les moyens d'intervenir auprès des apprenants, et indirectement en assistant les apprenants dans la réalisation d'une tâche particulière afin de le soulager. Dans un tel environnement informatique, c'est le tuteur humain qui garde le contrôle sur la conduite de la session pédagogique et applique sa propre pédagogie tout en s'appuyant sur un ensemble d'assistants intelligents capables d'assister localement et ponctuellement les apprenants. Le tuteur humain se trouve ainsi déchargé de certaines tâches que le système informatique est capable de réaliser ce qui lui permet d'être plus disponible vis à vis des apprenants pour les soutenir.

2.3 Systèmes classiques d'enseignement médiatisé par ordinateur

Les premiers essais d'utilisation de l'ordinateur pour assister l'enseignement dataient des années soixante, ce qui a fait émerger un nouveau domaine de recherche l'EAO (Enseignement Assisté par Ordinateur) [Charlot 98]. Les didacticiels réalisés à cette époque ont tous été influencés par les théories béhavioristes (ou comportementales) de Skinner [Bordeleau 99]. Cette théorie de l'apprentissage a donné naissance à une ingénierie pédagogique, nommée

enseignement programmé, dont l'objectif est d'automatiser le processus d'apprentissage sous la forme stimulus-réponse-renforcement.

On nommait ces systèmes les systèmes "si..alors..sinon" [Lelouche 87], car leur algorithme était basé sur ce principe : si l'apprenant répond bien, le système lui présente l'élément suivant du cours, sinon il lui réexpose l'élément de cours qui n'est pas assimilé.

Cependant, les différentes expériences menées en EAO traditionnel ont fait apparaître beaucoup de limitations dans les logiciels produits : absence de mécanismes de raisonnement, pas de séparation entre les connaissances à enseigner et les mécanismes du contrôle du programme, absence d'analyse du langage naturel, incapacité à comprendre les erreurs de l'apprenant et donc incapacité à le modéliser efficacement, etc.

Les résultats obtenus dans les recherches menées en psychologie cognitive et en Intelligence Artificielle (IA), pendant les années soixante dix, ont été d'un grand secours pour les chercheurs en EAO et leur ont permis de produire des logiciels beaucoup plus performants et plus intelligents. Le sigle EAO s'est fait alors évoluer du « I » de *Intelligence/Interactivité* ouvrant un nouveau domaine de recherche pluridisciplinaire impliquant plusieurs disciplines (IA, didactique, psychologie cognitive et sciences de l'éducation) : l'EIAO. Ce sigle a été employé par certains pour désigner « Enseignement Intelligemment Assisté par Ordinateur » et par d'autres pour désigner « Environnements Interactifs d'Apprentissage par Ordinateur ». Le premier s'appuie sur un comportement intelligent des systèmes, alors que le second s'intéresse plus particulièrement à l'apprentissage via l'interaction et donc par conséquent, deux types de systèmes d'enseignement ont vu le jour durant les deux décennies 70-80. Tout d'abord les micro-mondes puis les systèmes tuteurs intelligents. Dans ce qui suit, nous allons présenter ces deux types de systèmes tout en s'attardant un peu plus sur les systèmes tuteurs intelligents, ce qui nous permettra de mieux identifier les objectifs que nous nous sommes fixés.

2.3.1 Micro-mondes

Les micro-mondes s'attachent à une vision apprentissage plutôt qu'enseignement, c'est à dire qu'une très large initiative est laissée à l'apprenant. Les micro-mondes reposent sur une théorie d'apprentissage dite constructiviste qui s'est développée après la seconde guerre mondiale par Jean Piaget. Pour Piaget, l'intelligence permet à l'être humain de s'adapter continuellement à son environnement. Cette adaptation, que l'on appelle aussi apprentissage, est induite par les actions de l'homme sur le monde extérieur. Ainsi il ne peut y avoir apprentissage sans action ou plus

exactement sans interaction. Issus de cette théorie, beaucoup de micro-mondes ont été développés depuis la fin des années soixante. Certains d'entre eux sont distribués comme des produits commerciaux et ont eu un succès réel, comme Logo [Papert 80] et Cabri-Géomètre [Laborde 95]. On peut caractériser un micro-monde comme étant un logiciel éducatif dénué de toute connaissance, mais qui de par l'interaction qu'il offre à l'apprenant à travers des objets et d'opérateurs de manipulation, permet à ce dernier d'assimiler et de comprendre plus facilement des connaissances.

Le langage LOGO [Papert 80], créé au début des années 1970, est le précurseur des micro-mondes, il est consacré à l'apprentissage de la programmation récursive et la géométrie. La première version de ce langage date de 1966, mais ce n'est qu'en 1970 qu'apparaît la première version graphique, où la tortue (élément central du langage) est représentée à l'aide d'un petit triangle. En déplaçant cette tortue, l'élève peut alors dessiner des figures géométriques. Par la suite, une véritable tortue robot sera utilisée.

Le système Cabri-Géomètre (pour Cahier de brouillon informatique pour la Géométrie) [Laborde 95], est aussi un micro-monde qui a connu, et connaît encore un franc succès. Selon Laborde, ce micro-monde consiste en un monde d'objets et de relations, il y a un jeu d'opérateurs capables d'agir sur les objets, de créer de nouveaux objets avec de nouvelles relations. Ce système a pour objectif d'aider les élèves à mieux appréhender les concepts enseignés en géométrie. Son principe de fonctionnement est assez simple : l'enseignant seul ou avec ses élèves, construit une figure géométrique en explicitant certaines propriétés (par exemple, un point I au milieu d'un segment [AB]). Une fois la construction achevée, les élèves peuvent manipuler cette figure, c'est-à-dire modifier la position des points, agrandir ou réduire la figure, etc. et ainsi remarquer que les propriétés mathématiques sont conservées. La (**Figure 2.1**) montre par exemple que quelque soit la forme du triangle, les droites particulières (médiante, médiatrices et bissectrices) conservent leurs propriétés.

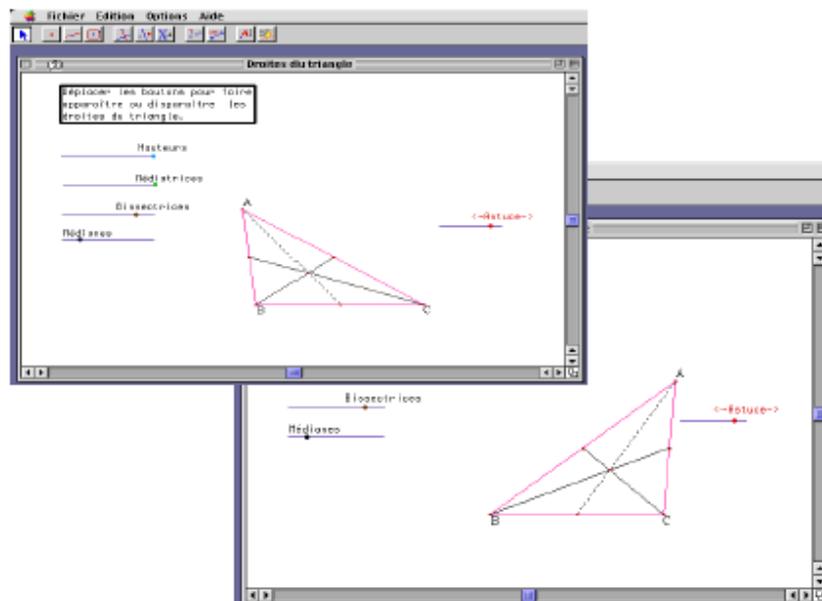


Figure 2.1. Exemple de manipulation d'un triangle avec le micro-monde Cabri-Géomètre

2.3.2 Systèmes Tuteurs Intelligents

L'évolution de l'intelligence artificielle dans le domaine de la résolution de problèmes et de la représentation des connaissances a permis d'ajouter une certaine dose d'intelligence dans les systèmes d'enseignement. On a tenté alors de produire des systèmes tuteurs qui pouvaient simuler un enseignant humain.

Les premières applications des techniques de l'IA dans ce domaine ont conduit à des systèmes capables de générer automatiquement des problèmes. Pour entraîner l'apprenant à faire des opérations d'addition par exemple, Uhr [Uhr 69] a implémenté un générateur de nombres aléatoires qui produit deux nombres que l'apprenant essaie d'additionner. Le résultat de l'apprenant sera ensuite comparé au résultat trouvé par la machine et jugé en conséquence. Les systèmes réalisés avec cette philosophie ont été qualifiés de systèmes générateurs (generative systems) et sont considérés comme les précurseurs des STIs qui sont apparus par la suite.

Un des premiers objectifs des STI était justement d'étendre le domaine d'application et augmenter la puissance de ces systèmes générateurs [Sleeman & Brown 82]. Le Système SCHOLAR [Carbonnel 70] qui enseigne la géographie de l'Amérique du sud, est le premier système apparu dans ce sens. Chose nouvelle à l'époque, SCHOLAR avait une représentation interne des connaissances qu'il devait enseigner, et non plus des textes d'exercices avec leurs solutions préenregistrées. SCHOLAR était alors capable d'entretenir une conversation à initiative mixte en langue naturelle avec l'apprenant.

Les premières recherches ont porté surtout sur la représentation des connaissances du domaine à enseigner et les mécanismes d'inférence. Les systèmes experts sont alors très vite apparus comme des outils privilégiés pour développer les STI [Nicaud & Vivet 88]. Par la suite, étant conscients qu'un bon expert dans une matière ne fait pas nécessairement un bon enseignant dans cette matière, les chercheurs ont focalisé leurs efforts sur l'individualisation de l'enseignement par une prise en compte du profil de l'apprenant et la compréhension de ses démarches et de ses erreurs. Un modèle de l'apprenant représentant toutes les connaissances de ce dernier sur le sujet enseigné (ce qu'il sait, ce qu'il ne sait pas, ce qu'il sait faire, ce qu'il ne sait pas faire, . . .) a donc été utilisé comme solution au problème. Ce modèle est créé et mis à jour par des stratégies tutorielles pour adapter au mieux le cours au niveau de compétence de l'apprenant. Beaucoup de systèmes intégrant ces deux aspects ont alors vu le jour. Nous citons en particulier :

- WEST [BrBu 82] enseignant le maniement des quatre opérations arithmétiques ;
- WHY [StCo 82] qui enseigne les conditions physico-climatiques de la culture du riz;
- BUGGY [Br&al 82] formant les instituteurs à l'analyse des erreurs des élèves concernant les opérations d'addition et de soustraction, en utilisant une bibliothèque des erreurs (bugs library);
- GUIDON [Clancey 82] enseignant le diagnostic médical, développé sur la base du célèbre système expert MYCIN.

En résumé, nous disons que les STI sont des logiciels très complexes à réaliser imposant la coopération entre spécialistes de disciplines aussi diverses que: Pédagogie générale, Psychologie cognitive, Didactique et Informatique. Néanmoins, les recherches effectuées pendant les deux décennies [1970-1990] ont pu délimiter les composants que doit inclure un STI en général [Wenger 87], [Nicaud & Vivet 88] : un module expert du domaine, un module pédagogique, un modèle de l'apprenant et une interface de communication avec l'apprenant (**Figure 2.2**).

Nous n'élaborerons pas un catalogue exhaustif de tout ce qui a pu être réalisé dans ces composants, mais nous mentionnerons certaines approches que nous jugeons essentielles pour notre travail.

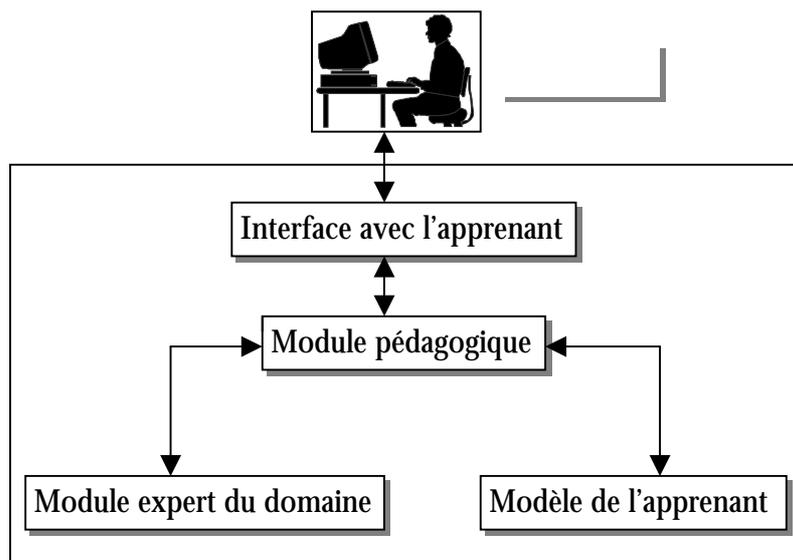


Figure 2.2. Architecture générale d'un STI

2.3.2.1 Module expert

a) Les fonctions du module expert

Le module expert remplit une double fonction [Wenger 87]. D'une part, il agit comme la « source des connaissances » à présenter à l'apprenant, cela inclut les explications et les réponses aux questions de l'apprenant. Et d'autre part, il sert comme « standard » pour l'évaluation des performances de ce dernier. Pour cette dernière fonction, il doit être capable de résoudre des problèmes dans le même contexte que celui de l'apprenant de façon à pouvoir comparer les deux solutions. Si l'objectif du tuteur est de guider l'apprenant dans la résolution d'un problème, le « module expert » doit être capable de générer tous les chemins possibles amenant à la solution. Ceci permettra de pouvoir comparer les étapes intermédiaires de la résolution du problème en vue d'évaluer l'apprenant et mettre à jour le modèle de ce dernier.

b) Les connaissances du domaine

Les formes principales de connaissances que l'on rencontre dans les différentes disciplines peuvent se décomposer en trois niveaux [Nicaud & Vivet 88]:

- Le niveau *objet* composé des connaissances factuelles (exemple: le pigeon est un oiseau) et des *règles de base* qui permettent d'effectuer des raisonnements en utilisant les connaissances factuelles (exemple: un oiseau est un animal qui vole);

- Le niveau *heuristique* composé des règles qui servent à guider le raisonnement (c'est à dire des *méta-règles* qui choisissent les *règles de base* à appliquer dans une situation donnée);
- Le niveau *mécanisme de raisonnement* (raisonnement déductif, raisonnement inductif, raisonnement par analogie, etc.).

La nature du travail de l'expert varie beaucoup en fonction des domaines et des types de problèmes à enseigner. Certains enseignements sont plus orientés vers *les connaissances factuelles*, d'autres vers les manipulations *de règles de base* (les causes de la pluie par exemple), d'autres encore vers *les heuristiques* (résolutions d'exercices, élaboration d'un diagnostic, jeu, etc.).

Cette variété de domaines induit des différences dans les techniques utilisées pour réaliser les *modules experts*. On y trouve, presque, toutes les représentations développées en intelligence artificielle (règles de production, réseaux sémantiques, frames, objets, etc.).

Certains modules experts sont capables d'expliquer leur raisonnement. On les appelle couramment *boîtes de verre*, par opposition aux autres, qualifiés de *boîtes noires* [Wenger 87]. Ce que nous entendons par *explication* dans un module expert d'un STI, c'est la capacité de reformulation et de justification d'une solution d'un problème, en indiquant les critères de sélection des règles nécessaires à la résolution du problème.

Le *module expert* de SCHOLAR par exemple qui repose sur un réseau sémantique, est considéré comme une boîte noire Le *module expert* de GUIDON, lui, est considéré comme une boîte de verre. Il s'appuie sur les règles de production du célèbre système expert de diagnostic médical MYCIN, adaptées pour le besoin de l'enseignement.. L'adaptation des règles est faite car le contexte d'enseignement à des étudiants est différent de celui de l'aide au diagnostic pour des médecins. Il manquait surtout des *connaissances utiles pour donner des explications et pour expliciter le raisonnement en des termes accessibles aux étudiants* [Clancey 82]. C'est pourquoi Clancey a dû modifier fondamentalement la base de connaissances de MYCIN afin de prendre en considération ces caractéristiques.

2.3.2.2 Modèle de l'apprenant

Les chercheurs dans le domaine des STI ont cherché assez tôt, à s'intéresser à la modélisation de l'apprenant. Le « modèle de l'apprenant » devrait fournir au système toutes les informations concernant les connaissances et les aptitudes de l'apprenant relatives à la matière enseignée. Ce composant doit, en plus, être capable de maintenir ces connaissances en analysant le comportement de l'apprenant au cours des sessions d'apprentissage.

Le processus permettant la construction et la maintenance du modèle de l'apprenant est appelé *modélisation de l'apprenant* (ou *diagnostique*) [Wenger 87]. Holmes et Hawkes [Holmes & Hawkes 94] suggèrent que la modélisation de l'apprenant doit faire face à trois types de problèmes:

1. le problème *d'allocation de crédit/débit* aux compétences élémentaires nécessaires à la résolution d'un problème;
2. le problème *d'élimination du bruit* (erreurs de frappe, réponses au hasard, etc.);
3. le problème *d'explosion combinatoire* résultant de la tentative du système à interpréter le comportement de l'apprenant à partir de toutes les actions primitives.

Les deux derniers points sont discutés dans [Holmes & Hawkes 94]. Quant au premier problème, il apparaît généralement quand un apprenant est appelé à résoudre un problème de grande importance (c'est à dire un problème dont la solution nécessite plusieurs étapes, ou bien, un problème dont la solution nécessite plusieurs compétences dans la matière). Quand un apprenant commet une erreur lors de la résolution du problème, une difficulté à laquelle le système doit faire face est de déterminer laquelle des compétences nécessaires à l'élaboration de la solution n'a pas encore été assimilée. Ceci se pose également dans le cas où l'apprenant arriverait à résoudre correctement le problème : faut-il assigner un crédit à toutes les compétences nécessaires à la solution, c'est à dire peut-on supposer que l'apprenant a assimilé toutes les compétences?.

Plusieurs méthodes sont utilisées pour résoudre ce problème. Les trois méthodes que nous citons ci dessous sont les plus citées dans la littérature. Nous trouvons une étude plus détaillée de ces méthodes dans [Lima 92].

a) Méthode à recouvrement (Overlay method)

Le *modèle à recouvrement* a été introduit par Carr et Goldstein [Carr & Goldstein 77] dans le système WUSOR-II. Dans ce paradigme, les unités de connaissances sont évaluées en faisant une comparaison entre le comportement de l'apprenant et celui du module expert du domaine. L'état des connaissances de l'apprenant est donc vu comme un sous-ensemble de l'état des connaissances du module expert. Cette modélisation peut être aisément implémentée quand la matière à enseigner est représentée par des règles de production. Le *modèle de l'apprenant* se construit alors progressivement en lui ajoutant pas à pas les règles qui sont jugées acquises [Lima 92].

Une première critique portée au modèle à recouvrement est que ce modèle ne tient pas compte du fait qu'un comportement incorrect de l'apprenant peut résulter d'une incorrection dans les connaissances conceptuelles (*misconception* en anglais) et non pas d'un *défaut* (*manque*) de connaissances de ce dernier. Un autre inconvénient est qu'il est impossible de prendre en compte les connaissances de l'apprenant qui sont correctes mais non prévues par l'expert.

b) Méthode d'identification de bogues

Cette méthode a été développée par Burton et Brown [Burton & Brown 78] qui portaient un regard critique au modèle à recouvrement. Ils considèrent que l'inconvénient majeur du modèle à recouvrement est de supposer que l'erreur de l'apprenant provient uniquement de l'ignorance de certains concepts du domaine, alors qu'en réalité, une erreur peut aussi résulter de *misconceptions* ou de confusions dans le raisonnement de l'apprenant (erreurs dans la connaissance procédurale de l'apprenant). Ils ont alors appelé *bogue* (en anglais *bug*) toute déviation des connaissances de l'apprenant par rapport aux connaissances correctes du domaine.

L'idée d'utiliser des *bogues* a été implémentée dans le système BUGGY [Burton 82], un logiciel dont le but est de former les instituteurs à diagnostiquer les causes des erreurs qui peuvent se produire lors de la résolution de problèmes arithmétiques d'addition et de soustraction par des élèves.

La philosophie utilisée dans BUGGY est de représenter explicitement aussi bien les *bogues* que les *connaissances correctes* du domaine. Burton et Brown affirment en effet, que les élèves appliquent consciencieusement les règles qu'ils possèdent mais que ces règles peuvent incarner l'erreur (règles modélisant les bogues). Connaître donc la règle utilisée c'est connaître la source de l'erreur [Palies 84].

La base de connaissances de BUGGY contient, en plus des connaissances expertes correctes, toute une bibliothèque d'erreurs élémentaires sur les opérations d'addition et de soustraction. A chaque erreur est associée une liste de conditions que doit vérifier un problème pour être un exemple de cette erreur. Le *modèle de l'apprenant* est alors formé d'un réseau procédural constitué des *connaissances expertes* (connaissances correctes connues par l'apprenant) et d'un ensemble de *bogues* (les connaissances incorrectes utilisées par l'apprenant).

Si l'avantage de cette méthode est la possibilité de connaître le type de l'erreur commise par l'apprenant et pouvoir y remédier, l'inconvénient reste cependant, l'impossibilité de prendre en

compte tous les comportements de l'apprenant, à moins que le catalogue des bogues soit exhaustif.

c) Méthode de modélisation différentielle

Cette technique a également été développée par Burton et Brown [Burton et Brown 82] dans le système WEST, un logiciel de jeu qui propose une course entre l'ordinateur et l'élève pour la conquête de l'Ouest. Ce jeu avait un intérêt éducatif double : enseigner les opérations arithmétiques (y compris l'utilisation des parenthèses) et enseigner les heuristiques du jeu. Le jeu consiste pour chaque joueur, à combiner trois nombres tirés au hasard par la machine, avec des opérateurs arithmétiques de son choix. Ceci détermine le nombre de cases (appelé *score*) dont il avance ou recule (on parle de *mouvement*) sur le chemin. Le jeu comporte des raccourcis de plusieurs types. Quand un joueur atteint la case de son adversaire, ce dernier recule de façon importante. Ces règles font que le jeu optimal n'est pas forcément obtenu avec le score le plus grand.

La modélisation différentielle utilisée dans WEST est à peu près semblable à la modélisation à recouvrement. Le mouvement joué par l'élève est comparé à un ensemble de mouvements alternatifs construits par le module expert dans la même situation de l'élève. La différence (d'où différentiel) entre la solution de l'élève (i.e l'expression mathématique choisie) et les bonnes solutions de l'expert doit permettre de fournir des hypothèses sur ce que l'élève ne connaît pas ou n'a pas encore maîtrisé. Le problème qui se pose est: comment assigner un débit à l'incapacité de l'élève à construire une solution parmi les "bonnes solutions" trouvées par l'expert? Une solution à ce problème, proposée par Burton et Brown, est d'imputer des débits à toutes les connaissances nécessaires à la construction de ces "bonnes solutions". L'inconvénient de cette solution est évidemment le fait de supposer que si l'élève n'utilise pas une connaissance de l'expert pour construire sa solution, alors c'est que l'élève n'a pas compris cette connaissance.

2.3.2.3 Module pédagogique

Ce module contient les connaissances sur la manière de communiquer les connaissances du domaine [Wenger 87]. Il est l'interlocuteur de l'apprenant, il dirige sa session, lui propose des exercices, sait expliquer et comprendre ses erreurs et les corrige. Bref, son rôle est de placer à chaque instant, l'apprenant dans une situation d'apprentissage optimale par rapport à lui-même, par rapport à ses connaissances et par rapport à l'objectif qui a été fixé. Cette situation peut être atteinte grâce à des mécanismes de guidage plus ou moins directifs et des mécanismes

d'explication qui dépendent du domaine, de l'apprenant et du contexte dans lequel se fait l'interaction [Nicaud & Vivet 88].

a) L'expertise pédagogique

Jusqu'à une date récente, l'idée de représenter les connaissances pédagogiques d'une manière explicite, n'a pas suscité autant d'intérêt que la représentation des connaissances du domaine [Wenger 87], [Nicaud & Vivet 88]. A quelques exceptions près, beaucoup de STI implémentent cette expertise sous forme procédural, ce qui leur donne beaucoup de rigidité. Un des objectifs des STI est d'explicitier cette expertise d'une manière déclarative pour la rendre disponible et évaluable. Les systèmes WHY [Collins 76] et GUIDON [Clancey 82] sont les pionniers des STI en matière de concrétisation de cette idée. Ils ont, tous deux, utilisé les règles de production comme moyen de représentation des connaissances pédagogiques. Voici deux exemples de telles règles (dites *règles tutorielles*) utilisées dans ces deux systèmes:

Dans le système WHY [Lelouche 84]:

SI l'apprenant explique sur un exemple un lien de causalité en faisant

intervenir un facteur non pertinent ;

Alors lui donner un contre exemple ayant une autre valeur de ce facteur.

Dans le système GUIDON [Clancey 82], [Baron 84]:

T-règle 2-04

Si 1. le nombre de facteurs apparaissant dans la règle à demander à

l'étudiant est **zéro** ;

2. le nombre de sous-buts restant à établir avant de pouvoir appliquer cette règle est **un** ;

Alors étape1. dites: "considérons le 'sous-but' . . ." ;

étape2. discuter le but avec l'étudiant selon le mode dirigé par
le but (procédure 001) ;

étape3. terminer la discussion de la règle considérée (procédure 017).

b) Conduite du dialogue pédagogique

Le problème fondamental au niveau pédagogique est de savoir quand intervenir, sur quel point le faire et quelle quantité d'informations fournir [Nicaud et Vivet 88]. Plusieurs approches ont été proposées comme solutions à ce problème. Dans [Boufaïda 95] a été définie une classification des principales méthodes qui ont été mises en oeuvre dans les STI disponibles actuellement. Quoique ces méthodes diffèrent dans leur conduite du dialogue pédagogique, elles ont cependant, une caractéristique commune: *l'initiative mixte*. Celle-ci consiste à autoriser l'apprenant à effectuer des interventions quand il le souhaite, y compris la possibilité de poser des questions quand le système attend de lui une réponse. Cette idée d'initiative mixte, introduite par Carbonnel dans le système SCHOLAR [Carbonell 70], a été reprise par la suite dans d'autres systèmes, surtout ceux orientés vers l'enseignement de *connaissances factuelles* ou de *diagnostic* tels que WHY et GUIDON, pour mener un enseignement souple et efficace. Nous retenons trois approches de conduite de dialogue utilisées dans différents STI : le « dialogue socratique », la « planification pédagogique » et le « guidage discret ».

b1) Le dialogue socratique

Dans le *dialogue socratique*, le STI n'a qu'un seul objectif : *mettre l'apprenant en contradiction avec lui-même lorsqu'il commet une erreur*. Ce type d'enseignement a pour but de découvrir les connaissances de l'apprenant en lui posant des questions et ce dans le but de faire justifier par l'apprenant lui-même ses hypothèses dans le contexte d'un cas spécifique. L'apprenant peut ainsi construire des principes généraux, découvrir des contradictions et extraire des inférences correctes à partir de faits connus [Lima 92].

Plus particulièrement dans WHY, A.Collins [Collins 76] a repéré vingt quatre *règles tutorielles* indépendantes du domaine pour contrôler l'interaction apprenant-système. Celles-ci décrivent les stratégies globales utilisées par le système afin de guider le dialogue. Elles permettent de décrire les modalités d'intervention, c'est à dire le moment, la manière et la cause de l'intervention. Certaines d'entre elles sont utilisées par exemple pour :

- Changer de sujet de conversation;
- Inciter l'apprenant à chercher des généralisations de cas;
- Présenter des contre-exemples au lieu de répondre tout simplement par "faux", notamment en cas d'erreur de raisonnement;
- etc.

Cependant, dans ce style d'enseignement, le tuteur ne possède pas d'objectifs pédagogiques indépendamment de l'apprenant. L'erreur de ce dernier est la ressource qui déclenche le dialogue socratique. Le tuteur se contente alors d'explorer les connaissances du domaine afin de remédier à l'erreur commise [Boufaïda 95].

b2) La planification pédagogique

Une autre approche aussi intéressante, mise en oeuvre dans d'autres STI, consiste en une *planification de l'enseignement* définie à l'aide d'objectifs pédagogiques. *La planification pédagogique* consiste en un programme d'actions (ou une séquence d'étapes) que le tuteur doit suivre pour satisfaire un objectif précis durant une session de travail.

Avec les didacticiels classiques, l'auteur spécifie un plan conditionnel unique qui détermine tous les cheminements possibles de l'apprenant. Ce type de planification reste insuffisant et inflexible pour conduire un dialogue continu et efficace, et dépend largement des compétences d'un auteur concevant son cours.

Avec les STI, la planification pédagogique repose généralement sur les *règles tutorielles*. Celles-ci utilisent des informations sur le *contexte courant* et sur la *session d'apprentissage précédente* de l'apprenant pour raisonner sur le choix du moment et du sujet à présenter à l'apprenant. Le système de Peachey et McCalla [Peachey & McCalla 86] en est un précurseur et utilise des méthodes de planification classiques de l'IA. Il intègre dans son architecture, entre autres composants, un *planificateur* générant un plan d'enseignement et un *exécuteur* interprétant ce plan. Un exemple de plan est représenté dans la (Figure 2.3).

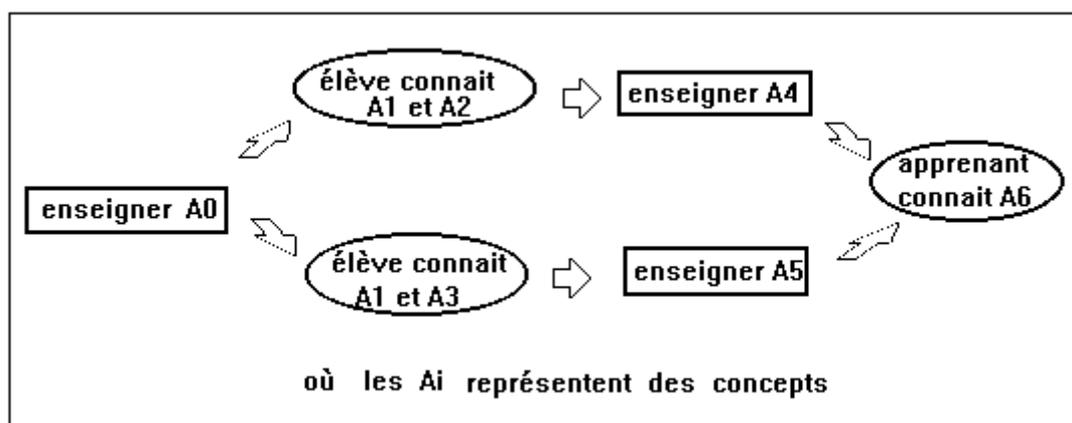


Figure 2.3. Un exemple de plan pédagogique

b3) Le guidage discret

Dans les domaines comportant des heuristiques pour guider le raisonnement, le mode d'enseignement de ces heuristiques a une grande influence sur le mode d'interaction. L'apprentissage des heuristiques s'effectue principalement de trois façons [Nicaud & Vivet 88] :

Par l'observation d'exemples : exemples de diagnostic, exemples de résolution d'un problème (dans un jeu à deux joueurs par exemple, le jeu de l'adversaire fournit de nombreux exemples);

Par l'expérience personnelle : effectuer des diagnostics, résoudre des exercices, jouer, etc.;

Par la présentation explicite : des heuristiques sont énoncées clairement par un expert.

Le guidage discret [Clancey 86] est particulièrement bien adapté à l'enseignement des heuristiques (plus spécialement l'observation et l'expérience). Ce mode d'interaction consiste à laisser l'apprenant agir, observer ce qu'il fait et n'intervenir qu'à certains moments bien choisis (généralement lorsque l'apprenant a commis plusieurs fois la même faute) pour enseigner l'heuristique pertinente. Le guidage discret a été implémenté dans les systèmes GUIDON et WEST.

2.3.2.4 Interface avec l'apprenant

Le problème de l'interface n'est en réalité, pas propre au domaine des STI, mais plutôt un problème qui relève du génie logiciel en général. L'interface utilisateur est en fait l'élément de référence qui permet à l'utilisateur de juger de la qualité d'un système. Un système qui offre une interface-utilisateur mal conçue est susceptible d'être rejeté quelles que soient les fonctionnalités qu'il offre [Sommerville 88]. Il n'est donc pas étonnant que l'interface apprenant soit très bien développée dès l'apparition des premiers STI. SCHOLAR, SOPHIE, GUIDON et bien d'autres systèmes contiennent une interface de compréhension du langage naturel (l'anglais particulièrement) plus ou moins limitée [Nicaud & Vivet 88].

C'est certainement dans le traitement du langage naturel que l'intelligence artificielle fût utilisée dans ce composant des STI. Le but est de donner plus de liberté à l'apprenant pour dialoguer avec le système. Mais les résultats espérés par les chercheurs dans ce contexte ne sont pas à vrai dire satisfaisants. D'abord, il y a la complexité du traitement du fait de l'ambiguïté inhérente au langage naturel. Et puis, il y a le problème qui se pose à l'apprenant lui même, du fait que le dialogue devient verbeux nécessitant beaucoup de frappe au clavier et une perte de temps pour la résolution de certaines ambiguïtés qui peuvent surgir [Sommerville 1988].

L'apparition des systèmes multifenêtres et des souris a permis à de nombreux STI de faire l'économie de la compréhension du langage naturel. Des menus nombreux, judicieux et rapidement obtenus offrent fréquemment un mode de communication suffisant et très rapide. C'est souvent ce que préfère l'apprenant, surtout en formation professionnelle [Nicaud & Vivet 1988].

Les interfaces graphiques obtenues par l'utilisation de ces techniques sont en train de prendre de plus en plus d'importance. Elles utilisent des moyens de désignation, tel le bouton de sélection de la souris, permettant la manipulation et la sélection des éléments d'interaction graphiques (menus déroulants, liste de choix, boutons radio, cases à cocher, icônes, etc.). Le résultat de ces actions est immédiat et l'on parle de style réactif, c'est à dire, la réaction en temps réel aux actions de l'utilisateur [Boufaïda 95].

2.4 Systèmes hypermédias éducatifs

Au cours de la seconde moitié des années 80, la technologie hypertexte/hypermédia a soulevé un certain enthousiasme dans le domaine de l'éducation [Delestre 00]. Beaucoup de manifestations scientifiques et de congrès montrent que cette technologie a suscité un intérêt croissant dans une communauté de plus en plus large, menant des travaux aussi bien sur les hypermédias proprement dits que sur leurs applications dans différentes disciplines. Cette section a pour but de présenter le domaine des hypermédias en général puis propose les modalités de leur application au domaine de l'éducation.

2.4.1 Hypertexte, Hypermédia, Multimédia

2.4.1.1 Terminologie

Le terme hypertexte a été introduit pour la première fois en 1965 par Ted Nelson [Nelson 65]. Nelson le définit alors comme étant « un ensemble de matériaux textuels ou picturaux interconnectés de telle façon qu'il serait impossible de les présenter ou de les représenter sur papier. L'hypertexte doit posséder des cartes et des index permettant l'accès aux différents documents qu'il contient... ».

La notion d'hypertexte évoque aujourd'hui principalement une organisation permettant la lecture non linéaire des documents sur l'initiative du lecteur [Nanard 95]. Toutefois, la racine *texte* dans le terme hypertexte n'a pas un caractère restrictif quant au contenu de l'hypertexte. Ce dernier n'est pas en effet limité au seul aspect textuel des documents, mais il concerne toutes les

formes de documents, y compris bien entendu les formes multimédias (son, image, vidéo, etc.) [Garzotto 94].

Le terme d'hypermédia a été introduit, surtout par la presse, pour insister sur l'aspect multimédia de certaines applications hypertextes. Il ne recouvre scientifiquement pas plus que le terme hypertexte [Nanard 95]. Il faut noter toutefois que, même dans la communauté scientifique, le terme hypermédia tend à se répandre en raison de sa vulgarisation.

Quant à la notion de multimédia, elle concerne l'aptitude pour une machine (et une application) de gérer plusieurs canaux de communication sensorielle avec l'utilisateur (usuellement le son, l'image, la vidéo, éventuellement même des mouvements ou des efforts dans le cas des réalités virtuelles). Cependant, la notion de multimédia n'implique pas celle d'hypermédia. La présence d'une image animée n'est pas le garant d'une véritable interactivité entre l'utilisateur et la machine. Il est parfaitement possible d'inclure dans un document linéaire une image, un fond sonore et/ou une séquence vidéo. Ceci en fera un document multimédia mais n'en fera en aucun cas un hypermédia [Nanard 95].

Le premier système hypertexte est apparu en 1968 : le système NLS (pour oN Line System) conçu par Douglas Englebart [Delestre 00]. Ce système avait pour objectif de permettre à l'utilisateur de pouvoir archiver toutes sortes de documents et de les relier par des liens associatifs. Ce système est l'aboutissement d'une dizaine d'années de recherche autour du projet *Augmentation System*, qui devait permettre aux utilisateurs d'accroître leurs capacités cognitives et intellectuelles [Englebart 62]. C'est autour de ce projet que beaucoup de concepts de l'informatique moderne ont été inventés comme celles de : traitement de texte, aide contextuelle, interactivité, interface graphique fenêtrée, souris, touches de fonction et courrier électronique.

Dès lors, plusieurs systèmes utilisant les techniques hypertextes sont apparus par la suite sur le marché. On peut citer entre autres [Delestre 00]:

- Le système Dynabook, conçu et développé par Alan Kay de la société Xerox à Palo Alto, logiciel basé sur une interface graphique et permettant de sauvegarder tout type d'information (texte, son, image, etc.).
- Le système Hypercard conçu en 1986 par Bill Atkinson, figure emblématique de la société Apple.
- Le WWW (World Wide Web) qui forme le plus grand hypermédia jamais conçu sur le réseau Internet.

2.4.1.2 Modélisation des hypermédias

Les hypermédias peuvent être définis suivant trois points de vue : du point de vue *structurel*, du point de vue *fonctionnel* (c'est à dire celui de l'interaction entre l'utilisateur et le système) ou bien du point de vue *sémantique* [Delestre 00].

Définition structurelle : Tel que défini dans [Balasubramanian 00] un hypermédia est un système composé de nœuds et de liens. Les nœuds peuvent être composés d'informations textuelles ou multimédias, tels que des graphiques, des images, des animations, des vidéos ou bien des programmes informatiques. Les nœuds sont reliés les uns aux autres par des liens. On distingue les nœuds qui sont à l'origine du lien (on parle de référence) et les nœuds qui sont les destinations des liens (on parle de référent). Les liens peuvent être plus ou moins complexes : ils peuvent être unidirectionnels, permettant d'aller d'une page à une autre page ou bien bidirectionnels, afin de faciliter le retour au point de départ. Ils peuvent aussi être typés afin de spécifier la sémantique du lien. Ce sont ces liens qui définissent l'architecture du système, que l'on nomme hyperespace.

Définition fonctionnelle : L'hypermédia peut être considéré comme étant un procédé informatique permettant d'associer une entité (souvent minimale, c'est-à-dire un mot, une portion d'image ou une icône) à une autre entité (souvent plus étendue comme un paragraphe détaillé, une image ou une page). Ce mécanisme permet donc à l'utilisateur de naviguer librement dans l'hypertexte. En activant à l'aide d'une souris, une zone du document qui est l'origine d'une association, l'utilisateur peut immédiatement atteindre une autre partie du document. Il n'est donc plus obligé de suivre le cheminement prévu par l'auteur, il définit son parcours en fonction de ses envies et de ses centres d'intérêt. Cette propriété de l'hypermédia en fait un document "interactif" dans lequel le lecteur tient une place prépondérante.

Définition sémantique : Comme l'explique Nanard [Nanard 95], le terme hypertexte évoque "plus que texte", sans toutefois que le mot "plus" soit interprété comme "plusieurs" textes interconnectés, mais comme texte plus un élément d'une autre nature qui, en terme informatique, est appelé "connaissance". Un hypermédia est donc une « entité composite » composée de deux types d'entités : les documents et les connaissances.

Les documents sont des entités destinées à l'homme, ils constituent l'unique surface d'échange entre l'homme et l'hypertexte. L'homme les voit tels quels et les interprète directement. Les documents, par leur nature, véhiculent une très forte quantité d'informations que la machine

restituée à la demande du lecteur. Durant l'interaction avec l'hypertexte, l'utilisateur en consulte un puis un autre, il navigue. La navigation est certes à l'instigation de l'utilisateur mais elle ne peut se faire qu'en fonction des propositions de destinations que lui offre l'hypertexte.

La connaissance (au sens formel et calculable) est un ensemble de relations spécifiquement destinées à la machine qui ont pour rôle de lui permettre de décider ce qui doit être proposé au lecteur comme suite possible de sa lecture en fonction du contexte courant. La représentation la plus pauvre de la connaissance est le lien inscrit à l'intérieur même des documents, c'est ce que Nanard nomme le lien "en dur" qui relie deux documents. A contrario, la forme la plus élaborée de cette connaissance peut être générée par un système complexe se basant sur une modélisation du domaine et sur une modélisation de l'utilisateur.

Les deux entités (documents, connaissances) étant connectées par un mécanisme dit d'ancrage. L'ancrage est le moyen d'associer librement la connaissance aux documents. Son sens usuel est l'attachement d'information à l'extrémité d'un lien (c'est à dire la mise en relation de deux informations portées par les documents).

Une bonne métaphore pour comprendre la nature réelle des hypermédias [Nanard 95] est d'imaginer un ensemble de documents accrochés à un mur et protégés par une feuille de plastique transparent. La connaissance ancrée sur ces documents serait matérialisée par des annotations, des lignes colorées reliant des parties de documents tracées au marqueur effaçable sur ce plastique. Il est alors possible de faire évoluer cette connaissance sans altérer les documents. Le matériau composite constitué des documents, du plastique qui assure l'ancrage (le positionnement) et des dessins faits sur le plastique a typiquement la structure caractéristique des hypermédias.

2.4.1.3 Avantages des Hypermédias

L'utilisation d'hypermédias a trois avantages principaux :

Facilité d'utilisation et de conception : Du côté de l'utilisation d'hypermédias l'utilisateur n'a pas à apprendre un langage d'interaction avec le système, ni les différentes fonctionnalités de celui-ci pour l'utiliser. Par exemple, cette facilité permet de faire une recherche d'informations sans passer par un langage de requête. Du côté de la conception, le concepteur voit au fur et à mesure ce qu'il fait et peut modifier localement ou globalement n'importe quel aspect de son système (le programme ou l'interface).

Liberté de choix : à chaque étape de l'utilisation de l'hypermédia l'utilisateur effectue le choix du prochain nœud à voir. Ce choix peut être "sémantique" : l'utilisateur clique sur un bouton en fonction de sa signification. Il peut aussi être "syntaxique" : l'utilisateur clique sur un bouton en fonction de sa fonction (nœud suivant, précédent, retour au départ, chapitre suivant, etc.). Ainsi les hypermédiats permettent de définir un contexte et des accès différents pour une même connaissance.

Buts flous : la grande nouveauté des hypermédiats comme "outils cognitifs" c'est de permettre aux utilisateurs d'avoir des buts mal définis. Dans l'activité de l'exploration (ou de la navigation), c'est en fonction des réponses du système que l'utilisateur va cerner progressivement son problème. Les hypermédiats sont susceptibles d'être intégrés dans des utilisations où la tâche de l'utilisateur est "occasionnelle", mal connue ou mal définie par celui-ci et où le but peut être cerné par approximations successives : aide, recherche d'informations, prise de décision, apprentissage, collaboration, etc.

Ayant précisé les termes hypertexte et hypermédia, nous allons maintenant nous intéresser aux modalités d'utilisation de la technologie hypermédia dans le domaine de l'apprentissage.

2.4.2 Hypermédias et Enseignement/Apprentissage

La modélisation des hypermédiats sous la forme composite (documents/connaissances) est particulièrement pertinente pour comprendre le rôle et l'intérêt des hypermédiats dans les techniques d'enseignement/apprentissage médiatisé par ordinateur, tant du côté de l'enseignant que de celui de l'apprenant [Nanard 95] :

L'auteur (l'enseignant) en construisant un hypermédia, organise un réseau de relations entre des informations ; il les valorise en leur greffant des connaissances additionnelles. Il crée aussi des documents pour constituer le contenu de l'hypermédia.. Ces documents ont une valeur informationnelle intrinsèque indépendamment de leur appartenance à l'hypermédia. Par contre, la richesse de l'hypermédia qui est responsable de l'aspect dynamique de l'exploration, dépend essentiellement de la structure de connaissances qui est ajoutée à ces documents et permet de les rapprocher. Le choc de la juxtaposition d'informations a souvent un impact plus important sur l'apprenant que le contenu du document lui-même.

L'apprenant, au cours de son activité d'exploration de l'hypermédia, met en jeu implicitement deux mécanismes : la *lecture des documents* (le plus souvent immuables), et le *montage des informations* (en explorant la structure hypermédia) [Balpe 90 in Nanard 95]. Chaque document est donc lu et

interprété dans un contexte qui ne prend sens qu'en fonction de ce qui a déjà été vu. Un même document observé au cours de deux cheminements distincts aura le plus souvent des interprétations différentes chez l'apprenant. De plus, l'apprenant peut aussi enrichir un hypermédia de ses propres annotations. Il met en évidence des relations nouvelles entre des informations déjà présentes dans l'hypermédia. En ancrant de nouvelles relations entre des documents, il matérialise sa connaissance du domaine ainsi que la synthèse qu'il a réalisé des informations qui lui ont été présentées.

2.4.2.1 Manipulations de l'information dans un hypermédia

Les hypermédias, bien que répondant tous au modèle fonctionnel document/connaissance, ils diffèrent largement en fonction de l'usage qui est fait de cette connaissance et de la finalité de leur mise en œuvre par l'utilisateur (auteur/lecteur). D'ailleurs le style d'interface de l'hypermédia est lui-même fortement dépendant de l'usage qui en est attendu. Il y a en gros trois façons de manipuler l'information dans un hypermédia :

- Extraire l'information brute à partir de bases de documents ;
- Valoriser (ou organiser) l'information existante pour mieux la présenter, faciliter son accès et augmenter son impact chez le lecteur ;
- Produire de nouvelles informations ou de nouvelles structures de connaissances.

Ces attitudes face à l'hypermédia ont conduit à des types de systèmes hypermédias fondamentalement différents dans la mesure où le besoin à satisfaire chez l'utilisateur enseignant ou apprenant est fondamentalement différent.

a) L'extraction d'informations

L'hypermédia est un outil remarquable d'extraction et de recherche d'informations. Par son style d'interaction basé sur les documents, il est plus naturel pour les novices que les systèmes à base de requêtes. Dans ce type d'approche, l'organisation de la connaissance se reflète dans la surface de l'hypermédia par la présence de nombreux documents de style tables des matières facilitant une navigation en profondeur. L'hypermédia destiné à la recherche d'informations est un outil pédagogique extrêmement flexible. Avec son style d'interaction directe, il peut être utilisé de deux façons :

a1. L'exploration libre « vagabondage » : dans laquelle l'utilisateur feuillette l'hypertexte comme on feuillette sans but initial une encyclopédie. Ceci est très caractéristique des premières

prises de contact des utilisateurs avec ces types de systèmes, par exemple le world wide web (WWW).

a2. La recherche intentionnelle d'informations : c'est le type d'emploi finalisé des hypermédias de cette catégorie. Elle a un rôle formateur extrêmement important, puisque, en obligeant l'apprenant à se concentrer sur un seul but précis et à résister aux tentations de digressions qui sont nombreuses lors d'une navigation, ce type d'exercices le conduit certainement aux informations recherchées.

b) La valorisation de l'information

Par valorisation, nous entendons une transformation de l'information destinée à la rendre mieux assimilable par l'homme. En établissant des relations entre des informations existantes grâce aux techniques pédagogiques de structuration de l'information, l'auteur valorise l'information. Les documents proposés aux lecteurs conservent le plus souvent le même contenu informationnel mais leurs interrelations matérialisées par les connaissances de l'hypermédia, facilitent leur interprétation par le lecteur.

En tant qu'outil pédagogique, les hypermédias basés sur la valorisation de l'information sont surtout utilisés en phase de lecture, ce qui est leur finalité. Mais il semble important d'insister sur le rôle formateur important que constitue pour un apprenant la construction d'hypermédia. Cette activité créatrice nécessite chez l'apprenant auteur d'abord de maîtriser la base informationnelle sur laquelle il construit l'hypermédia, mais surtout développe chez lui, le sens de l'observation, de l'évaluation, et des facultés de jugement pour décider quels sont les points à valoriser dans son document. Etre auteur c'est d'abord être capable de se mettre à la place du futur lecteur.

c) La production des informations et des structures de connaissances

Le troisième type d'activité de l'utilisateur se servant d'un hypermédia est de faire émerger de nouvelles informations, de nouvelles connaissances. Les systèmes destinés à ce type d'activité sont souvent appelés des systèmes de jardinage d'informations. Comme un jardinier avec ses plantes, le jardinier d'informations fait pousser de nouvelles informations et de nouvelles structures à partir d'informations existantes. Il commence à travailler sur un ensemble d'informations non hypertextualisé, puis par observation et explicitation des relations entre les informations conduit à l'émergence de structures qui s'ancrent sur les documents. Ces relations connectant des éléments informationnels, les rapprochent et donnent à leur couple un sens nouveau. Il est bien connu que la juxtaposition d'informations est susceptible de générer un sens

qui n'est explicite dans aucune d'elles. Cette démarche très classique de tout auteur qui se documente dans une bibliothèque, rapproche des informations, en déduit de nouvelles..., est désormais plus qu'une démarche intellectuelle. Elle est une suite d'actions, prises en charge par un système qui les aide, en garde trace, et surtout les conserve directement sous forme d'une structure hypertexte réexploitable.

Le jardinage d'informations vise à rendre explicites des connaissances implicitement portées mais non apparentes dans un ensemble de documents. Le temps et l'évolution sont donc fondamentalement deux caractéristiques majeures de cette activité qui vise la compréhension profonde des informations et l'explicitation de connaissances. Contrairement aux approches relatives à la valorisation d'informations dans lesquelles l'objet principal est le produit fini, ici l'objet principal est le processus mental qui conduit le jardinier d'informations à constater l'émergence d'une nouvelle idée. Le premier sait où il va et donc son travail n'est que le moyen pour atteindre le résultat visé. Le second par contre ne le sait pas et donc en cela que le jardinage d'informations appartient aux activités créatrices.

Les quelques systèmes hypermédiés orientés vers le jardinage d'informations Aquanet, Viki, Sepia, Dolphin, Macweb sont des outils d'édition interactive dont l'emploi facilite toute l'activité de construction incrémentale d'hypermédia.

Ces systèmes, offrant un mode auteur réellement convivial apportent une aide réelle aux enseignants pour organiser les informations et permettent surtout de passer de façons continue de la phase d'émergence des idées au document pédagogique où les informations sont valorisées.

Mais, c'est surtout du côté de l'apprenant que l'apport des outils de jardinage d'informations semble le plus profitable. Lui offrir un outil qui lui permette une démarche d'essai erreur, et lui permette de s'appuyer réellement sur les résultats des étapes intermédiaires pour avancer plus vite dans sa création est, en effet, une activité salutaire dans la formation. En faisant prendre conscience de la réalité du processus mental de sa démarche créative à l'apprenant, on libère sa spontanéité, sa créativité profonde, on lui apprend à penser, à créer librement, et non à savoir seulement mettre en œuvre des méthodes stéréotypées.

2.4.2.2 Avantages éducatifs des hypermédiés classiques

Deux grands atouts, issus de la structure intrinsèque des hypermédiés, émergent de leur utilisation dans un cadre éducatif : la composante multimédia et la composante hypertexte.

Plusieurs études ont essayé d'évaluer l'intérêt des systèmes multimédia dans le cadre des systèmes d'enseignement. Ainsi (Hoogeveen 95) a dégagé quelques « critères » (par exemple ce qu'il nomme « Level of Multimediality », « Level of Man-machine Interactivity », et « Level of Congruence») permettant d'évaluer les qualités d'un logiciel éducatif multimédia interactif. D'après cette étude, il conclut que l'utilisation d'un système interactif multimédia peut améliorer l'aspect visuel et ludique, et par conséquent renforcer l'intérêt de l'apprenant.

Outre la composante multimédia des hypermédias, la composante hypertexte peut aussi grandement améliorer la qualité de l'enseignement. En effet, les hypermédias, par leur structure, aident l'apprenant à mieux se représenter la connaissance, à mieux appréhender les tenants et les aboutissants de chaque concept. La non-linéarité de la progression de l'apprenant l'oblige à se construire sa connaissance en créant des connections entre les concepts. En effet, l'apprentissage comme la pensée ne se font pas par des idées isolées mais par des relations significatives ou associatives entre idées. Donc l'hypermédia devient un outil de structuration de la pensée.

2.4.2.3 Inconvénients éducatifs des hypermédias classiques

Malheureusement ces deux avantages peuvent devenir préjudiciables, puisqu'ils peuvent entraîner une désorientation et une surcharge cognitive chez l'apprenant [Rhéaume 93].

La désorientation est issue de la facilité que possède l'apprenant à se déplacer de nœud en nœud dans le système hypermédia. Ainsi cette liberté de déplacement peut finir par troubler l'apprenant. Il risque de se poser des questions du type «Où suis-je ?», «Pourquoi suis-je là ?» «Que dois-je faire ?». [Rhéaume 93] explique que ceci est principalement dû à notre mémoire à court terme, puisque comme l'a montré [Miller 56], les êtres humains ne sont capables de mémoriser sur le moment qu'un nombre limité d'informations (sept items à plus ou moins deux près).

La surcharge cognitive, quant à elle, est provoquée par «l'avalanche d'informations» que risque de «déverser» le système. En effet, la redondance, pour être bénéfique, doit être construite de façon intelligente. En aucun cas, il ne faut présenter la même information à l'aide de différents médias ne nécessitant pas tous le même niveau de connaissance.

De ce fait, on a cherché à guider l'étudiant dans son cheminement en fonction de ses connaissances sur le domaine enseigné, en modifiant aussi bien le contenu des pages que les liens entre ces dernières : c'est ce que l'on nomme les hypermédia adaptatifs, objet de la section suivante.

2.4.3 Hypermédias adaptatifs et enseignement/apprentissage

Des recherches ont donc essayé de minimiser l'aspect négatif des hypermédias classiques, en créant des Systèmes Hypermédias Adaptatifs (SHA). L'objectif de ces systèmes est d'adapter la présentation de la connaissance et d'aider l'apprenant à se diriger dans l'hyperespace. De ce fait, dans un hypermédia nous devons pouvoir modifier aussi bien le contenu des pages que les liens entre les différentes pages. Mais c'est surtout sur l'adaptation des liens que le plus grand nombre de techniques ont été développées [Brusilovsky 98]. On trouve par exemple des techniques de guidage direct, des techniques d'ordonnement des liens, des techniques de masquage des liens ou bien encore des techniques d'annotation des liens.

L'architecture des hypermédia adaptatifs, comme pour la plupart des systèmes d'enseignement médiatisé par ordinateur, s'appuie principalement sur deux modèles : le modèle du domaine et le modèle de l'apprenant. Les différents types de systèmes hypermédias adaptatifs se caractérisent par la relation qu'ils entretiennent entre le modèle du domaine et les médias utilisés pour présenter les concepts à l'apprenant. Les systèmes ont successivement employé différentes techniques (par exemple la méthode des pages d'index ou la méthode d'indexation fragmentée). Mais la technique la plus évoluée calque la structure de l'hyperespace sur la structure du modèle du domaine [Vassileva 97]. Ainsi chaque concept est relié à une ou plusieurs pages physiques et ces relations sont représentées par des liens hypertextes.

2.4.3.1 Technologies d'adaptation dans un système hypermédia adaptatif

Un document hypermédia typique, nous le rappelons, se compose d'un ensemble de noeuds (ou pages) reliés par des liens. Chaque page contient de l'information locale et un certain nombre de liens vers d'autres pages locales ou distantes. Le système hypermédia peut également inclure les outils spéciaux de navigation tels que la table des matières, l'index, et la carte de navigation qui pourraient être employés pour orienter vers toutes les pages accessibles.

Ce qui peut être adapté dans l'hypermédia c'est :

- la page elle même (adaptation au niveau du contenu de la page) ;
- l'aspect et le comportement des liens (adaptation de la structure).

Dans la littérature de l'hypermédia adaptatif ils sont référencés respectivement par : « présentation adaptative » et « support adaptatif de navigation ».

a) La présentation adaptative

Le but de la présentation adaptative est d'adapter le contenu d'une page hypermédia aux objectifs de l'apprenant, à son niveau de connaissances et à toute autre information stockée dans son modèle. Il peut y avoir des raisons multiples d'utiliser la présentation adaptative. Deux cas typiques du côté éducatif sont : « les explications comparatives » et « l'explication variante ».

a1. Les explications comparatives

L'idée de l'explication comparative est de relier le contenu de la page à présenter avec le niveau de connaissance de l'apprenant. Une page peut avoir un ou plusieurs fragments d'explication comparatifs cachés, qui vont être comparés avec les nouveaux concepts à présenter selon le modèle de l'apprenant. Par exemple, la boucle "while" en langage C++ pourrait être comparée à la même construction en Pascal. Seulement les étudiants qui ont des connaissances du langage Pascal verront la comparaison.

a2. L'explication variante

L'idée de l'explication variante est d'utiliser différentes explications dont chacune correspond à un certain niveau de connaissance du sujet présenté. Un système peut enregistrer plusieurs variantes d'explication pour différentes parties du contenu de la page hypermédia. Avec cette technique de présentation, les utilisateurs experts reçoivent l'information la plus détaillée et la plus profonde (complexe), alors que les débutants reçoivent une explication plus générale et moins compliquée. Un exemple d'application de la présentation adaptative dans les SHA sur le Web est utilisée dans le système C-Book [Kay & Kummerfeld 94].

b) Support adaptatif de navigation

Il s'agit de modifier les liens qui relient les différentes pages de la matière à enseigner. L'adaptation structurelle peut se présenter sous la forme *d'ajout de nouveaux liens* à une page comme c'est le cas dans la plupart des systèmes d'aide à la navigation sur le Web comme par exemple le système Brodway [Gonschorek & Herzog 95]. Une autre forme d'adaptation structurelle consiste à *cache des liens* existants afin de réduire la complexité de l'espace navigable, c'est à dire que le système essaye de cacher à l'apprenant les liens qui ne lui sont pas appropriés à l'heure actuelle, comme c'est le cas dans le système AHA [De Bra & Calvi 98]. D'autres formes consistent à *effacer des liens* existants ou à *inhiber certains liens* [Brusilovsky & Pessin 94]. Une autre forme d'adaptation structurelle consiste en *l'annotation de liens* où le système ajoute aux liens une certaine forme de commentaires, qui peuvent indiquer à l'apprenant plus d'information au sujet des pages derrière

les liens annotés. Ces annotations incluent habituellement les graphismes, couleurs de fonte, tailles et les styles. Une dernière forme d'adaptation structurelle consiste à *ajouter des pages d'index* comme c'est proposé dans [Gilbert & Han 99].

Le but du support adaptatif de navigation est d'aider l'apprenant à trouver son chemin d'accès dans l'hyperespace en adaptant la présentation et les fonctionnalités des liens aux objectifs visés par l'apprenant, à son niveau de connaissance, et à d'autres caractéristiques tirées de son modèle-apprenant.

Un certain nombre de formes et d'applications intéressantes de la présentation adaptative et du support adaptatif de navigation ont été développées depuis le début des années 90. Cette technologie est mise en application dans le système Elm-Art [Brusilovsky & al 96]. Une étude plus complète peut être trouvée dans [Brusilowsky 98].

2.4.3.2 Avantages des hypermédias adaptatifs

Les hypermédias adaptatifs représentent une avancée non négligeable vis à vis des hypermédias classiques. En effet, les différentes techniques utilisées permettent à l'apprenant d'être guidé dans son apprentissage, et aux enseignants de mieux structurer leurs connaissances (surtout lorsque la structure de l'hypermédia est calquée sur la structure du modèle du domaine). Ainsi, sans toutefois annihiler la liberté de navigation intrinsèque aux hypermédias, l'apprenant est constamment guidé dans son cheminement. De même, le fait de distinguer la connaissance des outils qui permettent de la présenter éclaircit le travail de l'enseignant. Ce dernier peut alors mieux structurer son travail, en pensant tout d'abord à l'organisation des connaissances, et ensuite à la façon de les exposer.

2.4.3.3 Inconvénients des hypermédias adaptatifs.

Cependant, quelques problèmes persistent. Tout d'abord, l'accent a surtout été mis sur l'adaptation des liens, afin de guider l'apprenant dans son cheminement. Or la deuxième composante de l'adaptation, c'est-à-dire l'adaptation du contenu, a souvent été mise de côté. Pourquoi ? Tout simplement parce que la méthodologie de développement de ces systèmes ne s'y est pas réellement prêtée. En effet, bon nombre de systèmes hypermédias adaptatifs sont issus de systèmes hypermédias classiques déjà définis, auxquels les chercheurs ont ajouté des outils d'adaptation. Or, alors qu'il est assez aisé de cacher des liens, ou bien de les annoter, il est beaucoup plus difficile de remplacer un item d'une page, ou bien de modifier la structure d'une page. Ensuite, l'uniformisation du système est apparue comme un facteur très important. D'un

point de vue ergonomique, il est important que tous les cours aient la même structure. Enfin tout comme un enseignant, il faut que le système puisse utiliser immédiatement toute nouvelle connaissance. Ainsi, si une personne trouve ou construit un nouveau média en rapport avec un des concepts enseignés, le fait de l'ajouter doit permettre au système d'enrichir instantanément les cours sur ce concept.

Ainsi depuis quelques années, les recherches s'orientent vers ce que l'on nomme les Systèmes Hypermédias Adaptatifs Dynamiques (SHAD), c'est-à-dire des systèmes qui créent dynamiquement les pages et les liens de l'hyperespace en fonction des caractéristiques de l'apprenant, en vue de simplifier l'espace exploré et de le réduire à la partie la plus adaptée à l'utilisateur. Ces SHAD font l'objet de la section qui suit.

2.4.4 Hypermédias adaptatifs dynamiques.

Afin d'améliorer la qualité de l'adaptation et de prendre en compte instantanément de nouvelles données, depuis quelques années, les recherches se sont orientées vers les hypermédias adaptatifs dynamiques.

La principale caractéristique de ces systèmes est d'offrir un hypermédia virtuel [Vassileva 95]. Le système n'est pas constitué de pages et de liens prédéfinis : ils sont construits dynamiquement. L'architecture de ces systèmes repose sur quatre composantes principales que sont : le modèle du domaine, le modèle de l'élève, une base de données de matériaux pédagogiques (ou teaching materials) et un générateur de cours. Le modèle du domaine, comme pour la dernière génération des hypermédias adaptatifs, permet de définir l'architecture globale du système. Il y a par conséquent adéquation entre les nœuds du modèle du domaine et les pages de l'hypermédia virtuel, ainsi qu'entre les relations du modèle du domaine et les liens de l'hypermédia virtuel.

L'utilisation d'un tel système apporte plusieurs avantages. Tout d'abord l'adjonction d'un nouveau support peut être immédiatement prise en compte, puisque encore une fois, les pages du système sont construites dynamiquement. Ensuite, les enseignants ne sont pas obligés de penser à la façon d'agencer les différents médias, ils doivent juste définir l'architecture générale du système (le modèle du domaine) et déterminer, récupérer ou créer les matériaux pédagogiques qui vont servir à présenter chaque concept. Une bonne synthèse de ces hypermédias adaptatifs dynamiques et un exemple de système (Metadyne) utilisant cette technologie avancée se trouve dans [Delestre 00].

Les hypermédias adaptatifs dynamiques reposant sur la technologie XML a fait également l'objet de la thèse de Magister de notre collègue de l'université de Batna Mme Behaz Amel [Behaz & al 03] et qu'elle continue actuellement dans le cadre de sa thèse de Doctorat.

2.5 Intégration des technologies des hypermédias et des tuteurs intelligents

L'analyse critique que nous avons réalisé sur le paradigme d'«hypertexte » et celui de «système expert» qui est la technique de base utilisée par les tuteurs intelligents nous a révélé un certain nombre d'insuffisances pour chacun d'eux mais aussi des points forts.

Un système expert est très limité dans son niveau d'expertise, long à développer, rigide et difficile à modifier, alors qu'un hypermédia est théoriquement illimité dans son expertise, développé rapidement et ses modifications et mises à jour sont faciles ;

Un système hypermédia, au vu de sa liberté et sa flexibilité, est réputé par les deux problèmes de désorientation et de surcharge cognitive qui fait que l'utilisateur se perd souvent dans la chaîne de liens qui lui sont proposés et part souvent dans des liens inutiles en perdant de vue son objectif initial, alors qu'un système expert est réputé par le fait qu'il guide pertinemment l'utilisateur vers son objectif.

Les groupes de recherche dans le domaine du e-learning se sont rendu compte qu'implémenter un système d'apprentissage sur le web en se basant sur un seul des deux paradigmes est donc insuffisant. De nombreux systèmes d'apprentissage mariant les bénéfices des deux techniques ont alors été développés.

Le but le plus important d'un STI, nous le rappelons, est de prendre en charge un apprenant au cours d'une session d'apprentissage. Aussi il est apparu clairement que l'hypermédia fournit la meilleure option pour organiser la matière à enseigner en ligne. Une combinaison d'un STI et une matière à enseigner organisée comme hypermédia était un point de départ pour la recherche sur les tuteurs intelligents hypermédias sur le web. Un certain nombre de groupes de recherche s'est indépendamment rendu compte qu'un système hypermédia adaptatif couplé à un STI offre plus de fonctionnalités d'adaptation et de guidage à l'apprenant à distance qu'un système hypermédia statique traditionnel. Nous appelons ces nouveaux systèmes des « Systèmes Tuteurs Intelligents Hypermédias » (STIH).

Outre les techniques d'adaptation appliquées dans les SHA cités plus haut (« présentation adaptative » et « support de navigation adaptatif »), d'autres adaptations hérités du domaine des

STI sont appliquées dans les STIH (planification des enchaînements des unités d'apprentissage, analyse intelligente des solutions de l'apprenant, support interactif de résolution de problèmes et support de collaboration) [Brusilovsky 98]. Nous allons les aborder ci-après.

2.5.1 Planification d'enchaînement

Le but de la « planification » est de fournir à l'apprenant l'enchaînement le plus approprié des unités d'apprentissage hypermédias (uah) et des tâches à accomplir (tests, exercices, problèmes, etc.) pendant une session d'apprentissage, et ce, dans le but d'atteindre les objectifs visés.

Nous distinguons deux types d'enchaînement des uah [Brusilovsky 99b]:

- L'enchaînement à niveau élevé ou encore l'enchaînement de la connaissance (*knowledge sequencing*) qui détermine le prochain concept ou sujet à enseigner à l'apprenant.
- L'enchaînement à niveau bas ou encore l'enchaînement de tâches (*task sequencing*) qui détermine la tâche suivante à exécuter (exemple, simulation, problème, test, etc.) pour maîtriser le sujet ou concept en cours d'apprentissage.

Dans le contexte de l'éducation sur le Web, le planificateur devient très important pour guider l'étudiant dans l'hyperespace d'informations disponible sur le Web. Le planificateur a été mis en application dans différents systèmes tels que : ELM-ART [Brusilovsky & al 96), CALAT [Nakabayashi & al 97), InterBook [Brusilovsky & al 96b) et DCG [Vassileva 97).

2.5.2 Analyse intelligente des solutions de l'apprenant

Les analyseurs intelligents des réponses de l'apprenant peuvent indiquer ce qui est exactement erroné ou incomplet dans la réponse de l'apprenant et indiquent la connaissance manquante ou incorrecte qui peut être responsable de l'erreur. Des exemples récents d'application de l'analyse intelligente des solutions de l'apprenant sont Camus II [Vanneste 94] et Elm-PE [Weber & Möllenberg 95].

2.5.3 Support interactif de résolution de problèmes

Le but du support interactif de résolution de problèmes est de fournir à l'apprenant une aide appropriée à chaque étape de résolution du problème. Les systèmes qui mettent en application cette technologie peuvent observer les actions de l'apprenant, les comprendre, et essayer de fournir l'aide appropriée au moment voulu et mettre à jour le modèle de l'apprenant. La technologie de support interactif de résolution de problèmes n'est pas aussi populaire dans les SHA sur le Web que dans les STI autonomes parce que, jusqu'ici, les applications serveur web ne

sont pas assez interactives pour supporter et observer les actions de l'apprenant et fournir l'aide sur chaque étape. Chaque interaction entre le navigateur et le serveur peut prendre un temps remarquable, et la condition de l'exécuter sur chaque étape peut ruiner le processus de résolution du problème. La situation changera probablement quand la technologie de Java arrive à une certaine maturité.

Trois systèmes démontrent la faisabilité de cette technologie sur le Web : PAT-Online adoptant l'architecture CGI [Brusilovsky & al 97], D3-WWW-Trainer [Faulhaber & Reinhardt 97] et ADIS [Warendorf & Tan 97] employant la technologie Java pour supporter la vraie interactivité.

2.5.4 Support adaptatif de collaboration

Le support adaptatif de collaboration est une technologie très récente des STI sur le web. Le but du support adaptatif de collaboration est d'utiliser les connaissances des différents utilisateurs sur un certain sujet (enregistrés dans les modèles d'apprenant) pour constituer un groupe collaboratif ou pour trouver le pair le plus compétent pour répondre à une question sur un sujet (c'est à dire. trouvant une personne avec un modèle montrant la bonne connaissance de ce sujet).

2.6 Exemples de STI sur le web

2.6.1 Le Système CALAT

CALAT [Kiyama & al 97] est un STI sur le Web. En utilisant un navigateur web standard sur le poste client (Internet Explorer ou Netscape Navigator par exemple), les apprenants peuvent se connecter au serveur de CALAT qui leur fournit des cours adaptés à leur niveau de connaissance set selon leurs objectifs à atteindre.

Le serveur de CALAT (**Figure 2.4**) se compose d'un processus « démon web » et d'un STI principal. Quand l'apprenant désire se connecter à CALAT pour un apprentissage, il envoie une requête à travers une URL comportant son identificateur (ou compte). Un mécanisme d'identification d'utilisateur est présenté pour maintenir la correspondance entre le processus du STI à générer et l'apprenant correspondant à travers le web. Un processus de STI est alors invoqué pour la prise en charge de cet apprenant en examinant le compte reçu dans l'URL.

Selon le niveau de compréhension et les unités d'apprentissage acquises par l'apprenant, CALAT détermine le prochain ordre d'affichage des pages (ou des unités d'apprentissage). Trois

types de pages HTML sont disponibles dans le cours de CALAT : des pages de cours (des pages d'explication), des pages d'exercice ou des QCM et des pages de simulation.

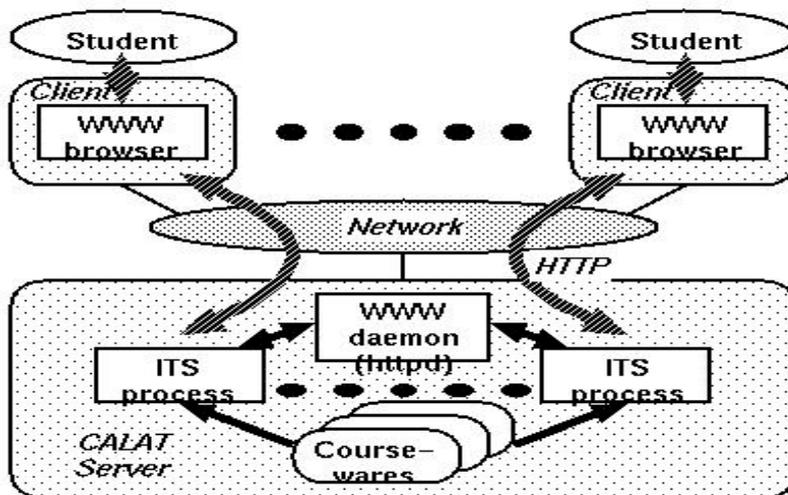


Figure 2.4. Architecture de CALAT

Dans le système CALAT, il est possible d'utiliser des pages HTML résidant sur d'autres serveurs web, ce qui signifie que n'importe quelles pages HTML résidant sur d'autres serveurs web peuvent faire partie d'un cours de CALAT. Chaque sous but du cours lui est associé l'URL d'une page de cours bien précise qui se trouve sur le serveur de CALAT ou sur d'autres serveurs web. Quand l'apprenant envoie une requête au serveur de CALAT, ce dernier détermine le sous but à expliquer et répond alors avec un script JAVA définissant le chemin d'accès à la page de cours spécifiée. Le navigateur web du poste client exécutera le script JAVA pour afficher la page de cours (Figure 2.5).

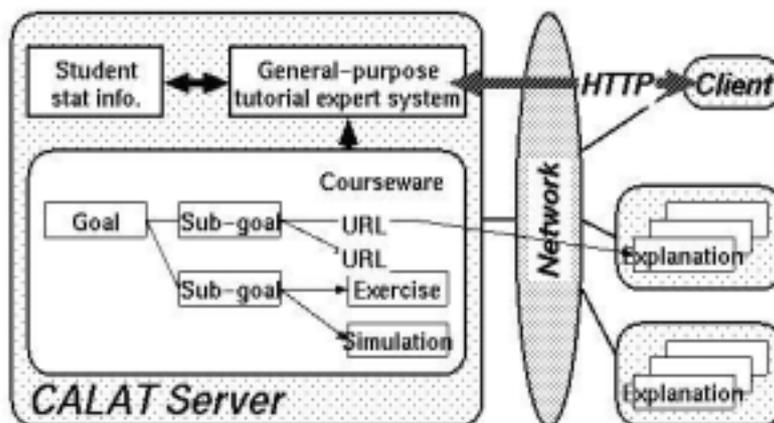


Figure 2.5. Processus de génération de pages de CALAT

Les pages d'exercice ou les QCM comportent trois types de questions : choix unique (Vrai ou faux), choix multiple, et description libre (texte libre). La page de test est un document HTML généré dynamiquement à partir d'un modèle de QCM ou d'exercice. Chaque question dans le modèle d'exercice est associée à un sous but. Quand l'étudiant soumet des réponses fausses, le STI met à jour le modèle d'apprenant, et lui donne le choix soit de refaire le test, soit de le guider vers les pages du cours associées aux sous buts qui correspondent aux tests réalisés.

2.6.2 Le système ADIS

Le STI d'animation de structure de données ADIS (An Animated Data Structure Intelligent Tutoring System) [Warendorf & Tan 97] est un STI développé comme aide pédagogique à la compréhension des structures de données telles que les listes chaînées, les piles, les files d'attente, les arbres et les graphes (**Figure 2.6**).

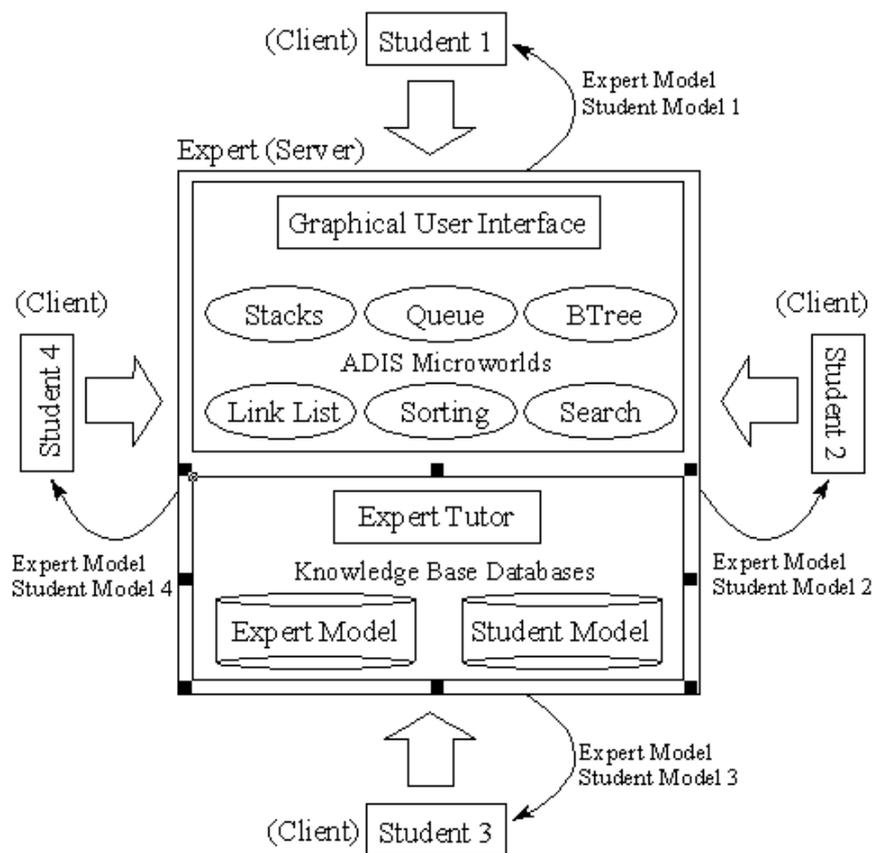


Figure 2.6. Architecture logicielle du système ADIS

ADIS a la possibilité d'afficher des structures de données graphiquement sur l'écran d'ordinateur et de permettre la manipulation graphique de ces structures créées. ADIS comporte un module tutoriel intégré d'exercices, grâce auquel les apprenants peuvent apprendre les

algorithmes de base (insertion, suppression etc.) sur les structures de données visuellement. ADIS est implémenté en Java et intégré avec une interface utilisateur graphique, pour agir avec l'apprenant.

Le système se compose de plusieurs modules de base (**Figure 2.6**) qui sont relativement indépendants les uns des autres, et ceci dans le but de permettre son amélioration et sa réutilisation facile pour d'autres domaines d'enseignement. Nous trouvons un module expert, un module pédagogue, un modèle apprenant et une interface avec l'apprenant

Chaque utilisateur a sa propre copie du modèle expert et du modèle étudiant. Un moteur d'inférence est implémenté indépendamment de la matière à enseigner pour permettre la réutilisation du système pour d'autres domaines autres que les structures de données.

2.7 Conclusion

Ce chapitre a mis en évidence les différents concepts et les méthodes utilisés dans la réalisation des tuteurs intelligents et des hypertextes dédiés à l'apprentissage. Nous avons commencé par décrire notre problématique qui concerne le guidage de l'apprenant à distance à travers les plateformes d'EAD. Nous avons ensuite décrit séparément les deux technologies adoptées dans le processus d'apprentissage à travers les différents EIAH : les hypermédias et les tuteurs intelligents -basés en grande partie sur les techniques des systèmes experts-, avant de proposer de les intégrer ensemble pour bénéficier de leurs avantages et éviter les inconvénients de chacune des techniques. Nous avons en effet constaté que :

- Un système expert est long à développer et rigide, alors qu'un hypertexte est développé rapidement, et ses modifications et mises à jour sont faciles.
- La liberté et la flexibilité offertes par les systèmes hypertextes font que souvent l'utilisateur se perd dans la chaîne de liens et part dans des liens peu importants en perdant de vue son objectif.
- Le paradigme de l'hypertexte manque du mécanisme d'inférence logique nécessaire pour valider les actions de l'apprenant et le guider
- Une limitation des applications hypermédias classiques est qu'elles fournissent le même contenu de page HTML et le même ensemble de liens à tous les apprenants malgré leur différente connaissance et intérêts.

Au terme de cette étude, nous pouvons dire qu'au fil des années, la recherche sur les systèmes d'enseignement utilisant les techniques des hypermédias s'est constamment affinée. Les chercheurs ont tout d'abord utilisé des hypermédias classiques et ont fini par utiliser des hypermédias adaptatifs dynamiques combinant les hypertextes classiques avec les techniques des STI. Cependant, pour l'instant les systèmes utilisant cette dernière technique sont assez rares, ils utilisent beaucoup plus la composante hypertexte que la composante multimédia des hypermédias, et utilisent rarement Internet comme mode de transmission de l'information.

Quoique nous avons présenté les hypermédias au sens large, nous indiquons que tous les concepts et techniques présentés restent valables dans le contexte du world wide web (WWW). Cette application liée à l'Internet est en effet le plus grand hypermédia imaginable à l'heure actuelle. Les documents qui forment les nœuds de l'hypermédia peuvent en effet se trouver sur n'importe quel serveur sur la planète et sont accessibles grâce à leur URL.



Chapitre 3

Systèmes auteurs – de l'autonomie à l'ouverture sur le web

3.1 Introduction

Un système auteur peut être défini comme un « environnement de développement logiciel de haut niveau ». Le concept de système auteur, dans un but d'enseignement/apprentissage, est apparu à la fin des années 70. Grâce aux avancées technologiques considérables de l'informatique dans les années 80, le multimédia a donné un bon coup de pouce aux systèmes auteurs [Nanard 96]. Parmi les exemples industriels les plus connus de ce type de logiciels, citons Authorware, Toolbook, HyperCard, Quest, IconAuthor ou encore Director.

Avec un système auteur, en principe, le temps d'apprentissage et de développement d'un logiciel multimédia est beaucoup inférieur au temps requis en utilisant un autre système de programmation. Nous pouvons penser par exemple à la création d'un bouton, émettant un son lorsque nous cliquons dessus, avec le logiciel Director.

Un informaticien de base, ne connaissant pas le logiciel Director a mis environ 2 minutes pour réaliser cette tâche. Il aurait mis au moins 15 minutes pour réaliser l'équivalent en langage C. Un non informaticien, a mis 15 minutes environ avec Director, alors qu'avec le langage C, il aurait fallu qu'il apprenne les bases (fonctions, boucles, compilations, etc.), et leurs syntaxes. Cela nécessitera plusieurs heures voire des semaines de travail [Wu Yao Kuang 00].

Si les systèmes auteurs ne sont pas initialement destinés au seul monde éducatif, on peut considérer que la réalisation de tutoriels est l'un de leurs débouchés principaux, au point que certains en arrivent parfois à définir les « systèmes auteurs » comme des outils de développement de didacticiels. Dans la diversité des produits, et face à une évolution technologique que l'on sait de plus en plus rapide, celui qui pense utiliser un système auteur dans une optique éducative, ne peut que se sentir démuni, et hésiter d'autant plus à prendre une quelconque décision quant au choix du système qui lui convient.

Un état de l'art sur les systèmes auteurs éducatifs multimédias, que nous désignons par la suite par le sigle SA, est nécessaire car il nous permettra de :

- Comprendre la manière dont un SA est conçu,
- Prendre connaissance des variétés des SA existants actuellement,
- Définir les critères de sélection d'un SA,
- Comprendre la conception d'un logiciel éducatif à travers un SA,
- Savoir quelle influence le fantastique essor du Web a-t-il sur les SA.

Autant de questions auxquelles nous espérons avoir des réponses permettant de nous guider pour la conception de notre propre système.

3.2 Définitions d'un système auteur

La réponse donnée dans les FAQ (Frequently Asked Questions) sur Internet (<http://home.tiac.net>) lorsque l'on pose la question « qu'est-ce qu'un système auteur ? » est : « un système auteur est une application disposant d'éléments préprogrammés et utilisés pour développer des produits multimédias ».

« What is an Authoring System? »

An Authoring System is a program which has pre-programmed elements for the development of interactive multimedia software titles ».

Le terme multimédia a été adopté simplement à cause du fait qu'il est difficilement concevable aujourd'hui d'avoir des programmes destinés à des utilisateurs finaux informaticiens ou non, n'utilisant que des données textuelles, et donc ne profitant pas des différentes possibilités et progrès offerts par la technologie multimédia [Gouyet & al 95]. En revanche, multimédia implique aussi travail d'organisation et de programmation supplémentaire pour et le concepteur et l'utilisateur d'un système auteur.

Cette définition peut être complétée par celle d'Ackerman [Ackerman 92] : « un système auteur permet à des gens qui ne savent pas programmer, ou qui veulent gagner du temps dans la programmation, ou intéressés par la programmation, de créer des logiciels ».

Mais ces définitions nous paraissent vagues. Aussi, nous allons donner notre définition du terme système auteur par rapport au contexte de notre étude et donc celle qui sera adoptée dans la suite de notre thèse : « un système auteur est un environnement disposant d'éléments préprogrammés et utilisés pour développer des logiciels multimédias éducatifs »

Afin de pouvoir profiter des progrès effectués en multimédia, un système auteur nécessite au moins la configuration suivante :

- Un écran graphique couleur ;
- Une grande capacité de mémoire vive ;
- Une grande capacité de disque dur pour le stockage de fichiers images et vidéos ;
- Des périphériques spéciaux comme une carte son, des enceintes, un scanner, etc. ;
- Des logiciels de traitement d'images, de graphes et de sons.

3.3 Objectifs d'un système auteur

Pour inciter les individus ayant des compétences dans divers domaines à créer des logiciels, il faut mettre à leur disposition des outils complets, simples et utilisables par des auteurs n'ayant en principe aucune compétence informatique [De La Passardière 85]. Par conséquent, étudier les problématiques dans le but d'utiliser un système auteur, signifie entre autres, chercher et prendre en considération tout critère pouvant mettre l'utilisateur dans une mauvaise situation.

Plusieurs attributs susceptibles d'influencer une méthode de développement, quel que soit le type et le niveau de l'utilisateur du système auteur, doivent être étudiés soigneusement et considérés par le concepteur. Nous allons présenter ci-dessous les critères que nous considérons comme principaux, d'après [Wu Yao Kuang 00] :

- *La convivialité* : La première approche avec un logiciel a un rôle psychologique fondamental pour tout utilisateur, même chez les informaticiens confirmés. En effet, si l'écran ou la présentation du logiciel n'est pas soigné, l'individu peut être plus ou moins bloqué pour la suite. L'importance de l'interface homme-machine justifie la généralisation des icônes et l'intégration du multimédia dans les systèmes auteurs. La transposition graphique, au sens visuel, est incontournable quel que soit le domaine, pour celui qui veut projeter son idée, faire comprendre, expliquer, fabriquer ou modéliser [Bourron & al 95]. Le concepteur doit par conséquent attacher une attention particulière à la présentation de son produit.

- *La transparence* : La gestion des données, et le fonctionnement interne du point de vue relation entre les différents éléments (variables, objets, etc.) doivent être complètement transparents à l'utilisateur. Certes, la sélection des données à inclure dans le logiciel construit à partir du système auteur lui est réservé, mais s'il doit y avoir des compressions (fichiers images par

exemples) ou autres traitements (emplacement sur disque des données), l'utilisateur ne doit pas se soucier ni du moment ni de l'endroit, ni du type de fonctions et librairies à inclure et à lancer pour l'exécution de ces diverses tâches. En effet, si l'utilisateur doit comprendre le fonctionnement des registres de données, la notion de système auteur n'a plus lieu d'être !

- *L'assistance* : Comme un système auteur est supposé, entre autres, être un outil de programmation pour les non informaticiens, il est fondamental d'assurer un minimum d'aide en ligne ou au moins une partie explicative des principales tâches ou commandes ou icônes du système. Même pour des tâches devenues banales comme la saisie de données textuelles, le système doit indiquer au moins à l'utilisateur dans quelle fenêtre et quand il peut le faire.

- *L'interactivité* : La communication entre le système auteur et son utilisateur, souvent sous la forme d'une assistance, test et contrôle automatique des tâches de l'utilisateur, doit être particulièrement soignée car elle est vitale pour l'utilisateur et pour la réussite sur le marché du produit. Un système auteur doit intéresser son utilisateur et non l'ennuyer.

- *La fiabilité* : Le système doit être le moins bloquant possible quelle que soit l'action menée par l'utilisateur sur le logiciel. Ce dernier doit pouvoir revenir en arrière, modifier ou supprimer sans difficulté tout composant de son logiciel. De même, il doit pouvoir assembler les pièces qui constituent son application dans un ordre pas trop contraignant, indépendamment les uns des autres et ce à n'importe quel moment du développement.

3.4 Outils de développement d'applications éducatives multimédia

Les outils de développement d'applications éducatives multimédias sont à classer dans deux catégories [Goffinet 98]. Tout d'abord, dans un premier temps lors de la conception d'une application, on utilise des outils de création et de manipulation d'un média particulier (soit un son, soit une image, etc.). Ceux-ci servent à manipuler une donnée ayant un type bien précis. Par exemple, on écrit à l'avance ses textes avec Word, on dessine ses graphiques avec Canvas ou Designer, on retouche ses images avec Photoshop, etc. Tous ces outils sont ici repris sous l'appellation « d'outils monomédia ».

Ces données de différents types sont ensuite assemblées dans une présentation multimédia. C'est là qu'interviennent les systèmes auteurs. Ils servent en quelque sorte à lier le tout, en intégrant les données préalablement créées avec la première catégorie d'outils. Ce sont eux les véritables outils « multimédias ».

3.4.1 Outils de création monomédia

Les outils monomédia principaux sont classifiés en fonction des médias qu'ils permettent de traiter. La plupart du temps les systèmes auteurs ne permettent pas de faire des retouches à un média particulier, il faut alors retravailler la donnée avec l'outil qui a servi à la créer.

Si ceci peut sembler un désavantage à première vue, cela a par contre le mérite de ne pas alourdir inutilement les systèmes auteurs. Et puis, chaque concepteur de système auteur ne doit pas réinventer tous les outils existant par ailleurs. Cette modularité permet aussi de réutiliser des outils déjà maîtrisés antérieurement (quel éditeur de texte ou éditeur d'image pourrait-il être plus rapide et plus agréable à apprendre que son propre traitement de texte ou son propre traitement d'image?).

3.4.1.1 Traitement du texte

On retrouve dans cette catégorie les outils travaillant sur le texte. Bien souvent hélas, le formatage complexe que l'on peut apporter à un texte dans son logiciel de traitement de texte n'est pas importable dans le système auteur. Il faut donc se contenter d'utiliser ces outils comme de simples éditeurs. On a aussi intérêt à passer ses textes au travers de formats partagés par tout le monde : ASCII, RTF, ANSI, etc.

Encore une petite remarque : si on veut par après porter son logiciel multimédia d'un ordinateur à un autre ou sur une autre plate-forme, il n'est jamais garanti que la police de caractères que l'on a choisie pour ses textes soit disponible sur la machine sur laquelle on est passé. La solution peu pratique, mais inévitable, est alors de transformer tous ses textes en graphiques bitmaps, afin que l'affichage reste le même sur toutes les plates-formes.

Parmi les produits phares, citons pour mémoire *MS-Word*, *WordPerfect* (Mac et PC).

3.4.1.2 Traitement du son

On retrouve dans cette catégorie les outils travaillant sur le son. Ils permettent l'enregistrement, la digitalisation, le mixage et autres modifications de sons enregistrés. La taille d'un son digitalisé peut vite devenir colossale (par exemple en haute-fidélité, avec un échantillonnage de type CD-audio à 44 Khz), alors il faut parfois savoir mesurer l'utilité de certaines vocalises. Citons quelques outils de travail dans le domaine du son : *SoundEdit Pro* (Macintosh), *Wham*, *WaveEdit* (PC).

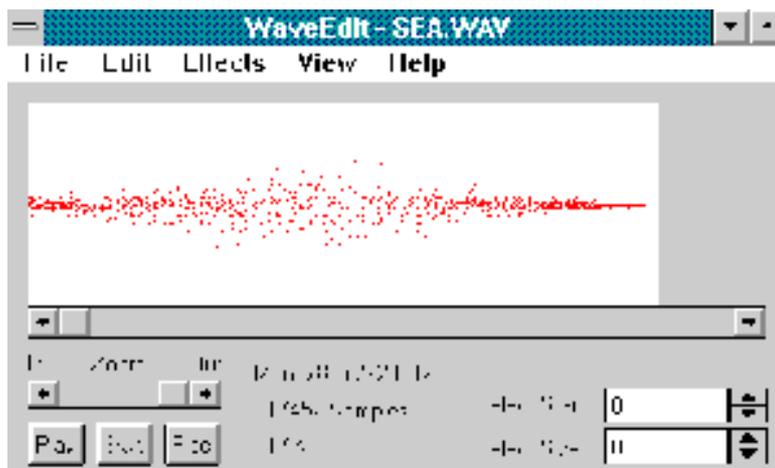


Figure 3.1. Un son édité avec WaveEdit

3.4.1.3 Traitement de l'image

On retrouve dans cette catégorie les outils travaillant sur l'image. L'image peut être soit *bitmap*, c'est-à-dire décrite par une matrice de points, soit *vectorielle*, c'est-à-dire qu'elle est décrite par un système de vecteurs. Les avantages du système vectoriel sont, premièrement, sa concision en place mémoire (on ne stocke que la description de l'image, pas l'image elle-même) et, deuxièmement, sa précision (un dessin agrandi ou réduit ne perd rien de sa résolution). Néanmoins, son inconvénient majeur est sa gourmandise en temps de calcul : à chaque affichage de l'image, il faut recalculer la position de chaque objet qui la compose.

Cette dernière restriction explique que, pour des raisons de performances, la majorité des systèmes auteurs travaillent surtout avec des images bitmaps. Il existe en effet dans tous les systèmes d'exploitation des routines d'affichage à l'écran optimisées pour afficher rapidement des matrices de points. Il faut donc penser à préparer une version bitmap de ses graphiques si l'on travaille en vectoriel. La question ne se pose pas évidemment pour les images scannées, puisque celles-ci sont toujours représentées dans un mode bitmap.

Parmi les outils les plus connus du domaine, citons pour les outils orientés dessin bitmap : *PhotoShop*, *PaintBrush*, *PhotoMagic* (PC), et pour les outils orientés dessin vectoriel : *FreeHand*, *Canvas* (Mac), *CorelDraw*, *Designer* (PC).

3.4.1.4 Traitement de la vidéo

On retrouve dans cette catégorie les outils travaillant sur la vidéo. La vidéo en elle-même peut être de deux natures : soit *analogique* (comme une image de télévision), soit *digitale*. Quasiment tous

les systèmes auteurs sont capables de jouer des séquences vidéo digitalisées, par l'intermédiaire le plus souvent du système d'exploitation ou de routines externes manipulant un format courant de vidéo digitale. La vidéo analogique, elle, ne peut s'incruster à l'écran que via une carte spéciale sur l'ordinateur, et elle est moins supportée par les systèmes auteurs. On a donc souvent peu le choix : il faut digitaliser ses films ou ses séquences d'animation. A l'heure actuelle, ce processus demande l'acquisition de cartes de digitalisation supplémentaires.

Les formats vidéo les plus courants sont *QuickTime* (Mac et PC), *AVI* (PC), *MPEG*.

3.4.1.5 Format des données

Il ne faut pas oublier que chaque média est enregistré dans un fichier, et que celui-ci est codé selon un certain format. Ce format n'est pas toujours universel, et chaque plate-forme a ses standards pour chaque type de média. Ainsi, souvent, les systèmes auteurs n'accepteront pour un type de média précis que certains formats. Bien souvent ces formats standards sont ceux qui sont propres à la plate-forme sur laquelle ils tournent. Il est donc important de travailler dans les bons formats, ceux de son système auteur.

Si les outils utilisés sont suffisamment puissants, ils permettent en général de travailler avec plusieurs formats hétéroclites. Si ce n'est pas le cas, on a alors besoin d'un outil supplémentaire pour les conversions de format. Le moyen le plus simple pour ne pas se compliquer la vie est de travailler, si possible, sur la même plate-forme tout au long du développement d'un produit multimédia.

3.4.2 Systèmes auteurs : outils d'intégration multimédia

Il y a une décennie, si l'on voulait réaliser un logiciel multimédia interactif, il fallait être un programmeur spécialisé pour pouvoir exploiter les bibliothèques de son langage de programmation favori. Cette situation n'a heureusement pas perduré, et l'on a depuis mis au point de nombreux systèmes auteurs pour aider à la création d'applications éducatives multimédias.

3.4.2.1 Niveaux de puissance

Pour pouvoir bien apprécier les possibilités offertes par un système auteur particulier, nous allons détailler les niveaux de puissance offerts aux concepteurs d'applications multimédia par les systèmes auteurs existants. Le premier niveau est celui que nous appellerons le niveau de la *table de montage*. Le second niveau est celui du *scripting* et le troisième celui des *commandes externes*.

- ***La table de montage***

C'est le premier niveau auquel le concepteur peut travailler, c'est-à-dire intégrer ses médias et définir leurs interactions mutuelles avec l'utilisateur final, sans devoir passer par un langage de programmation. Le concepteur se trouve devant son écran et définit de façon interactive lui-même son application multimédia. Les tables de montage peuvent être plus ou moins évoluées. Certaines permettent tout juste de lier les blocs de média entre eux, tandis que d'autres vont plus loin dans l'art de décrire les interactions entre les différents médias.

A ce niveau-ci, tout le monde quasiment peut travailler. Quiconque opère couramment avec des logiciels bureautiques un peu complexes peut se mettre à créer une application simple, manipulant des médias différents sans interactions complexes. Nous détaillerons plus loin les types de tables de montage offerts par les systèmes auteurs.

- ***Le scripting***

C'est le niveau auquel on peut écrire, dans un langage de haut niveau fourni avec la plupart des systèmes auteurs, des procédures qui peuvent être déclenchées suite à l'occurrence d'un événement particulier. Ces procédures permettent de spécifier un comportement plus complexe en réaction à une action bien précise de l'utilisateur. Ces procédures sont des scripts. Ils apportent puissance et souplesse à l'application.

Par exemple, suite au clic d'un utilisateur sur un bouton présent à l'écran, on voudrait qu'il y ait un dialogue avec cette personne, au travers d'une question posée et de la saisie d'une réponse. Selon cette réponse, des écrans différents seraient proposés. Pour réaliser cette séquence apparemment simple, il faut passer par un *langage de scripting* qui permet de définir des variables, de tester leur contenu et de décider des branchements en fonction de ceux-ci. Le résultat de tout ceci est une procédure qui est soit compilée, soit interprétée au moment de l'exécution du programme par le système auteur.

Les langages de scripting sont la plupart du temps bien plus simples que les langages de programmation. Les instructions de scripting s'écrivent dans un langage proche de la langue naturelle (même si la logique, elle, reste informatique) et les contraintes syntaxiques sont fortement réduites. Ceci a permis à plus d'une personne non-spécialiste de rédiger des scripts dans de nombreuses applications multimédia.

Le père spirituel de beaucoup de langages de scripting reste *Hypertalk*, inventé dans les années 80 pour piloter des applications créées avec Hypercard. Beaucoup de langages de scripting

s'inspirent de ses fondements : langue assez naturelle, programmation événementielle, en couches (par exemple : script d'un bouton lance un appel script carte script fond script pile Hypercard).[6] Quelques membres de la famille : *SuperTalk* pour Supercard, *Lingo* pour Director, *OpenScript* pour Toolbook

Ce niveau de travail permet de réaliser la plupart des applications multimédia simples que l'on rencontre couramment (bornes de présentation, etc.). Il reste accessible à des non-spécialistes, à condition bien sûr de s'investir un minimum dans l'apprentissage du langage de scripting. Il offre le meilleur rapport qualité du produit/coût de réalisation. C'est pour cette raison que les systèmes auteurs sont très souvent utilisés pour réaliser rapidement des prototypes de logiciels multimédias (ou non), étape qui demande d'avoir une bonne idée de l'aspect final d'un logiciel et qui doit se faire à un coût temporel et humain minimal.

- ***Les commandes externes***

Il arrive parfois que l'on atteigne les limites d'un langage de scripting. Par exemple si l'on veut que des procédures fassent des choses non prévues par le système auteur, telles que : afficher des dialogues non-standards, faire des effets spéciaux qui sortent de l'ordinaire, dialoguer avec d'autres applications, etc. Il est évident qu'un langage de scripting, s'il veut rester simple et accessible au commun des mortels, doit se limiter à un jeu d'instructions clair et concis. Ceci ne fait pas toujours l'affaire des gens qui ont des besoins bien particuliers.

Il faut alors se rabattre sur un langage de programmation *externe* au système auteur, rédiger ses procédures dans ce langage et les compiler. Ce sont les *commandes externes*. Au moment de l'exécution de l'application multimédia, ces commandes sont appelées par leur nom dans des scripts écrits avec le système auteur et puis exécutées.

Cette possibilité de travail est offerte dans la plupart des systèmes auteurs, mais elle ne touchera que les personnes les plus expertes dans l'art de la programmation. Tout le monde ne peut s'y aventurer sans risques. Néanmoins, cette fenêtre ouverte sur le monde de la programmation est vitale pour certaines applications. D'où le fait que la puissance offerte par les systèmes auteurs ne peut être appréciée sans cette possibilité d'appel à des programmes externes.

Ces trois niveaux offrent une grille assez complète d'estimation de la puissance de ce que l'on peut attendre d'un système auteur. Evidemment, plus on explore et plus on descend dans les possibilités offertes, plus la mise en oeuvre de ce que l'on veut faire est complexe. Il faut bien savoir que l'on ne pourra pas créer des applications multimédia révolutionnaires au travers d'une

table de montage. Mais pour des besoins standardisés, on peut espérer s'en tirer sans devoir s'offrir les services d'un programmeur éprouvé.

3.5 Taxinomie des systèmes auteurs

Pour permettre de situer ce que peuvent faire les systèmes auteurs, il ne suffit pas de les classer selon le niveau de puissance qu'ils offrent, il faut aussi les distinguer par le type d'applications qu'ils permettent de réaliser au mieux. Généralement, des systèmes auteurs de puissance comparable se prêtent plus à certains types de réalisations que d'autres. La typologie de classement proposée ici repose sur la métaphore de la table de montage employée par chacun des systèmes auteurs. En effet, tous les systèmes auteurs possèdent au minimum le niveau de la table de montage, et, souvent, des systèmes auteurs utilisant la même métaphore de montage sont appropriés pour construire le même type d'applications multimédia.

Les systèmes auteurs existants peuvent donc être classés en quatre catégories, suivant la métaphore qu'ils utilisent pour intégrer tous les médias dans une application (image, son, vidéo, texte) [Goffinet 02].

3.5.1 La métaphore de la carte

L'intégration et l'interaction des médias se font autour du concept de la *carte*. L'unité logique sur laquelle on travaille est la carte. Elle porte un nom, elle peut avoir des scripts associés, elle contient des champs, des boutons et des images. Elle est le réceptacle naturel de tous les objets, donc de tous les médias. Le fonctionnement de l'application est centré sur des liens entre objets situés sur une même carte ou sur des cartes différentes.

Les systèmes basés sur cette métaphore sont bien adaptés à des applications de type "livre électronique" où l'on tourne les pages à l'écran, et où l'on parcourt des informations selon une structure arborescente pas trop compliquée. Ils permettent aussi des choses plus complexes, puisqu'ils comportent tous un langage de scripting, et souvent admettent des commandes externes (mais au prix d'efforts proportionnels à leur complexité).

Par contre, lorsque l'interactivité devient fort complexe, ou lorsque les animations à réaliser sortent des simples effets prévus, ils ne peuvent plus satisfaire à la demande. De même, le fait pour la plupart de posséder un langage de scripting interprété peut poser des problèmes si la performance en temps d'exécution est un facteur clé. On réservera donc ces systèmes auteurs à

des applications simples ou à la réalisation d'interfaces avec des logiciels externes qui en sont mal pourvus. On trouvera dans cette catégorie *Hypercard*, (Mac), *MetaCard* (Unix) *Toolbook* (PC).

3.5.2 La métaphore de la ligne du temps

Les différents médias appartenant à l'application que l'on réalise sont affichés ou joués en fonction d'un scénario temporel qui est spécifié par son concepteur. Les objets apparaissent et disparaissent de l'écran selon un schéma temporel bien défini. On ne retrouve plus ici le concept de carte, mais celui de *scène*, d'*acteurs* et de *scénario*.

La table de montage se prête bien ici à construire des logiciels multimédias qui sont essentiellement des animations, telles que, par exemple, des shows de présentation interactive, des bornes de musée, etc. Mais, comme pour les systèmes de cartes, l'interaction ne doit pas être trop complexe (multiples branchements décidés en fonction de nombreux paramètres), sinon l'outil se révèle moins bien adapté à ce que l'on veut faire. Et ceci même s'il y a des possibilités assez riches de scripting et de commandes externes.. Exemples : *MacroMind Director* (Mac et PC), *Passport Producer* (Mac).

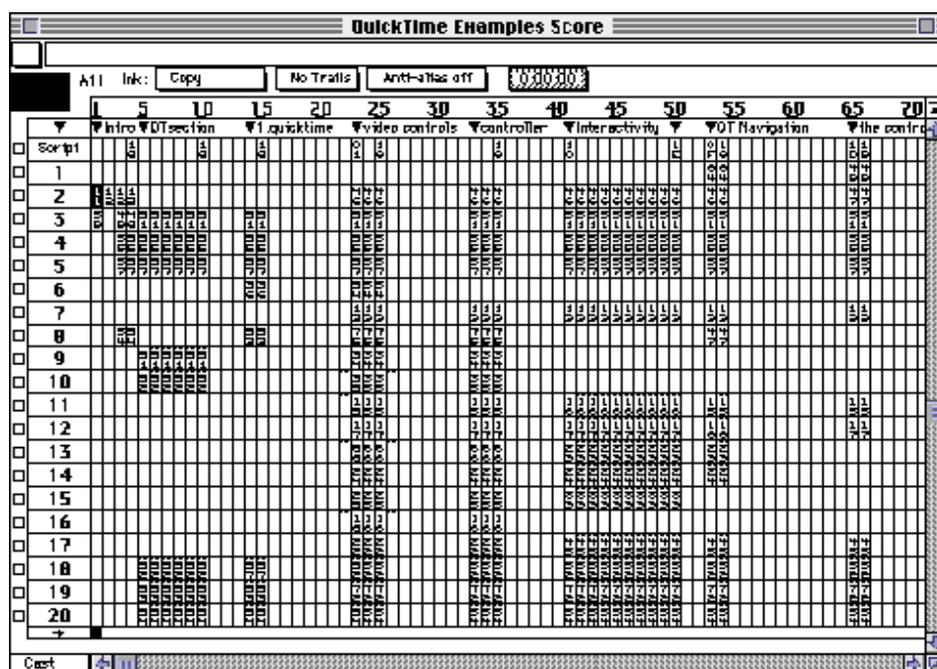


Figure 3.2. La table de montage de Director (une piste contient au plus un média)

3.5.3 Les diagrammes d'icônes

Dans la logique du montage par diagramme d'icônes, les écrans et les objets qui y sont liés s'enchaînent en fonction d'un organigramme défini par le concepteur de l'application. On part donc de la dynamique de l'interaction avec l'utilisateur pour construire, de façon visuelle, son application multimédia. La colonne vertébrale de l'application est un diagramme d'icônes qui décrit les événements qui s'enchaînent dynamiquement (par exemple, jouer un son ou entamer un dialogue avec l'utilisateur).

Les systèmes basés sur des diagrammes d'icônes se prêtent bien à la formation. Ils sont parfaitement adaptés à l'interactivité et aux traitements que cela entraîne, parce qu'ils offrent dès le départ l'arborescence qui correspond aux multiples branchements possibles. Ils offrent aussi un niveau de scripting approprié, bien que l'algorithmique qu'ils entraînent ne soit pas toujours à la portée du premier venu. Exemples de ce type de logiciels : *Authorware* (Mac et PC), *IconAuthor* (PC).

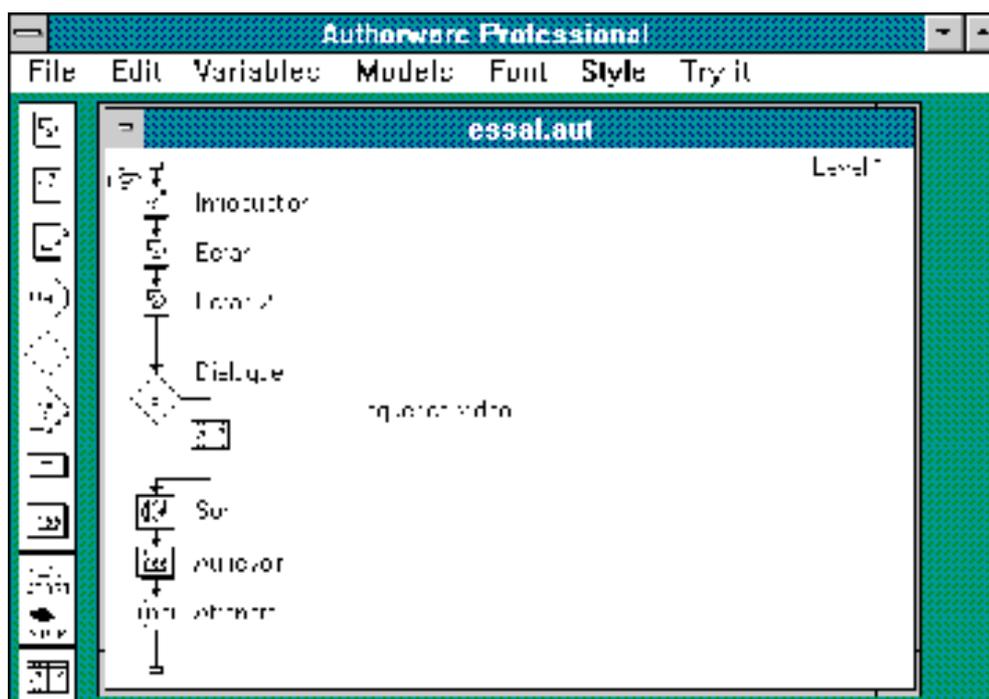


Figure 3.3. Construction d'une application avec AuthorWare

3.5.4 Les systèmes hypertextes améliorés

La logique employée ici pour construire l'application est basée sur l'utilisation de liens de type hypertexte. Les différents éléments de l'application multimédia sont décrits dans un fichier texte,

qui est ensuite parcouru manuellement pour marquer les différents objets, les relier entre eux et leur associer des actions. Au moment de l'exécution de l'application on parcourt les écrans successifs au travers de clicks souris sur des morceaux de texte, des images, etc.

Avec cette catégorie de systèmes auteurs, on reste fort limité dans la gestion de l'interactivité avec l'utilisateur. Ces systèmes s'adaptent bien à l'élaboration d'aides en ligne pour d'autres logiciels. Ils améliorent ainsi les systèmes d'aide habituels, limités à de l'hypertexte. Ils s'adaptent aussi particulièrement bien aux systèmes manipulant de grosses quantités de texte ou de données textuelles, du type encyclopédies. On trouvera dans cette catégorie de systèmes : *Microsoft MultiMedia Viewer* (PC).

3.5.5 Systèmes généralistes et systèmes spécialisés

Bien souvent, les systèmes auteurs les plus répandus sur le marché sont des *systèmes généralistes*, c'est-à-dire qu'ils contiennent des fonctionnalités standards qui satisfont aux besoins de nombreux utilisateurs venant d'horizons différents. Mais le prix à payer pour cette généralité est souvent que le système auteur n'est pas optimal pour une catégorie d'utilisateurs bien précise. La généralité implique une multitude de fonctionnalités dont l'utilisateur n'a pas vraiment besoin, et un noyau d'autres fonctions qui devrait être développé davantage pour correspondre à ses attentes. C'est dans cette optique de spécialisation des fonctionnalités que l'on retrouve les « systèmes auteurs spécialisés », par exemple un système auteur dédié à l'apprentissage du langage Prolog.

3.5.6 Portabilité

Une propriété des systèmes auteurs à ne pas négliger est leur capacité à pouvoir créer des applications multimédias tournant sur d'autres plateformes que celle sur laquelle ils ont été développés. On observe ainsi que le temps des systèmes auteurs dévolus à une plateforme unique sera bientôt révolu. Par exemple, beaucoup de systèmes auteurs de renom qui ont vu le jour dans le domaine de la micro-informatique sur Macintosh (*Director*, *Authorware*) permettent maintenant de générer une application tournant aussi bien sous le MacOS que sous MS-Windows. C'est un aspect qui peut être important, tant les coûts de passage d'une plateforme à l'autre ont été dans le passé un obstacle majeur à la diffusion de certains produits.

Par contre, ce qui est loin d'être acquis, c'est la compatibilité des systèmes auteurs entre eux. Si, par exemple, une animation créée avec *MacroMind Director* peut s'intégrer facilement dans *Authorware*, c'est plutôt l'exception dans ce domaine. Lorsque l'on travaille avec un système

auteur, on peut difficilement en changer en cours de route sous peine de devoir tout recommencer à zéro. Une application créée avec l'un de ces systèmes n'est bien souvent pas récupérable par un autre. Il y a là une réelle fermeture des systèmes auteurs entre eux.

3.6 Critères de sélection d'un système auteur

Les utilisateurs souhaitant concevoir des logiciels éducatifs rapidement et facilement, font généralement appel à un système auteur suite à une rapide démonstration ou lecture d'un article. Actuellement, ce sont des institutions (entreprises ou universités) spécialisés en informatique ou dans l'enseignement, qui adoptent le plus cette solution.

L'utilisateur doit par conséquent, après avoir défini son objectif, se poser un certain nombre de questions avant de s'investir dans un système auteur. Ces questions peuvent être d'ordre logistique, fonctionnel et/ou technique. Une partie de ces questions sont tirées des questionnaires publiés par le Ministère Français de l'Education Nationale et de la Formation Professionnelle sur Internet ([Http://www.men.lu/_didac/choisir.html](http://www.men.lu/_didac/choisir.html)) intitulés « Questions à poser lors du choix d'un logiciel ».

3.6.1 Questions d'ordre logistique

Les questions logistiques concernent tout ce qui a trait à l'approvisionnement, à l'entretien, au transport des équipements, qu'il s'agisse d'une simple barrette de mémoire vive, d'une carte son, de l'ordinateur ou des logiciels. Nous allons proposer quelques questions relatives à notre domaine d'intérêt, les systèmes auteurs :

- Le logiciel est compatible avec quel type de CPU et quel système d'exploitation ? (Mac, PC, CD-I, Workstation, ...)
- Le logiciel nécessite-t-il des périphériques particuliers pour tourner ? Si oui, sont-ils faciles à intégrer ?
- Pour ne pas restreindre l'utilisateur final à travailler sur un environnement de travail particulier, est-il possible de porter le résultat produit sur d'autres plates-formes ?
- Le fournisseur du logiciel donne-t-il la garantie de la durée de vie du logiciel, dans le sens où le produit peut être amélioré ?
- Quel est le prix du logiciel ? (certains outils coûtent très cher)

3.6.2 Questions d'ordre fonctionnel

Le mode de fonctionnement d'un système auteur est primordial, surtout si l'utilisateur est néophyte en informatique. Les quelques questions ci-dessous semblent alors être incontournables :

- Le logiciel est-il basé sur les connaissances de l'utilisateur en matière de didactique et de pédagogie ?
- Est-ce que le logiciel est facile à utiliser ? Y a-t-il des assistances à l'utilisateur ?
- Existe-il des interactions entre le logiciel et l'utilisateur ?
- Y a-t-il plusieurs approches possibles suivant la compétence de l'utilisateur ? Le logiciel accroît-il la motivation de l'apprenant ?
- Le contenu du logiciel est-il modifiable, peut-on y ajouter des éléments nouveaux, l'enrichir et l'adapter aux besoins et/ou capacités de l'utilisateur ?

3.6.3 Questions d'ordre technique

Les questions techniques sont aussi importantes que celles relatives à la logistique et au fonctionnement. Voici quelques exemples :

- Le logiciel est-il un système ouvert afin de permettre à l'utilisateur d'intégrer des programmes externes au système auteur ?
- Le logiciel met-il à la disposition de l'utilisateur les outils nécessaires à des traitements spécifiques comme le graphisme ? les systèmes experts ?
- L'application générée par le système auteur est-elle capable de traiter des requêtes spécifiques ou des données nouvelles que l'utilisateur final veut intégrer au fur et à mesure dans le programme ? Par exemple, si le logiciel n'accepte que les images en format bitmap (bmp), et que l'utilisateur dispose de certaines images gif.

En plus de ces questions concernant les caractéristiques du système auteur, l'utilisateur devrait fixer clairement ses objectifs :

- De quel type d'application éducative va-t-il s'agir ? (apprentissage, tutorat, encyclopédie, animation, etc.) ?
- De quel niveau d'interactivité a-t-on besoin ? (faible, complexe) ?
- Quel mode de navigation veut-on offrir ? (linéaire, hiérarchique, hypertexte) ?
- De quel budget dispose-t-on ? (certains outils coûtent très cher) ?

Une fois que l'on a les réponses à ces questions essentielles, on peut à ce moment-là s'orienter vers un type d'outil bien précis. Pour donner une petite idée des différents systèmes auteurs existants, on peut classer dans un tableau les quelques principaux systèmes auteurs sur le marché, en fonction du niveau de puissance offert et de la métaphore qu'ils emploient pour intégrer les différents média entre eux.

Système auteur	Niveaux de travail	Table de Montage	Plateforme
Hypercard (Claris)	Montage, scripting, commandes externes	cartes	Mac
ToolBook (Asymetrix)	Montage, scripting, commandes externes	cartes	PC
Director (Macromédia)	Montage, scripting, commandes externes	ligne du temps	Mac/PC
Producer Pro (Passport)	Montage, commandes externes	ligne du temps	Mac
AuthorWare (Macromédia)	Montage, scripting, commandes externes	diagramme d'icônes	Mac/PC
IconAuthor (AimTech)	Montage, scripting, commandes externes	diagramme d'icônes	PC
Apple MédiaTool (Apple)	Montage	diagramme d'icônes	Mac
Multimédia Viewer (Microsoft)	Montage, commandes externes	hypertexte	PC

Table 3.1. Quelques exemples de systèmes auteurs avec leur niveau de puissance

3.7 Réalisation d'un logiciel éducatif à partir d'un système auteur

Il n'existe aucune méthode à imposer aux utilisateurs pour réaliser un logiciel éducatif à partir d'un système auteur. En revanche, les conseils donnés dans la majorité des systèmes auteurs pour écrire son logiciel [Wu Yao Kuang 00], quelles que soient les catégories de systèmes à produire en fonction du besoin, sont globalement les suivants :

1. Diviser l'application en différentes étapes et chaque étape en plusieurs fonctions.
2. Sélectionner les procédures que l'on peut généraliser.
3. Constituer les différents composants de l'application finale : les données textuelles à saisir, le scan des photos, la construction des schémas, etc.

4. Se familiariser avec le système auteur : son langage, ses icônes, etc. Les subtilités et complexités comme les syntaxes seront à travailler plus tard dans les démarches et doivent donc être vues superficiellement pour ne pas bloquer l'approche avec le système.
5. S'appuyer sur les exemples donnés généralement avec le système, afin de s'inspirer de certains points se rapportant à son aventure.
6. Partir de ces exemples, enlever les points inutiles et ajouter petit à petit des fonctions spécifiques.
7. Développer les parties complexes et intégrer les bibliothèques et routines externes au système auteur.
8. Et enfin, Commenter le maximum de codes et des modifications éventuelles.

Aussi, il ne faut pas oublier qu'avant de pouvoir réaliser un produit à partir d'un système auteur, l'utilisateur doit se familiariser avec la machine sur laquelle il travaille, en comprendre les rouages et en interpréter les messages [De La Passardière 93]. Notre étude est orientée vers l'enseignement, c'est-à-dire tout ce qui est logiciel destiné à l'apprentissage. Par conséquent, la réalisation d'un logiciel éducatif à partir d'un système auteur nécessite des efforts pédagogiques et ergonomiques de la part du concepteur du logiciel final. La figure ci-dessous [Wu Yao Kuang 00] nous donne une vision globale des étapes de réalisation de logiciels éducatifs à partir d'un système auteur.

Comme le montre la (**Figure 3.4**) et conformément à la démarche génie logiciel connue de tous, l'auteur du logiciel éducatif commence par la définition de l'objectif de son projet ainsi que la détermination de la population cible. Il est important de noter que l'auteur du système, fait partie lui-même, inconsciemment ou non, de la population cible car ses goûts et souhaits vont intervenir sur sa méthodologie de travail, sur la structure du logiciel, donc sur le résultat final.

Ensuite, l'auteur fait une étude fonctionnelle qui permet de cerner les champs d'action ou d'utilisation du logiciel qu'il souhaite mettre à disposition dans son système, donc de définir l'ensemble des problématiques et en décrire les conséquences. Relativement aux spécifications fonctionnelles, l'auteur donne les spécifications techniques permettant la concrétisation de ces fonctions. Le développement et la validation ne font que concrétiser et respecter au mieux ces différentes étapes de la conception.

Enfin, il faut noter que les données traitées par les systèmes auteurs sont souvent de différents types et peuvent alors être intégrées de différentes manières. Par exemple, si un utilisateur veut intégrer dans son logiciel des modules de questions-réponses, se basant sur un système expert et

comprenant des analyseurs syntaxiques, tout en explorant les capacités du multimédia ; il faut mettre à sa disposition les outils permettant de réaliser ces tâches, dont les plus connus sont :

Le traitement textuel pour intégrer des données sous forme texte que nous appellerons passif ou actif : actif dans le sens où l'utilisateur peut agir sur ce texte ou que le texte s'adapte au fur et à mesure à l'utilisateur ; passif lorsque le texte est présent juste à titre d'information.

Le graphisme qui permet de représenter la logique de conception dans le sens où un objet graphique regroupant plusieurs données simule généralement une action. De plus, le graphisme est une partie intégrante du multimédia et influence directement la relation homme-machine. Le caractère synthétique des schémas aide beaucoup les utilisateurs qu'ils soient informaticiens ou non, car la représentation se réalise dans un espace à deux dimensions, donc on perçoit une logique non linéaire, plus facilement compréhensible. De ce fait, les compétences ergonomiques et graphiques du concepteur du système auteur sont essentielles.

Des outils complémentaires d'intégration de données, souvent des logiciels à part entière, sont intégrés dans le système auteur: traitement du son et éventuellement de la parole, compression des images, animation, etc. Il s'agit souvent d'une fonction demandant l'exécution (lancement) d'un autre logiciel tel que QuickTime pour les images animées.

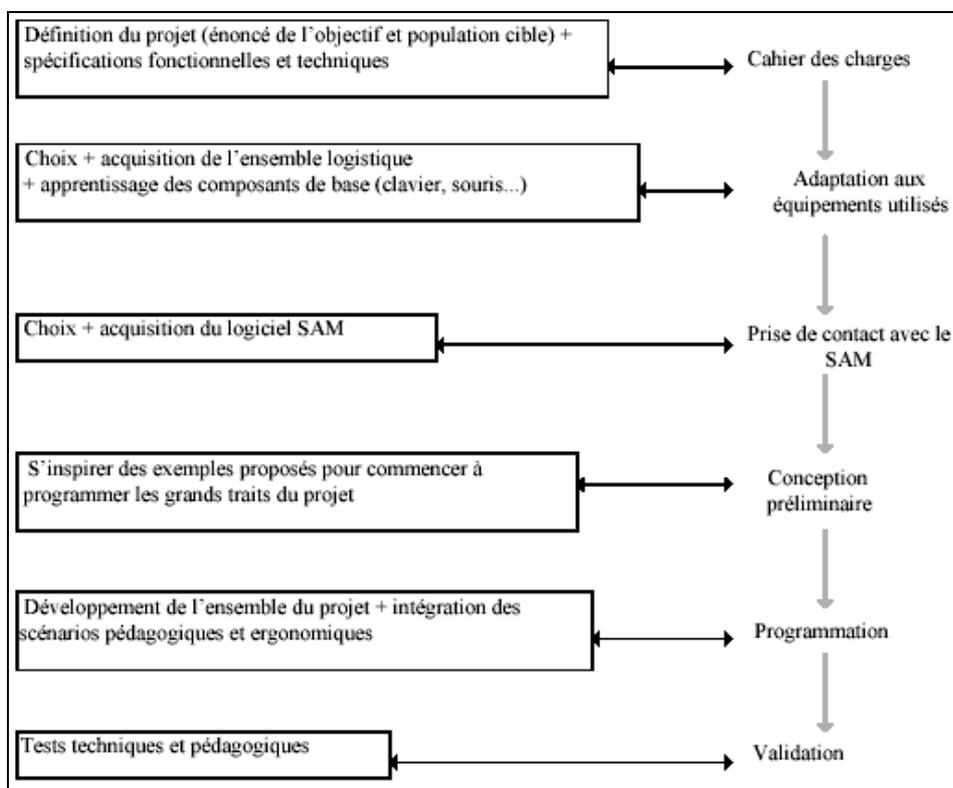


Figure 3.4. Processus de réalisation d'un logiciel éducatif à partir d'un système auteur

3.8 Systèmes auteurs de l'EAO classique

3.8.1 Préliminaires

Dans le contexte de l'EAO classique, les années fin 70 - début 80 ont vu apparaître les premiers systèmes auteurs pour la production de didacticiels. Ces systèmes ont déchargé entièrement les enseignants de la tâche de programmation que ce soit avec les langages de programmation classiques ou les langages auteurs apparus par la suite [Madaule & al 87]. Grâce à des (éditeurs) conviviaux, les enseignants (auteurs) ne font alors qu'agencer les données pédagogiques (i.e. connaissances du domaine) en un didacticiel, suivant un modèle pédagogique généralement prédéfini. Ces données pédagogiques sont ensuite interprétées par un autre programme (l'interpréteur) pour assurer la diffusion du didacticiel à l'élève. Dans les modules destinés à l'enseignant (appelé mode auteur), on trouve généralement les procédures suivantes :

- création de leçons et d'exercices que l'on intègre aux cours déjà élaborés;
- modification de leçons en vue de les adapter aux difficultés propres à un groupe d'élèves ou à des buts pédagogiques différents;
- suppression provisoire ou définitive de certaines parties d'un cours ou d'une série d'exercices;
- présentation du contenu du système obligatoirement sur l'écran de l'ordinateur et parfois sur l'imprimante;
- gestion d'une population d'élèves, c'est-à-dire enregistrement ou élimination de noms et suivi des leçons utilisées par chacun, des réponses erronées ainsi que des résultats obtenus.

Avec cet ensemble de fonctionnalités, l'auteur de didacticiels peut non seulement mettre en parfaite correspondance le contenu du système et son enseignement traditionnel mais encore contrôler les progrès de chacun de ses étudiants.

Pour les apprenants, à travers le mode apprenant, le système auteur offre généralement les possibilités suivantes :

- afficher à l'écran la liste des leçons et des exercices disponibles afin que l'étudiant puisse choisir ce qu'il désire;
- fournir toutes les indications techniques nécessaires à l'emploi du système. L'élève ne doit jamais être bloqué à cause d'une maladresse ou d'une manœuvre non prévue par le logiciel;
- expliquer clairement le contenu de la leçon et la nature des réponses attendues;
- interroger l'élève et l'aider en cas de difficultés;
- permettre de ne pas enregistrer les résultats d'un exercice;

- donner à l'étudiant une vue précise du travail qu'il a effectué durant une période déterminée.

Dès l'origine de l'enseignement assisté par ordinateur, les concepteurs de didacticiels ont développé des systèmes auteurs de sorte qu'on en trouve pour tous les types d'ordinateurs. Sur micro-ordinateur, les systèmes auteurs se sont multipliés de sorte qu'il est presque impossible d'en donner une liste exhaustive. On trouvera ci-dessous les principaux :

3.8.2 Principaux systèmes auteurs sur gros systèmes

- PLATO, le plus ancien, dont la première mise au point a été réalisée en 1967, par la firme CONTROL DATA. Ce système, extrêmement coûteux, comporte plusieurs milliers d'heures d'enseignement et concerne quelque soixante-dix disciplines. PLATO propose des leçons pour plus de vingt-cinq langues différentes parmi lesquelles figurent l'anglais, l'allemand, l'espagnol, le français, le latin, le russe, le grec, le chinois, l'hébreu ou encore le swahili. Au point de vue technique, on retiendra que PLATO est doté d'écrans tactiles. Ce type de matériel qui réagit au toucher est utile pour développer des séquences d'enseignement destinées à des enfants qui commencent à lire ou encore des modules d'apprentissage des langues à l'intention des débutants. Plusieurs milliers de terminaux connectés à PLATO ont fonctionné aux Etats-Unis et au Canada [Viré 86].
- IMG (Instruction Module Generator) développé et diffusé par IBM France.
- CAN 8 mis au point par la société française Bull, sur l'ordinateur MINI 6.
- TICCIT, créé par Hazeltine, et concurrent de PLATO aux Etats-Unis.

3.8.3 Principaux systèmes auteurs sur micro-ordinateurs

Il est quasiment impossible de passer en revue tous les systèmes auteurs qui fonctionnent sur les micro-ordinateurs. Les constructeurs de ce type de matériel, conscients de l'importance du marché éducatif, ont presque tous tenté d'offrir avec leurs machines, des logiciels pédagogiques plus ou moins acceptables. On se contentera de retenir :

- PRIVATE TUTOR utilisé sur les ordinateurs personnels IBM;
- SEF (Self Education Facility), commercialisé dès 1985, par IBM Allemagne. Selon la firme IBM, les possibilités de ce système sont plus étendues que celles de PRIVATE TUTOR. Malheureusement, SEF demande un temps d'apprentissage plus long que la plupart des autres systèmes.
- ARLEQUIN élaboré en France par deux professeurs de mathématiques. Selon ses auteurs, ARLEQUIN s'efforçait de rencontrer les besoins pédagogiques des enseignants, notamment

en ce qui concerne l'analyse des réponses et le suivi du travail des élèves. En fait, l'utilisateur d'ARLEQUIN percevait très vite ses inconvénients qui résultaient de sa rigidité, de sa lenteur et de l'apprentissage assez long qu'il imposait [Torcq & Flaschner 83].

- PIGE (Programme Informatique de Génération d'Exercices) est une réalisation de l'Université des Sciences et des Techniques de Lille (France). Ce système était constitué de trois parties, la première, destinée à l'enseignant, permettait de construire le didacticiel. La deuxième partie, utilisée par l'étudiant, fournissait toutes les indications dont il avait besoin pour travailler (choix de l'exercice, consignes techniques), le soumettait à des questions, enregistrait ses réponses et, en cas de besoin, lui apportait de l'aide. Dans le domaine des langues, PIGE a notamment servi à mettre au point de nombreux exercices à l'intention des grands débutants en anglais [Thibaut & Verrier 83] .
- SAM (Système d'Apprentissage par Microordinateur) a été développé à l'Université de Paris III, par le groupe de recherches sur l'apprentissage assisté par ordinateur de l'allemand. Ce progiciel, conçu et mis au point sous la direction d'un professeur d'allemand, est probablement supérieur aux précédents en ce qui concerne l'analyse des réponses [Janitza 85].

3.9 Systèmes auteurs de tuteurs intelligents classiques

3.9.1 Préliminaires

Les systèmes auteurs de tuteurs intelligents classiques (STI) ont suscité beaucoup l'intérêt des chercheurs ces deux dernières décennies. Tom Murray en a cité plus de deux douzaines dans sa synthèse la plus récente et la plus complète [Murray 99]. Parmi ces systèmes nous pouvons citer : ECAL [Elsom & Omalley 90], EDDI [Marcenac 92], Cream-tools [Nkambo & al 03], Eon [Murray 98], Iris [Arruarte & al 97], Metalinks [Murray 98], Training Express [Clancey & Joerger 88], etc. L'auteur en a dressé une classification en sept catégories relativement au type de STI qu'ils produisent :

1. Enchaînement et planification de curriculum ;
2. Stratégies tutorielles ;
3. Simulation et entraînement ;
4. Système expert en domaine ;
5. Types de connaissances Multiples ;
6. Système à but spécial ;
7. Système intelligent hypermédia/adaptif.

Etant donné que les STI se trouvent souvent structurés en quatre composants (expert du domaine, pédagogue, modèle-apprenant, et interface apprenant), les systèmes auteurs doivent donc théoriquement fournir tous les outils nécessaires permettant de les construire. Néanmoins, il faut le reconnaître, très peu de systèmes exigent des auteurs de tout construire comme c'est le cas par exemple du système Eon [Murray 98]. Les autres systèmes se limitent en général à des outils permettant de construire un, deux ou à la limite trois composants parmi les quatre. Le reste des composants est généralement prédéfini dans un canevas de STI et l'auteur du STI n'est sollicité que pour introduire des paramètres nécessaires à leur fonctionnement.

TalHits, le système présenté dans cette thèse et offrant des fonctionnalités de collaboration aux auteurs, génère des STI qui se situent dans la première et septième catégorie de cette classification : enchaînement et planification de curriculum, système hypermédia intelligent/adaptatif.

Les systèmes auteurs de cette catégorie structurent généralement la matière à enseigner sous forme d'un réseau d'Unités d'Apprentissage (UA) et où chaque UA possède certains objectifs pédagogiques. Les UA sont liés entre elles par des liens de type *prérequis*, *partie de*, *défini par*, *expliqué par*, etc. Quoique ces systèmes n'utilisent pas de représentation explicite des connaissances du domaine, ils investissent cependant l'intelligence au niveau de l'enchaînement des UA, la manipulation des liens hypertextes et l'adaptation du cursus par l'utilisation d'un modèle de l'apprenant. Les UA à présenter à l'apprenant sont alors déterminées dynamiquement en se basant sur les performances de ce dernier, les objectifs pédagogiques de la leçon et les relations qui existent entre les différentes UA..

Rappelons que l'objectif de notre travail est de réaliser un système auteur pour des STI Hypermédias. Pour cela, il nous paraît essentiel d'analyser les architectures et les fonctionnalités de quelques systèmes en nous limitant aux points qui paraissent pertinents pour notre recherche, à savoir :

1. L'architecture générale de ces systèmes;
2. La structure et le déroulement des tuteurs produits;
3. Les outils mis à la disposition de l'auteur;
4. Les moyens utilisés pour la réalisation de ces systèmes.

3.9.2 Exemples de systèmes auteurs

3.9.2.1 Le système ECAL : Un environnement basé sur un réseau de concepts

Le système que nous présentons dans cette section est un système auteur développé à Open University Cite et baptisé ECAL (pour **Extended Computer Assisted Learning**) [Elsom & Omalley 90]. Ses concepteurs, ayant repris les mêmes principes de l'EAO orienté frames et enrichis de quelques techniques d'IA, le considèrent comme un "pont" entre l'EAO traditionnel et l'EIAO. De ce fait, les tuteurs produits par ECAL, se placent à un niveau intermédiaire entre les didacticiels classiques et les tuteurs intelligents. ECAL est organisé autour de trois composants :

1. « Outils pour l'auteur » : permettant à celui-ci de générer la matière à enseigner. Cette matière est construite à l'aide d'un ensemble de frames (ou unités d'apprentissage) purement textuels qui sont de deux natures : « frames de présentation » et « frames de diagnostic ». Chaque frame traite d'un certain nombre de concepts appelés ILOs (Intended Learning Outcomes).
2. « Système de présentation » : qui contrôle l'interaction avec l'apprenant en choisissant au mieux la séquence de frames à lui présenter. Les décisions sont prises en se basant sur deux modèles: le modèle de dialogue et le modèle de l'élève. Aussi, pour chaque interaction, le "système de présentation" développe un historique du dialogue qui peut être utilisé par l'auteur pour faire la mise au point du tutoriel.
3. « Outils de mise au point » : permettant à l'auteur de tester le système en examinant son comportement face à des situations particulières. Ceci lui permettra éventuellement de porter des modifications au contenu et à la structure du tutoriel.

La structure d'un tuteur créé par ECAL se compose de quatre modules : le « modèle de connaissances », le « modèle de l'élève », le « modèle de dialogue » et le « modèle de présentation ». A part le modèle de dialogue qui est propre à ECAL, nous retrouvons bien les trois modules rencontrés dans la plupart des tuteurs intelligents. Ils correspondent en effet, respectivement, aux modules : module expert, modèle de l'apprenant et module pédagogique.

a) Le modèle de connaissances

ECAL utilise une technique simple pour représenter les connaissances du domaine. Elle consiste à subdiviser la matière à enseigner en un certain nombre de frames purement textuels et

d'associer à chaque frame une séquence de mots clés désignant les concepts enseignés par le frame. Ces mots clés, appelés ILOs (**I**ntended **L**earning **O**utcomes), constituent les objectifs pédagogiques autour desquels le système est structuré. Plus particulièrement, c'est sur ces mots clés que le "système de présentation" s'appuie pour décider des différents enchaînements de frames à présenter à l'apprenant.

ECAL associe deux caractéristiques à chaque ILO : "l'importance" qui indique combien de fois l'ILO est utilisé dans le cours et la "généralité" qui est relative à la largeur avec laquelle cette utilisation se produit dans le cours. Deux ILOs peuvent alors être d'*importance* égale mais l'un d'eux est considéré plus *général* s'il est plus largement distribué à travers le cours.

b) Le modèle de l'élève

Pour modéliser l'élève, ECAL utilise la technique à recouvrement (overlay). Nous rappelons que cette technique considère la connaissance de l'élève comme un sous-ensemble des connaissances représentées dans le module expert du domaine.

Pour ECAL, le modèle de l'élève est constitué tout simplement d'une liste d'ILOs avec une valeur numérique associée à chacun d'eux. Cette valeur comprise entre 0 et 1 et appelée *facteur de confiance*, est utilisée par le système pour mesurer la compréhension de l'élève concernant le concept associé à cet ILO. La valeur 0 indique que l'élève n'a pas compris le concept. La valeur 1 indique qu'il l'a bien assimilé. Ces valeurs sont mises à jour au fur et à mesure que les frames sont présentés à l'élève. Quand un frame est présenté à l'élève, tous les ILOs associés à ce frame sont évidemment mis à jour. L'évaluation de l'élève est effectuée par les frames de diagnostic. Chaque *frame de diagnostic* est conçu pour vérifier l'assimilation d'un ou de plusieurs concepts.

c) Le modèle de dialogue

Afin de planifier les actions à entreprendre durant une interaction avec l'apprenant, le système de présentation utilise un « modèle de dialogue ». Cette modélisation utilise trois composantes:: « l'historique du dialogue », « l'agenda courant » et « le reste des buts à satisfaire ». Chacune de ces composantes est constituée d'un ensemble de « buts de dialogue ». ECAL représente ces buts par les ILOs eux-mêmes. *L'historique du dialogue* correspond alors à l'ensemble des buts (liste d'ILOs) qui ont été déjà satisfaits. *Le reste des buts à satisfaire* correspond quant à lui, à l'ensemble de buts qui ne sont pas encore planifiés pour être satisfaits. Quant à *l'agenda courant*, il correspond aux buts que le système est entrain d'essayer de satisfaire.

Ces trois *modèles* sont les trois supports de connaissances qui permettent au « modèle de présentation » (décrit ci dessous) de décider des choix pédagogiques à effectuer pour assurer un enseignement optimal et individualisé.

d) Le modèle de présentation

Ce modèle est aussi appelé « algorithme d'enseignement ». Il consiste en un ensemble de règles de décision regroupées en trois catégories, portant respectivement :

- sur le choix à effectuer entre une présentation de connaissances ou un diagnostic ;
- sur la sélection d'un frame approprié ;
- et enfin, sur le choix d'un nouveau ILO (i.e. le prochain but à satisfaire).

Les différents choix sont effectués en s'appuyant sur des heuristiques portant sur l'ordre de priorité accordé aux frames. Le choix d'un frame est effectué sur la base des informations contenues dans « l'agenda courant », « l'historique du dialogue », le « reste des buts à satisfaire » et le « modèle de l'élève ». Le système tente d'abord de choisir un frame qui maintient l'état courant de l'interaction. Si cela n'est pas possible, il récupère un nouveau état de dialogue dans « l'agenda courant » et sélectionne un frame approprié.

Etant donné un ILO à enseigner, le système collecte tous les frames mentionnant cet ILO et qui ne sont pas encore présentés à l'apprenant. Un ordre de priorité est ensuite attribué à chacun de ces frames. La priorité maximale est attribuée aux frames dans lesquels tous les ILOs (autres que l'ILO principal) ne sont pas connus par l'apprenant. Les frames seront ordonnés de telle sorte que celui qui contient le minimum de nouveaux ILOs est celui qui aura la plus haute priorité.

Une fois les frames concernés par l'ILO en question sont ordonnés selon ces critères, le système en sélectionne celui qui a la priorité maximale, le présente à l'apprenant, puis il obtient une réponse de celui-ci. Cette réponse lui permettra de mettre à jour le modèle de l'élève et le modèle de dialogue en conséquence.

d) Outils pour l'auteur

Plusieurs outils sont mis à la disposition de l'auteur pour réaliser des tutoriels avec ECAL. Nous trouvons :

1. un « éditeur de frames » permettant à l'auteur de créer des scènes textuelles et leur associer des ILOs;

2. un « éditeur de réseaux » permettant à l'auteur de consulter la structure du réseau des ILOs. Ce réseau est normalement généré automatiquement, mais il est permis à l'auteur de faire certaines modifications en ajoutant ou en supprimant des ILOs;
3. un « explorateur de réseaux » permettant à l'auteur d'explorer le réseau des ILOs afin de consulter tous les détails concernant les arcs qui relient deux noeuds entre eux. Nous rappelons qu'un arc correspond à tous les frames partagés par les ILOs contenus dans les deux noeuds.
4. des « outils de test » et de « mise au point » de tutoriels en vue de les rendre satisfaisants aux objectifs pédagogiques visés. Ces outils sont capables de simuler toute interaction avec l'apprenant et de montrer toutes les décisions prises par le système dans différentes situations.

3.9.3 EDDI : Un environnement basé sur une expertise du domaine

Contrairement au système présenté ci-dessus, l'environnement EDDI (Environnement de Développement de Didacticiels Interactifs) s'appuie sur une modélisation de l'expertise du domaine sous forme de réseaux sémantiques. Le domaine à enseigner est structuré en un ensemble d'objets et de relations typées (agrégation, déduction, généralisation) formant un réseau sémantique. Cette structure permet d'exprimer implicitement les savoir-faire, par le parcours du réseau lors de la recherche d'une solution d'un problème. Le raisonnement déductif est assuré par un ensemble de stratégies constituant le moteur d'inférence du système. L'approche menée dans EDDI vise à développer un système auteur en partant d'un système expert du domaine existant, auquel on ajoute un niveau générique de stratégies tutorielles et un modèle de l'apprenant. Pour rendre ces stratégies tutorielles indépendantes du domaine à enseigner, elles sont connectées à la structure des objets et des relations typées et non pas à l'expertise elle-même. Voici deux exemples de ces stratégies:

- **Fournir de l'aide à l'apprenant:**

Si *l'environnement de l'objet courant X contient des relations de type **agrégation** ;*

Alors *fournir le message: "**X est composé de . . .**".*

- **Poser une question à l'apprenant :**

Si *l'environnement de l'objet courant X contient une relation de type **généralisation** ;*

Alors *poser la question: "**pouvez-vous me dire à quel catégorie appartient X**".*

Le système EDDI n'est pas adapté à tout domaine ; il est surtout adapté aux domaines dans lesquels les connaissances sont bien structurées tel que le diagnostic médical auquel le prototype a été appliqué. Le modèle de l'apprenant utilisé est de type overlay. Le prototype qui a nécessité quatre ans de développement a été réalisé avec le langage orienté objet LOOPS.

3.10 Les systèmes auteurs sur le web

3.10.1 Préliminaires

Les soudaines fiançailles des systèmes auteurs et du web constituent clairement la tendance de la dernière décennie la plus spectaculaire. Un simple coup d'œil aux sites officiels des logiciels concernés suffit à convaincre de l'étendue publicitaire menée autour de ce qui est présenté comme une évolution majeure [Uyterbrouck 98] :

« Authorware was the first authoring tool enhanced specifically for the web » (site de Macromedia [Url 0]).

« ToolBook II offers one of the only authoring solutions that lets you quickly convert your applications directly into html and Java, the industry standards for Internet content » (site d'Asymetrix [Url 0]).

« Announcing IconAuthor Net Edition! Take your training to the net with the ultimate authoring software for multimedia and Internet-based training » (site d'AIMTECH [Url 0]).

Il va de soi que les principaux vendeurs de systèmes auteurs ne pouvaient rester sourds à l'appel du Web. Via Internet et le Web, il devient possible de toucher un public international, de diffuser des cours en ligne, de faire abstraction des problèmes de plates-formes, d'effectuer des mises à jour rapides de l'information, etc.

Si les points forts du Web sont indéniables, et si les vendeurs de systèmes auteurs affirment leur volonté de tirer profit de cette technologie, le concept de "compatibilité avec Internet" sur laquelle se base la publicité recouvre en réalité une multitude de fonctionnalités différentes. En voici une série pour exemple, sans souci d'être exhaustif :

- La possibilité, à partir d'un bouton ou d'un lien hypertexte, de lancer un navigateur Web (Netscape Navigator, Internet Explorer) sur une page web donnée.
- La possibilité, pour un navigateur, d'interpréter et d'afficher du HTML.
- L'envoi de courrier électronique.
- La gestion des échanges de fichiers via le protocole FTP.

- L'affichage d'une application produite par un système auteur dans un navigateur, grâce à une extension spécifique à celui-ci.
- La gestion des cours et des apprenants sur le réseau Internet.
- L'optimisation du transfert des données par Internet pour diminuer les temps de chargement.
- La conversion d'une application produite par un système auteur en un mélange du html et de Java.

Chaque système auteur fournit une ou plusieurs de ces fonctionnalités. Les fonctionnalités névralgiques sont évidemment celles qui permettent de visualiser le produit d'un travail de développement dans un navigateur web. Deux approches concurrentes sont exploitées :

- La première est celle de l'incrustation. Le principe est ici d'étendre les possibilités du navigateur à l'aide d'une extension (appelée "plugin") qui permet de visualiser un fichier de format x, y ou z au sein d'une page web. Les fichiers Authorware peuvent ainsi être visualisés à l'aide de l'extension Shockwave [Url 1], les fichiers Toolbook à l'aide de l'extension Neuron [Url 2], les fichiers Supercard à l'aide de l'extension Roadster [Url 3], etc. Les avantages de cette approche sont que la très grande majorité des fonctionnalités du fichier original est conservée (l'application incrustée dans la page web fonctionne comme elle le ferait sur disque dur ou sur Cd-Rom), et que la plupart des systèmes auteurs offrent désormais cette possibilité. Les inconvénients ne sont cependant pas négligeables : obligation pour l'utilisateur de télécharger et d'installer préalablement une extension qui peut faire plusieurs mégabits, perte de l'indépendance par rapport à la plate-forme, perte de la standardisation offerte par le web, lenteur du téléchargement, absence d'intégration réelle entre l'environnement du système auteur et celui du web (même si le fichier auteur semble faire partie intégrante de la page web, tout se passe en réalité comme si on avait ouvert deux applications distinctes).
- La deuxième approche est celle qui consiste à convertir l'application développée par le système auteur en un mélange de html (pour la présentation) et de java ou un autre langage de programmation Internet (pour l'interactivité). Les avantages théoriques sont ici clairs : la consultation des pages produites peut se faire à partir de n'importe quelle plate-forme, sans installation préalable d'extensions, et le chargement est plus rapide. Le problème est que le procédé ne tient guère ses promesses. En effet, les systèmes auteurs ont, en matière de fonctionnalités offertes, une dizaine d'années d'avance sur les outils web. La fameuse

"conversion" ne se fait que dans le sens d'un appauvrissement sensible, voire radical, de l'application originale. Ainsi par exemple, Toolbook permet bien d'exporter vers le web ses fichiers, mais n'exporte pas la programmation sous-jacente. Le développeur s'avère en fait obligé de se cantonner à une bibliothèque spécifique d'objets du système auteur.

3.10.2 Aspects pédagogiques

À l'exception peut-être d'Authorware, les systèmes auteurs comme HyperCard, Toolbook, Director, etc., étaient à l'origine (et restent pour certains) des outils de développement multimédia au sens large. L'inévitable corollaire est que ces logiciels, par définition, ont longtemps péché par une totale ignorance des besoins propres au monde de l'enseignement : facilité de mise en oeuvre pour des enseignants non-spécialistes de l'informatique, organisation d'une leçon, suivi pédagogique, etc. En substance, un formateur et un créateur de cédérom grand public disposaient exactement des mêmes outils. Force est cependant de constater qu'aujourd'hui - et l'on ne peut que s'en réjouir - les systèmes auteurs (ou une partie d'entre eux en tous cas) ont largement comblé ces lacunes. L'évolution de Toolbook, système auteur multimédia largement " inspiré " d'HyperCard à l'origine, est à cet égard plus qu'éclairante. La version 3.0 de Toolbook était une version généraliste. À partir de la version 4.0, la gamme se décline en deux variantes : une version généraliste, et une version 4.0 CBT (Computer-Based Training). La version 6, comporte toujours deux variantes, mais cette fois la version CBT (renommée Toolbook Instructor) est devenue le produit phare.

Toolbook Instructor demande beaucoup moins d'efforts de la part du concepteur de leçons, puisqu'il met à la disposition de celui-ci une riche bibliothèque d'exercices préprogrammés (questions à choix multiples, appariements, puzzles, etc.), assortis chacun d'une impressionnante série de paramètres (temps limite, nombre de réponses autorisées, etc.). En outre, il est possible d'assurer un suivi pédagogique des activités de l'apprenant (pages consultées, résultats aux exercices, etc.) sans programmation aucune.

Les produits dominants du marché ont donc désormais intégré les fonctionnalités de base nécessaires à l'élaboration de didacticiels. Les résultats d'une enquête réalisée aux États-Unis auprès d'un millier de sociétés pratiquant le Computer-Based Training [Url 4] sont à cet égard révélateurs : parmi les fonctionnalités considérées comme importantes pour le choix d'un système auteur, les fonctionnalités pédagogiques sont en baisse par rapport aux années précédentes. Nous ne pensons pas qu'il faille interpréter ceci comme une baisse d'intérêt pour ce type de

fonctionnalités, mais plutôt comme un indice du fait que celles-ci sont désormais considérées comme nécessaires, à la manière des fonctionnalités multimédias.

La parution de Designer's Edge [Url 5], un outil de pré-authoring, même s'il ne s'agit pas à proprement parler d'un système auteur, a sans nul doute marqué une date dans l'histoire du domaine. Designer's Edge est en effet le premier logiciel de planification de développement de didacticiels, qui aide l'équipe de conception dans sa tâche, depuis l'analyse initiale des besoins jusqu'à l'évaluation du produit final, en passant par l'écriture du script. Designer's Edge marque un progrès important par son souci des préoccupations pédagogiques au sens large, et par son approche systématique et structurée : c'est sur la base de besoins que l'on définit des objectifs, avant de leur associer une ou plusieurs stratégies, qui elles-mêmes engendreront un ou plusieurs "écrans" donnés.

Dans cette optique, la phase de développement proprement dite, celle qui utilise un système auteur, reprend la place qui lui est due : non plus celle de la phase essentielle de production, mais bien celle d'une simple maille dans une chaîne complexe, qui demande une importante réflexion aussi bien en amont qu'en aval. Designer's Edge est par ailleurs immédiatement entré au "hit-parade" des produits les plus utilisés, où il occupait déjà la sixième place en 1997.

L'intérêt d'un tel outil ne s'arrête pas là, puisque Designer's Edge version 2 dispose à présent d'un produit compagnon, baptisé Synergy, qui permet, à partir du script rédigé avec Designer's Edge, de générer une application au format d'un système auteur particulier.. Si le produit remplit ses promesses, il présente le double avantage de combler un grand vide dans le processus de développement d'une part, et d'ouvrir des pistes vers une forme de standardisation d'autre part - problématique sur laquelle nous reviendrons plus bas.

3.10.3 Aspects de gestion des apprenants.

L'une des faiblesses de la plupart des systèmes auteurs était leur orientation résolument monoposte : les didacticiels créés étaient pensés pour fonctionner sur une machine, et rarement sur un réseau. On peut dire que la flambée de popularité du WWW a fait gagner aux systèmes auteurs une étape de taille : avant même d'avoir pensé réseau local, les voilà soudain forcés de penser réseau mondial. La classe, de simple salle d'ordinateurs, est devenue classe virtuelle de dimension internationale. Le besoin d'outils de gestion des apprenants et des leçons (systèmes CMI, pour Computer Managed Instruction) - comment enregistrer un apprenant, comment lui

donner (ou lui refuser) l'accès à telle ou telle séquence d'apprentissage, comment conserver ses résultats, etc. - se fait donc brusquement sentir de façon d'autant plus aiguë que le dit apprenant peut se trouver physiquement à l'autre bout du monde.

Cette problématique commence certes à être prise en compte, de plus en plus, les systèmes auteurs s'assortissent de systèmes CMI, souvent sous la forme de produits complémentaires. Il en est ainsi par exemple de Librarian [Url 6], qui s'inscrit dans la gamme des produits Toolbook, et qui permet la gestion des modules d'apprentissage, permettant de régler les problèmes d'accès, de rassembler les données du suivi dans une base de données centralisée, etc. La communication étant assurée sur base des protocoles d'Internet, le système peut être utilisé aussi bien sur Internet que sur un Intranet local.

Dans le créneau plus spécialisé de l'évaluation, le CMI Perception, le dernier-né de la firme Question Mark [Url 7] créateur du système auteur du même nom, permet de réaliser des tests, enquêtes et examens à travers Internet. Une base de données sécurisée permet de gérer sessions, utilisateurs, groupes, etc. Le système CMI et le système auteur sont ici vendus ensemble.

Il reste cependant deux problèmes. Le premier est celui du prix : le coût encore élevé de tels produits (plus de 5000 US\$ pour Toolbook Librarian, plus de 2000 US\$ pour Perception Question Mark) les place trop souvent hors de portée des institutions d'enseignement. Le second est que, plus encore que pour les systèmes auteurs eux-mêmes, le risque est grand de voir le choix d'un système CMI lier l'utilisateur à un format propriétaire. Mais dans ce domaine également, les choses évoluent rapidement, et des propositions de standards - même si c'est sur papier - voient actuellement le jour. Ce qui nous amène au point suivant.

3.10.4 Aspects de standardisation

On sait le temps nécessaire pour produire un didacticiel de qualité : de 150 à 200 heures de production pour une heure d'utilisation [Uytebrouck 98]. Si ce temps a certes tendance à diminuer légèrement avec le perfectionnement des outils, il reste de toute façon élevé, avec pour corollaire que le coût d'un tel travail reste, lui aussi, très important. Dès lors, la protection des investissements consentis - à la fois des sommes investies et des moyens humains utilisés - demeure un enjeu crucial. Dans cet univers en mutation constante, le meilleur moyen de rentabiliser ces investissements serait clairement de pouvoir s'appuyer sur des standards. Le problème est qu'à l'heure actuelle, il n'existe toujours pas de standard de fait en matière de didacticiels. S'il existe des formats relativement universels pour les médias particuliers (rtf pour le

texte, gif ou jpeg pour les images, etc.), rien n'existe en revanche en matière d'interactivité, de navigation, de suivi pédagogique, etc. Chaque système enferme le concepteur dans un format propriétaire que les éditeurs, pour d'évidentes raisons de concurrence, n'ont guère envie d'ouvrir (si ce n'est au détriment des concurrents bien sûr). Sans grande exagération, on peut dire qu'à l'heure actuelle, porter un didacticiel d'un système vers un autre demande (d'un point de vue technique s'entend) un travail de réécriture à peu près complet. Quatre lueurs se dessinent cependant à l'horizon, même s'il est difficile de préjuger de ce que nous réserve l'avenir :

1) Le succès du World Wide Web a imposé un standard de fait en matière de présentation de l'information (le langage Html) et en matière de structuration de l'information (le langage Xml). Du côté interaction et adaptation des pages web, les langages Java, JavaScript, Jsp, Asp, Php et autres deviennent de plus en plus puissants avec leur nouvelle version et permettent de gérer des sessions vraiment conviviales que ce soit du côté enseignant que du côté apprenant.

2) On assiste à une tendance de plus en plus marquée des systèmes auteurs à conserver les médias en dehors des fichiers exécutables (Authorware, IconAuthor, Apple Media Tool, Media File System, etc.) Si ceci ne résout pas tous les problèmes, il est certain qu'il s'agit d'un pas vers la "réutilisation" des matériaux didactiques.

3) Un outil comme Synergy, que nous avons déjà mentionné et qui permet de convertir un scénario pédagogique en un format auteur donné, constitue une passerelle inattendue mais prometteuse entre les différents systèmes auteurs. La standardisation se réalise ici non plus au niveau des systèmes auteurs eux-mêmes, mais en amont, à l'aide d'un langage de description générique.

4) Enfin, l'idée de définir des standards ouverts spécialement adaptés aux applications pédagogiques fait lentement son chemin. Récemment, on a vu émerger une série d'initiatives dans ce sens :

a) Les exercices et tests du logiciel Perception sont stockés sous un format appelé QML (pour Question Markup Language), qui ressemble à du html et permet de délimiter les énoncés, les propositions de réponse, les rétroactions, etc. Il s'agit ici de l'initiative isolée d'une société commerciale, mais qui a placé les spécifications de QML dans le domaine public [Url 8] et dont on retiendra l'idée.

b) Beaucoup plus ambitieux, tant par le domaine qu'il couvre que par les forces qui le soutiennent, est le travail d'Educom consortium [Url 9] qui regroupe plus de 500

établissements d'enseignement supérieur, principalement aux Etats-Unis. Dans le cadre du projet IMS (Instructional Management System) [Url 10], Educom a, le 29 avril 1998, rendu public des spécifications touchant à peu près tous les aspects de l'enseignement via le Web. Le but est d'essayer de lancer un standard international qui garantirait l'interopérabilité de matériaux didactiques provenant de sources variées. Le principe général consiste à décomposer les matériaux didactiques en objets caractérisés par une série d'attributs (auteur, titre, langage, date de création, domaine, etc.). Ces "méta-données" permettent de décrire les objets de manière relativement fine, et facilite donc leur intégration dans un ensemble d'objets plus large. Théoriquement, il devient ainsi possible de construire un curriculum "à la carte" en assemblant des objets de sources variées. Le texte, bien évidemment disponible sur Internet [Url 11], a déjà le soutien d'un grand nombre d'acteurs majeurs du paysage informatique (Apple, IBM, Microsoft, Sun, Oracle,...).

- c) Enfin, le projet européen Ariadne [Url 12] constitue en quelque sorte le pendant européen du projet d'Educom. Si son ambition est plus limitée, il a l'originalité de proposer, outre un système d'indexation des ressources pédagogiques, une banque de ressources européennes appelée « Knowledge Pool System ».

3.11 Conclusion

Ce chapitre a présenté les systèmes auteurs dédiés à l'éducation. Pour ce type de logiciels, il apparaît clairement que nous nous trouvons à une période charnière, à un moment-clé où un monde s'efface à la faveur d'un autre. La période du système monoposte, ou des petits réseaux locaux fermés, est révolue. Internet a ouvert la voie à la classe planétaire et à de multitudes d'usages encore impensables hier. Cependant, il est clair qu'un tel bouleversement des technologies éducatives ne va pas sans son cortège de doutes et d'instabilités.

Les anciens outils de production de cours, qui commençaient à être fiables et bien rodés, ont dû soudain se reconverter pour être plus ou moins "compatibles" avec le Web. Avec le succès mitigé que l'on sait, le Web lui-même, en matière d'interactivité, nous ramène - ironiquement - une bonne décennie en arrière, et a sans doute fait naître plus de tutoriels constitués d'une simple suite de pages à tourner que de réelles innovations pédagogiques.

De tous côtés, la bataille est rude pour offrir l'outil qui rendra le Web plus interactif, mieux adapté aux exigences de la formation. La pédagogie, autrefois simple vernis, pénètre les outils. La mondialisation des ressources a fait prendre conscience de l'urgence de définir des standards;

même si ceux-ci ne sont encore que virtuels, la mode semble bien désormais aux systèmes ouverts et évolutifs.

Il fut un temps où les systèmes auteurs formaient un ensemble relativement clos et imperméable ; impossible aujourd'hui d'en parler sans évoquer une foule d'outils qui ne leur sont pas a priori liés (comme les éditeurs Web par exemple). Dans le même ordre d'idées, systèmes auteurs et systèmes de gestion des leçons et des apprenants ont longtemps été distincts; ils fusionnent aujourd'hui sur et via Internet. Et ce qui est vrai des outils l'est tout autant des démarches et des méthodes. Les didacticiels produits hier cantonnaient souvent les activités proposées à la transmission d'information et à l'exercice. Actuellement, les outils de communication intégrés au Web, la masse quasi infinie des connaissances disponibles en ligne, permettent de fusionner ces activités à d'autres, plus complexes, comme la production ou l'exploration.

Les autoroutes de l'information rendent possible le travail collaboratif - non seulement pour les apprenants comme démarche d'apprentissage, mais aussi pour les enseignants lors de la production des matériaux didactiques. Dans le chapitre suivant, nous présentons notre système auteur qui prend en considération cette dimension de « coopération » entre auteurs.



Chapitre 4

Spécification des besoins et modèle conceptuel

4.1 Introduction

Dans ce chapitre nous présentons un modèle de système auteur coopératif pour la construction de tuteurs intelligents hypermédias. Le canevas de STI qui lui est sous-jacent et qui est basé sur la pédagogie par objectifs, est réutilisable à une grande variété de domaines.

Nous rappelons que ce travail rentre dans un cadre de recherche plus large : la plate-forme d'enseignement collaboratif à distance Ibn Sina, dédiée aux universités algériennes qui font face à des effectifs de plus en plus grands et donc sont en quête de nouvelles méthodes et stratégies pour assurer leur tâche pédagogique. Les enseignements s'effectuant en grande partie en présentiel actuellement, doivent être repensés sous une autre forme « mixte », c'est à dire en partie sur campus en présentiel et en partie à distance via l'Internet. La plate-forme Ibn Sina a pour ambition de satisfaire la deuxième forme, en permettant un enseignement/apprentissage plus ouvert et sans contraintes spatio-temporelles.

Le modèle de système auteur que nous présentons dans cette thèse, n'est qu'une facette d'une multitude de fonctionnalités offertes dans la plate-forme. Il permet via l'Internet, d'une part aux enseignants et formateurs de coopérer pour construire des systèmes tuteurs, et d'autre part, permet-il à l'étudiant de prendre en charge sa formation à son rythme soit en individuel soit en collaboration avec des tiers (étudiants ou formateurs). Etant donné que le système est destiné à toutes les disciplines et que la grande majorité des enseignants et étudiants ne possède pas de compétences en informatique, ils doivent donc être libérés de toute activité de programmation. Pour ce faire un modèle de STI générique et réutilisable est mis au point. Grâce à des interfaces conviviales de part et d'autres, l'auteur n'est sollicité que pour instancier ce canevas avec toutes les connaissances et données nécessaires à l'apprentissage. L'étudiant lui, n'est appelé qu'à interagir et dialoguer avec le STI en vue d'un apprentissage individualisé et adaptatif.

En ce qui concerne l'aspect pédagogique à adopter, nous avons décidé, dans le but de faciliter l'utilisation du système, de reproduire (ou de calquer) la même approche pédagogique

d'enseignement employée dans le cycle universitaire, mais aussi, de conserver au maximum la même terminologie utilisée par les enseignants. Cette approche, dont nous décrivons les principes dans la section suivante, s'appuie sur la notion d'objectif pédagogique et est indépendante du domaine à enseigner dans une large mesure.

4.2 Une pédagogie réutilisable : la pédagogie par objectifs

Selon Tagliante [Tagliante 91], la Pédagogie Par Objectifs (PPO) est née aux Etats Unis au début de ce siècle et elle a connu un grand succès depuis une trentaine d'années en Europe et dans le reste du monde. Cette méthodologie consiste en général à décomposer un module d'enseignement complexe en ses éléments simples et essentiels afin d'en faciliter l'enseignement/apprentissage et l'évaluation. La décomposition est effectuée sur la base d'objectifs pédagogiques.

4.2.1 Qu'est-ce qu'un objectif pédagogique ?

En pédagogie, un objectif est un énoncé d'intention décrivant ce que l'apprenant saura (ou saura faire) après apprentissage. Les objectifs sont normalement dérivés des **finalités** de l'éducation et des **buts** de formation, lesquels se décomposent en objectifs intermédiaires de différents niveaux.

En d'autres termes, un objectif pédagogique (OP) est une expression claire qui décrit en terme de comportement, ce que l'apprenant doit être capable de réaliser à la fin de son activité d'apprentissage. Par exemple « à la fin de ce module, l'apprenant doit être capable de concevoir une base de données relationnelle ». Les objectifs pédagogiques peuvent être définis à plusieurs niveaux hiérarchiques. En partant d'un objectif global « But », le nombre et l'appellation de ces niveaux diffèrent selon les auteurs. Nous avons retenu une hiérarchie à trois niveaux (**Figure 4.1**) définie dans [Hameline 90] et [Tagliante 91] : les « objectifs généraux », les « objectifs spécifiques » et les « objectifs opérationnels ».

- *Le But Pédagogique* est un énoncé définissant de manière générale les intentions poursuivies soit par une institution, soit par une organisation, soit par un groupe, soit par un individu à travers un programme ou une action de formation. Exemple « l'enseignement du module bases de données aux ingénieurs informaticiens a pour but de développer chez l'apprenant les capacités d'analyser et de concevoir les bases de données appliqués à la gestion et de comprendre le fonctionnement des SGBD ».

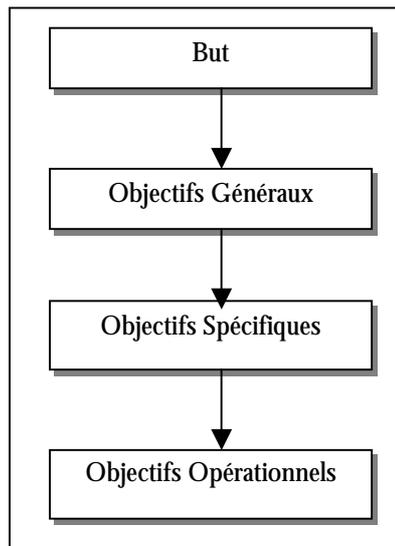


Figure 4.1. Hiérarchie des 3 niveaux d'objectifs pédagogiques d'un but pédagogique

- *Les objectifs généraux* (OPG) indiquent en terme de capacités, les résultats escomptés à la fin de l'activité d'apprentissage d'un module ou d'une unité d'enseignement. Un objectif général est généralement un énoncé d'intention pédagogique relativement large, il est également appelé objectif terminal d'intégration. Par exemple « à la fin de cette séquence l'apprenant devra être capable de normaliser une base de données ».
- *Les objectifs spécifiques* (OPS) affinent un objectif général en le démultipliant en autant d'objectifs spécifiques qu'il est nécessaire pour que l'apprenant l'atteigne. Par exemple « à la fin de cette séquence l'apprenant devra être capable de mettre une relation en 2^{ième} forme normale et en 3^{ième} forme normale ». On remarque que ces deux objectifs spécifiques affinent l'objectif général précédent.
- *Les objectifs opérationnels* (OPO) affinent à leur tour un objectif spécifique de telle façon qu'ils soient directement évaluables. Par exemple « à la fin de cette séquence d'apprentissage, l'apprenant sera capable de montrer, en 5 minutes et sans s'aider de documents, que la relation possède au moins une clé et que toutes les dépendances fonctionnelles de la clé vers les autres propriétés de la relation sont élémentaires ». On remarque que ces deux objectifs opérationnels affinent l'objectif spécifique précédent qui vise la capacité de mettre en 2^{ième} forme normale une relation.

Sur le plan de la formulation des objectifs, deux éléments sont à considérer. D'une part, cet objectif porte sur une partie principale de la matière enseignée, d'autre part, l'objectif est associé à l'une ou l'autre des taxonomies : il doit refléter un des différents niveaux taxonomiques selon

l'intention de celui qui le formule. Pour permettre de mieux cerner la nature de ce qui est attendu, il s'est développé des taxonomies qui portent sur l'un ou l'autre des domaines du savoir : **le domaine cognitif** concerne les connaissances et les habilités intellectuelles, **le domaine affectif** est associé aux attitudes, aux valeurs, aux intérêts, aux représentations, **le domaine psychomoteur** concerne les habilités motrices.

Selon Hameline [Hameline 90], pour qu'un OP soit véritablement opérationnel, quatre principes doivent être respectés lors de la formulation de cet OP :

- **l'univocité** : vérifier que rien n'est équivoque dans la formulation de l'objectif, c'est à dire aucune ambiguïté;
- **le comportement observable** : décrire une activité de l'apprenant identifiable par un comportement observable. Dans la formulation, on utilisera un verbe d'action qui aura l'apprenant pour sujet et qui permet de visualiser le résultat. On doit proscrire les verbes mentalistes qui ne permettent pas l'observation du comportement (comme: connaître, savoir, comprendre, apprécier, etc.).
- **les conditions** : mentionner les conditions dans lesquels le comportement souhaité de l'apprenant doit se manifester. Par exemple: « . . . , en cinq minutes sans l'aide de documents, . . . ».
- **le critère** : il s'agit ici de déterminer le seuil d'erreurs acceptées pour qu'on puisse considérer l'objectif comme néanmoins atteint.

Afin de bien distinguer un objectif opérationnel d'un autre qui ne l'est pas, nous allons présenter ici les caractéristiques que doit posséder un tel objectif :

- Il est toujours exprimé en fonction de l'apprenant.
- Il est univoque, c'est-à-dire précis et n'acceptant qu'une seule interprétation.
- Il décrit un comportement observable chez le sujet.
- Il précise, s'il y a lieu, les conditions spéciales de manifestation de ce comportement et les critères qui permettront de juger si l'objectif est atteint.

L'objectif opérationnel est exprimé en fonction de l'apprenant

Le fait d'exiger qu'un objectif soit exprimé en fonction de l'apprenant pour qu'il puisse être désigné comme étant opérationnel signifie qu'il faut se concentrer sur l'apprentissage de l'étudiant plutôt que sur l'enseignement du formateur. En effet, il ne suffit pas que le formateur ait donné son enseignement, il faut que ses apprenants aient appris.

Il serait souhaitable que cette exigence liée à la rédaction d'objectifs incite à la réflexion sur le lien enseignement apprentissage. En effet, le formateur devrait chercher à préciser ses objectifs non plus en se concentrant sur ce qu'il enseigne, mais sur les changements qu'il veut susciter chez ses apprenants.

L'objectif opérationnel est univoque.

L'objectif opérationnel doit être univoque, c'est-à-dire qu'il doit être précis et n'avoir qu'une seule signification. Les expressions nébuleuses sont sujettes à une interprétation libre. L'objectif doit exprimer clairement ce que l'étudiant doit faire. Ce n'est qu'à cette condition que l'objectif remplisse son rôle :

- assurer une communication claire entre le formateur et les apprenants et entre les formateurs eux-mêmes;
- aider le formateur à choisir les activités d'apprentissage et d'évaluation appropriées.

Un objectif tel que « l'apprenant fera preuve de curiosité » est sujet à interprétation. Il en serait de même pour l'objectif suivant: « l'étudiant sera capable de concevoir une base de données ». Afin d'arriver à rendre ces objectifs univoques, il faut se demander ce que l'étudiant fera pour démontrer qu'il fait preuve de curiosité ou qu'il est capable de concevoir une base de données.

Certains termes peuvent sembler sans équivoque et c'est à l'usage qu'on s'aperçoit qu'il faut les préciser davantage pour que l'objectif soit univoque. Des termes généraux comme connaître, savoir, comprendre, se familiariser, prendre conscience, apprécier, etc., prêtent à diverses interprétations. Il faut spécifier ce à quoi on s'attend de la personne qui connaît, sait ou comprend, afin d'inférer qu'elle connaît, sait ou comprend. Des termes aussi vagues renvoient plutôt à des objectifs généraux ou à des objectifs spécifiques.

Pour clarifier l'objectif spécifique, on déduit de celui-ci un échantillon d'objectifs opérationnel. On choisit parmi diverses possibilités les objectifs qui paraissent les plus pertinents, compte tenu du contexte, pour permettre d'atteindre l'objectif spécifique dont ils découlent. Par exemple, l'objectif spécifique « comprendre la signification de termes techniques » peut engendrer les objectifs suivants: « définir le terme en ses propres mots », « identifier le sens d'un terme quand il est utilisé dans un contexte », « distinguer entre des termes ayant des significations similaires ». Les utilisateurs doivent donc s'entendre sur la terminologie utilisée et rendre ainsi l'objectif univoque.

L'objectif opérationnel décrit un comportement observable.

L'objectif doit décrire une capacité que le sujet doit avoir acquise à la fin de son apprentissage. Il s'agit de préciser une capacité observable et non pas une catégorie de comportements comme le suggèrent les mots "comprendre" et "savoir". On doit pouvoir se représenter cette capacité au moyen d'une action concrète, observable. Ainsi, un objectif comme "voir la valeur de telle expérience" n'est pas opérationnel.

Il faut faire ici la différence entre le processus et le résultat, entre les moyens et ce qu'on veut atteindre. Par exemple, "découvrir tel principe de classification" n'est pas un objectif, mais un processus d'apprentissage qui ne peut être répété; "lire tel texte" n'est pas un objectif, mais une activité, une tâche à accomplir.

Il faut également distinguer les objectifs d'une liste de contenus, comme on en trouve dans certains plans de cours. En effet, un objectif comprend au moins trois parties : un sujet (l'apprenant), un verbe d'action (le comportement observable qui est attendu) et un complément (le contenu). Ainsi, dans l'objectif opérationnel « l'apprenant devra être en mesure d'énumérer les différents types de modèles de SGBD existants », le sujet est bien l'apprenant, le verbe énumérer réfère à un comportement observable et le contenu est lié aux types de modèles de SGBD. On ne peut donc confondre des objectifs et des listes de contenus. Il faut que le verbe et le complément soient précis pour que l'objectif soit opérationnel.

L'objectif opérationnel précise les conditions de manifestation du comportement

Cette caractéristique signifie que l'objectif doit, s'il y a lieu, faire mention des conditions dans lesquelles le comportement doit se manifester et des critères qui serviront à déterminer si l'objectif est atteint ou non.

Les conditions spéciales relatives à la manifestation du comportement sont toutes les conditions qui peuvent faire varier la difficulté d'un objectif et qui permettent de concevoir une situation concrète dans laquelle le comportement doit se manifester. S'il n'est pas nécessaire que tous les objectifs fassent mention de telles conditions spéciales, certains objectifs, l'exigent. Par exemple, l'objectif: « l'étudiant sera capable de dactylographier à la vitesse de quatre-vingt-dix mots à la minute » ne pourrait être suffisamment précis pour être opérationnel si les conditions de manifestation relatives au temps n'étaient pas indiquées.

L'indication des critères qui serviront à déterminer si la performance est satisfaisante est, elle aussi, facultative. Atteindre ou non un objectif s'exprime parfois en termes de tout ou rien

(exemples: on est capable ou non d'énumérer les quatre caractéristiques obligatoires d'un objectif opérationnel). Dans ces cas, on n'indique pas de critères quantitatifs pour vérifier si la performance a été atteinte; les critères qualitatifs mentionnés ou sous-entendus suffisent. Par contre, on doit parfois indiquer un critère exprimé en pourcentage ou en proportion de "bonnes réponses" à l'instrument d'évaluation de l'objectif; il peut aussi s'agir de la déviation acceptable établie à partir d'une norme donnée (exemple: couper une pièce de métal en acceptant une tolérance de 0,0150 mm dans les dimensions).

L'opérationnalisation des objectifs est faite surtout dans un souci d'évaluation car il est difficile d'évaluer l'atteinte d'un objectif spécifique, et plus difficile encore celle d'un objectif général.

Les quatre exigences opérationnelles sus-citées (univocité, comportement observable, condition, critère) facilitent en effet, grandement la tâche de l'enseignant qui doit élaborer ses tests d'évaluation, en particulier en ce qui concerne les qualités les plus difficiles à atteindre: la validité du test et l'objectivité de l'évaluation. En conséquence, clarté des objectifs et objectivité d'évaluation sont deux notions solidaires et il ne saurait être question d'un enseignement de qualité tant que ces deux notions ne sont pas réunies [Louanchi 87].

Pour l'enseignant comme pour l'apprenant, si l'évaluation est intégrée à l'apprentissage, du début à la fin du cursus, l'évaluation va être un appui, une aide. Elle ne sera plus une sanction mais plutôt un outil, dont on se servira pour construire l'apprentissage en sachant vraiment où l'on va. Le premier, en faisant son diagnostic, ne pourra ignorer les lacunes et les points forts du second et ce dernier saura à tout moment où il en est, sur quoi s'appuyer et l'étendue de l'effort qui lui reste à fournir [Tagliante 91].

Cette évaluation continue, que nous privilégierons dans notre modèle de système auteur, est appelée « **évaluation formative** » par opposition à « **l'évaluation sommative** » qui n'intervient qu'au terme d'un cursus d'apprentissage en vue de certifier un niveau ou une maîtrise de compétences. Les spécialistes s'accordent généralement pour définir trois grandes fonctions de l'évaluation [Tagliante 91]:

- **le pronostic** qui intervient avant la phase d'apprentissage et a pour rôle d'évaluer les prérequis de l'apprenant ;
- **le diagnostic** qui intervient pendant la phase d'apprentissage et a pour rôle de réguler l'enseignement ;

- **l'inventaire** qui intervient après la phase d'apprentissage et a pour rôle d'évaluer la somme des connaissances acquises.

Aussi, l'évaluation nécessite-t-elle des outils pour sa mise en oeuvre. Ces outils sont extrêmement divers. Des plus fermés comme les Questions aux Choix Multiples (QCM) aux plus ouverts comme les sujets de synthèse, chaque enseignant peut, dans son domaine, concevoir et élaborer des outils nouveaux, adaptés aux objectifs qu'il souhaite évaluer. Des outils comme test lacunaire, test d'appariement, test de classement sont désormais devenus classiques.

4.2.2 Les taxonomies :

Une taxonomie est un outil de traitement ou d'organisation d'un cours dans une perspective d'évaluation. En pédagogie, on parle de taxonomie d'objectifs pédagogiques pour classer les niveaux de définition de ces objectifs. Une taxonomie présente en général :

- Un principe de classement d'objectifs ;
- Une classification hiérarchisée qui met en œuvre ce principe en produisant des catégories ;
- Des exemples illustrant ces catégories.

Benjamin BLOOM [Bloom 56], célèbre pédagogue américain, est le père de la première classification hiérarchisée ou taxonomie des objectifs pédagogiques. Du fruit de ses travaux émerge une classification des niveaux de pensée qu'il considère comme importants dans le processus d'apprentissage. Bloom fait l'hypothèse que les habiletés peuvent être mesurées sur un continuum allant de simple à complexe. La taxonomie des objectifs pédagogiques de Bloom est composée de six niveaux, dont : la connaissance, la compréhension, l'application, l'analyse, la synthèse et l'évaluation (**table 4.1**).

La taxonomie des objectifs pédagogiques de Bloom peut s'avérer intéressante pour des enseignants et formateurs. Ces derniers peuvent concevoir des activités pédagogiques et présenter les informations selon les niveaux de pensée (**table 4.2**). Plusieurs de ces activités peuvent être prises en considération lors de la conception d'un cours dans un nouvel environnement de formation et d'apprentissage.

Niveau	Mots-clés
1. Connaissance	<ul style="list-style-type: none"> • Observe et se souvient de l'information • Arrange, définit, décrit, associe, ordonne, retient, nomme, note, répète, etc.
2. Compréhension	<ul style="list-style-type: none"> • Comprend l'information et en saisit le sens • Altère, change, classe, définit dans ses propres mots, discute, explique, donne des exemples, traduit, etc.
3. Application	<ul style="list-style-type: none"> • Utilise l'information et utilise des méthodes • Applique, calcule, construit, pratique, etc.
4. Analyse	<ul style="list-style-type: none"> • Voit des modèles et organise les parties • Analyse, évalue, catégorise, compare, conclue, contraste, critique, pose un diagnostic différentiel, etc.
5. Synthèse	<ul style="list-style-type: none"> • Utilise de vieilles idées pour en créer de nouvelles • Assemble, compose, crée, améliore, synthétise, etc.
6. Évaluation	<ul style="list-style-type: none"> • Compare et discrimine les idées • Évalue, argumente, choisit, certifie, critique, décide, déduit, défend, distingue, évalue, recommande, etc.

Table 4.1. Taxonomie des objectifs pédagogiques de Bloom

Niveau	Activités
1. Connaissance	Mémoriser des informations, définir des terminologies, des techniques, etc.
2. Compréhension	Comprendre un article afin d'en faire un résumé.
3. Application	Utiliser les connaissances acquises pour les appliquer dans une situation concrète.
4. Analyse	Demander à l'apprenant de disséquer un sujet, d'en expliquer les tenants et les aboutissants.
5. Synthèse	Reformuler les parties d'un sujet ensemble mais d'une toute nouvelle manière en se basant sur plusieurs sources.
6. Évaluation	Juger la valeur d'un sujet dans un but spécifique

Table 4.2. Taxonomie des objectifs et activités pédagogiques

4.2.3 Les dix avantages de la pédagogie par objectifs selon [Mager 75]

- 1) C'est la seule méthode valable de planification rationnelle en pédagogie, car elle construit la programmation et la progression autour de l'activité de l'apprenant.

- 2) Elle oblige les enseignants, en particulier ceux qui ont la charge de confectionner des programmes, à penser et à préparer les activités de façon spécifique et détaillée.
- 3) Elle encourage à expliciter les valeurs, les désirs, les choix jusque là évacués dans le « non dit ».
- 4) Elle fournit une base rationnelle pour l'évaluation formative et permet l'autoformation.
- 5) Elle subordonne le choix des moyens d'enseignement aux objectifs d'apprentissage, inversant une situation assez courante.
- 6) Elle forme la base d'un système qui s'améliore lui-même par un constant feedback.
- 7) Elle permet à ce système, tout en assurant sa mobilité, d'acquérir une certaine consistance interne, non par l'effet rhétorique du discours de l'enseignant, mais par l'articulation des tâches des apprenants sur les objectifs sans cesse approfondis des apprentissages.
- 8) Elle permet de faire sortir les buts de l'éducation du domaine des vœux théoriques et leur donne un champ de réalisation pratique.
- 9) Elle permet la communication entre enseignants et enseignés et avec les autres partenaires de l'éducation (parents, administration, collègues, etc.), sous le signe de la clarté, et permet un contrat bilatéral de formation que l'évaluation finale des apprentissages comme de l'enseignement viendra vérifier.
- 10) Elle permet d'établir les bases d'un apprentissage individualisé.

En résumé, nous pouvons dire que la PPO est une méthode pédagogique qui construit la planification et la progression de l'enseignement/apprentissage autour de l'activité de l'apprenant. De plus, ce dernier peut lui-même s'auto-évaluer puisqu'il sait à l'avance quels objectifs on se propose d'atteindre avec lui et à quoi il reconnaîtra lui-même que le résultat est atteint. D'arbitraire, la relation éducative enseignant/apprenant devient coopérative [Louanchi 87]. Dans la section suivante nous justifions notre choix quant à la pédagogie par objectifs en apprentissage à distance.

4.3 Pédagogie par objectifs et apprentissage à distance

Selon les différentes fonctions qui leurs sont dévolues, les activités d'apprentissage que l'on retrouve dans les cours de formation à distance ont des caractéristiques qui diffèrent. Leurs formes se distinguent : questions à choix multiples, questions ouvertes, énoncés de procédures ou exercices à l'écran. La réponse que l'étudiant doit produire ou l'action qu'il doit effectuer sont de natures variées : du repérage à la résolution de problème, de la définition d'objectifs personnels à la manipulation d'appareil. Toutes ces activités sont développées par des enseignants qui

prennent des décisions en fonction de leurs intentions pédagogiques. Ces intentions sont généralement associées à des objectifs pédagogiques.

Les objectifs pédagogiques sont devenus un outil indispensable pour l'auteur de cours à distance [Legendre 93]. Une des premières étapes du développement pédagogique consiste à se demander quel est le but de l'apprentissage, quelle habileté, compétence ou comportement vise-t-on à développer chez l'apprenant ou encore quelle activité cognitive désire-t-on stimuler. Si on s'entend sur la nécessité d'avoir un objectif centré sur l'étudiant et l'apprentissage, les auteurs divergent quant au contenu des objectifs et à leur formulation.

L'approche par objectifs a le mérite de se centrer sur l'apprentissage plutôt que sur l'enseignement. Elle oriente le processus de formation sur l'apprenant et favorise une individualisation de l'enseignement en fonction du rythme de l'apprenant.

Pour correspondre au critère « observable et mesurable », cette approche oriente la situation d'apprentissage sur le résultat. Elle permet de mesurer les progrès de l'apprenant de façon précise et d'atteindre ainsi une évaluation dite « objective » [Saint-Onge 92].

Les taxonomies des objectifs pédagogiques permettent d'aider l'auteur de cours à vérifier la portée de ce qu'il propose aux apprenants [Legendre 93]. Bloom en 1956 constatait que 80 % des questions d'examen correspondaient aux deux niveaux inférieurs de sa grille taxonomique cognitive (connaître et comprendre). Malgré l'engouement pour la pédagogie par objectifs, la même étude reprise par d'autres vingt-cinq ans plus tard confirmait que les choses n'ont pas changé, l'accent est souvent mis sur les plus faibles niveaux des taxonomies [Schierman et al. 92].

Plusieurs auteurs de guides destinés aux auteurs des cours à distance proposent d'inclure des objectifs explicites pour accompagner les activités d'apprentissage dans les documents destinés aux apprenants à distance. Ils constituent des indices pour assister la lecture et pour organiser le contenu [Rowntree 94] [Zubot 93]. Ils y voient un excellent moyen de communiquer les intentions pédagogiques du concepteur du cours et de guider l'apprenant [Dessaint 95].

Selon Tardif [Tardif 92], la discussion des objectifs de la « tâche à accomplir » est une étape de la phase de préparation à l'apprentissage dans le modèle de l'enseignement stratégique. Cette phase « permet à l'apprenant de donner un sens à ce qui lui est demandé ». La communication des objectifs pourrait également favoriser l'autonomie des étudiants à distance. Deschênes [Deschênes 91] affirme qu'il est important que les apprenants soient informés explicitement de

toutes ces décisions; mieux encore, il faut solliciter leur participation en leur permettant de se situer par rapport aux objectifs généraux et de définir leurs propres sous-objectifs.

La présence des objectifs joue aussi un rôle positif sur la motivation et l'état affectif des apprenants à distance [Landry 86]. Deschênes et ses co-auteurs [Deschênes & al 93] proposent : « un balisage clair de la démarche d'apprentissage (contenu, concepts, objectifs) qui répondront au besoin d'être rassuré ». Des facteurs importants de la motivation, telle que la définit Tardif [Tardif 92], comme la conscience de l'utilité de la « tâche à accomplir », la perception de la valeur et des exigences de la tâche, peuvent être favorisés par la présence des objectifs.

Une étude menée par Jegede et ses collaborateurs [Jegede & al 95] auprès de 286 étudiants universitaires à distance révèle que ceux-ci utilisent les objectifs : ils s'en servent pour guider leurs études (ils sont l'énoncé des attentes du cours et aident à mesurer leur compréhension du matériau pédagogique). Les étudiants affirment également qu'il est nécessaire d'inclure les objectifs dans la matière d'étude au début, durant et à la fin.

Cette perception positive des objectifs corrobore une recherche de Martens et Valke [Martens et Valke 93] dans laquelle les objectifs sont notés comme utilisés par 80% d'un groupe d'étudiants universitaires à distance et servent à l'apprentissage en profondeur dans 65% des cas.

La communication des objectifs opérationnels présente cependant quelques désavantages. Elle pourrait limiter l'étudiant à ce qui est indiqué dans l'objectif. Dans une recherche élaborée par Melton in [Saint-Onge 92], des étudiants à qui on avait communiqué un objectif avant une lecture ont par la suite démontré une connaissance plus superficielle du texte. Il semble que ces étudiants se soient limités à sélectionner dans le texte les informations permettant de répondre à l'objectif. Ce cas démontre cependant davantage les limites des objectifs comportementaux qu'une conséquence défavorable de la communication des objectifs. En effet, comment pourrait-on reprocher aux étudiants d'utiliser les objectifs comme guide et de concentrer leurs efforts vers l'objectif si celui-ci décrit l'intention pédagogique véritable? Le problème se pose si on utilise un objectif superficiel, mais qu'on juge ensuite le travail de l'étudiant sur le manque de profondeur de sa compréhension.

Les désavantages liés à la communication des objectifs tiennent davantage au type de formulation et à l'orientation pédagogique qu'ils reflètent. Si la nécessité des objectifs définis selon les critères comportementaux classiques ne fait pas l'unanimité, la clarté de la démarche proposée semble nécessaire. Certains soutiennent d'ailleurs qu'il est important d'utiliser un moyen

de communiquer les intentions d'apprentissage mais que sa forme exacte n'est pas critique [Misanchuk 94] et peut inclure des objectifs flous (fuzzy objectives) [Zubot 93], pouvant correspondre aux domaines mal définis de la connaissance.

Notre choix de la pédagogie par objectifs comme théorie d'apprentissage indispensable dans le contexte de la formation à distance étant établi, nous montrons dans la section suivante comment cette théorie mise en pratique durant le 20^{ième} siècle nous inspire dans la conception de notre modèle. Nous commençons par décrire les besoins fonctionnels de notre système auteur.

4.4 Définition des besoins fonctionnels

Tout logiciel est conçu pour satisfaire un certain nombre de besoins des utilisateurs. En ce qui nous concerne, nous exprimons ces besoins en termes de services offerts aux deux acteurs potentiels de l'activité d'enseignement/apprentissage qui sont : les auteurs du cours (i.e. les enseignants ou formateurs) d'un côté et les apprenants de l'autre côté (**Figure 4.2**).

4.4.1 Expression des besoins du côté de l'auteur

BAU1. Le système auteur devra fournir à l'*auteur de cours*, via Internet, un ensemble d'outils qui soit convivial et facile à utiliser, pour assurer les tâches suivantes pour chaque module d'enseignement :

- Saisie et maintenance d'une « base de documents d'apprentissage du module d'enseignement » (BDAM). Cette base contient les documents d'apprentissage eux-mêmes (DAM : Documents d'Apprentissage du Module) et les métadonnées concernant ces documents (METADAM).
- Saisie et maintenance d'une « base de connaissances pédagogiques » (BCOPEM) permettant de décrire la manière d'utiliser et d'enchaîner les documents d'apprentissage.
- Saisie et maintenance d'une « base de réseaux de prérequis » (BREP) décrivant trois points de vue de parcours du module d'enseignement à savoir : le *réseau de prérequis des documents d'apprentissage* (REPDAM), le *réseau de prérequis des objectifs pédagogiques* (REPOP) et le *réseau de prérequis des concepts* (REPC).
- Communication synchrone et/ou asynchrone avec d'autres auteurs ou formateurs distants en vue de collaborer avec eux pour la création et la maintenance des bases BDAM, BCOPEM et BREP. Des outils de communication et de partage de ressources du type messagerie électronique, chat, et éditeurs partagés sont donc nécessaires.

- Communication avec des apprenants qui sont en quête d'autres explications, d'autres ressources d'informations ou qui ont besoin d'être corrigés ou assistés lors de la résolution de problèmes. Des outils du même genre que les précédents devraient être fournis.

BAU2. La base DAM est constituée de deux composants : la matière théorique formée des documents d'apprentissage THEoriques (DAMTHE) et la matière d'évaluation formée des documents d'apprentissage d'EVALuation (DAMEVA). Chaque document d'apprentissage que ce soit dans DAMTHE ou dans DAMEVA est conçu sous forme d'une page hypermédia et doit satisfaire un ou plusieurs objectifs pédagogiques opérationnels fixés par l'auteur. Tous les vecteurs d'information peuvent donc être utilisés dans un tel document : texte simple, texte mis en forme, image fixe, image animée, son, vidéo et programme sous forme de script ou d'applet.

BAU3. Pour un module donné du cursus, la matière théorique d'apprentissage (DAMTHE) est structurée en trois niveaux hiérarchiques :

- les *parties* (satisfaisants le objectifs généraux) et comportant un ou plusieurs chapitres;
- les *chapitres* (satisfaisant les objectifs spécifiques) et comportant un ou plusieurs unités d'apprentissage (uah) ;
- les *uah* (satisfaisant les objectifs opérationnels) représentant les unités élémentaires d'apprentissage hypermédiés qui peuvent être évaluées.

BAU4. Quant à la matière d'évaluation (DAMEVA), elle est liée à la matière théorique d'apprentissage et permet au système d'adapter le tutorat au niveau de progression de l'apprenant, elle contient les entités suivantes :

- *un pretest-module* : permettant une évaluation *pronostic* du module ;
- *des diagtests-uahs* : permettant une évaluation *diagnostic* des unités d'apprentissage ;
- *des posttests-chapitres* : permettant une évaluation *inventaire* des chapitres ;
- *des posttests-parties* : permettant une évaluation *inventaire* des parties ;
- *et un posttest-module* : permettant une évaluation *inventaire* du module.

BAU5. Il est prévu deux modes d'apprentissage à l'apprenant : le mode exploration libre (apprentissage sans évaluation) et le mode formation (apprentissage avec évaluation). Seule la matière théorique d'apprentissage (DAMTHE) et les connaissances pédagogiques (BCOPEM) sont donc obligatoires. C'est le minimum de données requises pour construire un STI. L'absence des documents d'évaluation DAMEVA signifie que le STI ne peut être utilisé qu'en mode exploration. Leur présence n'interdit pas pour autant l'apprenant à utiliser ce mode.

BAU6. En vue de permettre à l'auteur de contrôler le tuteur du point de vue affichage en mode apprenant, tous les services offerts à l'apprenant doivent également être offerts à l'auteur.

4.4.2 Expression des besoins du côté de l'apprenant

BAP1. Comme il a déjà été mentionné dans la formulation du besoin (**BAU5**) ci-dessus, il est prévu deux modes d'apprentissage pour l'apprenant : le mode exploration et le mode formation. Le système doit donc fournir tous les outils nécessaires pour assurer ces deux besoins de l'apprenant. En mode exploration, l'apprenant est avant tout concerné par l'aspect recherche d'informations. Il est alors d'abord un fouineur plus qu'un lecteur. La matière d'apprentissage est pour lui un minerai qu'il explorera librement sans aucun guidage du système. Ceci est très caractéristique des premières prises de contact avec le tuteur. Des outils doivent donc être fournis à l'apprenant pour naviguer librement à travers les différents documents d'apprentissage. En mode formation, l'apprenant a besoin d'être guidé, assisté et évalué dans son apprentissage pour savoir, à tout moment, où il en est, et qu'est ce qui lui reste à faire. Le système doit alors être capable de prodiguer cette aide. Tout ce qui suit concerne spécialement le *mode formation*.

BAP2. Afin de permettre au système de générer automatiquement une leçon adaptée à chaque apprenant, une session d'apprentissage doit toujours débiter par une *négociation* de deux paramètres : le « *type de parcours* » et le « *point d'entrée* » dans le module d'enseignement. L'apprenant possède trois choix de *type de parcours* : *parcours basé sur les composants* (par exemple exprimer le désir de débiter sa formation à partir de telle partie, de tel chapitre ou de telle uah), *parcours basé sur les objectifs pédagogiques* (par exemple décider de commencer par atteindre tel ou tel objectif pédagogique) et *parcours basé sur les concepts* (par exemple décider de commencer à se former à tel ou tel concept). Cependant, la décision de démarrer la formation par un certain niveau (partie, chapitre, uah, objectif pédagogique, concept) est conditionnée par l'acquisition (i.e. compréhension) du niveau qui lui est prérequis. Des évaluations de prérequis doivent donc être effectuées avant de satisfaire le désir de l'apprenant.

BAP3. Aussi le système doit permettre à l'apprenant de négocier les endroits où il veut subir les tests au cours de sa formation : en fin du module seulement, en fin des parties seulement, en fin des chapitres seulement, en fin des uah seulement ou n'importe quelle combinaison de ces endroits. Avec ces possibilités, l'apprenant n'aura donc pas à négocier seulement le *point d'entrée* et le *type de parcours* mais également le degré d'évaluation qu'il désire subir. Ce degré est alors choisi entre les deux extrêmes : 0 pour une formation totalement libre (i.e. exploration libre) et 5 pour une formation totalement guidée par l'évaluation.

BAP4. La matière contenue dans un module d'apprentissage nécessite souvent des connaissances prérequis. La formation au module peut donc être précédée par un *pretest de pronostic* permettant de vérifier ces prérequis. L'apprenant ne peut alors suivre le module que s'il réussit à ce test.

BAP5. Certaines uah sont évaluables (i.e. des questions et/ou des exercices leur sont attachés), d'autres ne le sont pas. Une uah non évaluable est supposée acquise dès que l'apprenant affirme en avoir terminé la lecture. L'étude d'une uah évaluable, en revanche, est toujours suivie d'un test de diagnostic. L'apprenant n'est alors acheminé vers l'uah suivante que si le système juge que l'uah courante est acquise et ceci après certaines remédiations éventuellement.

BAP6. Que ce soit en mode exploration ou en mode formation, l'apprenant a besoin de communiquer d'une manière synchrone ou asynchrone avec des tiers (d'autres apprenants ou des formateurs distants) en vue de collaborer avec eux durant certaines activités d'apprentissage tels que la résolution de problèmes par exemple. Des outils de communication et de partage de ressources du type messagerie électronique, chat, et forums sont donc nécessaires dans l'environnement d'apprentissage.

BAP7. Enfin, le système doit permettre à tout moment à l'apprenant, de prendre momentanément l'initiative du dialogue pour effectuer certaines *requêtes* en cas de besoin. Une *requête* exprime par exemple le désir :

- d'arrêter le déroulement du tuteur ;
- de consulter des ressources d'information annexes localement ou à travers le web (dictionnaires, lexiques, index, des sites web, etc.) ;
- d'obtenir de l'aide du système ;
- de revenir en arrière dans le but de renforcer ses connaissances ;
- de communiquer avec des tiers ;
- etc.

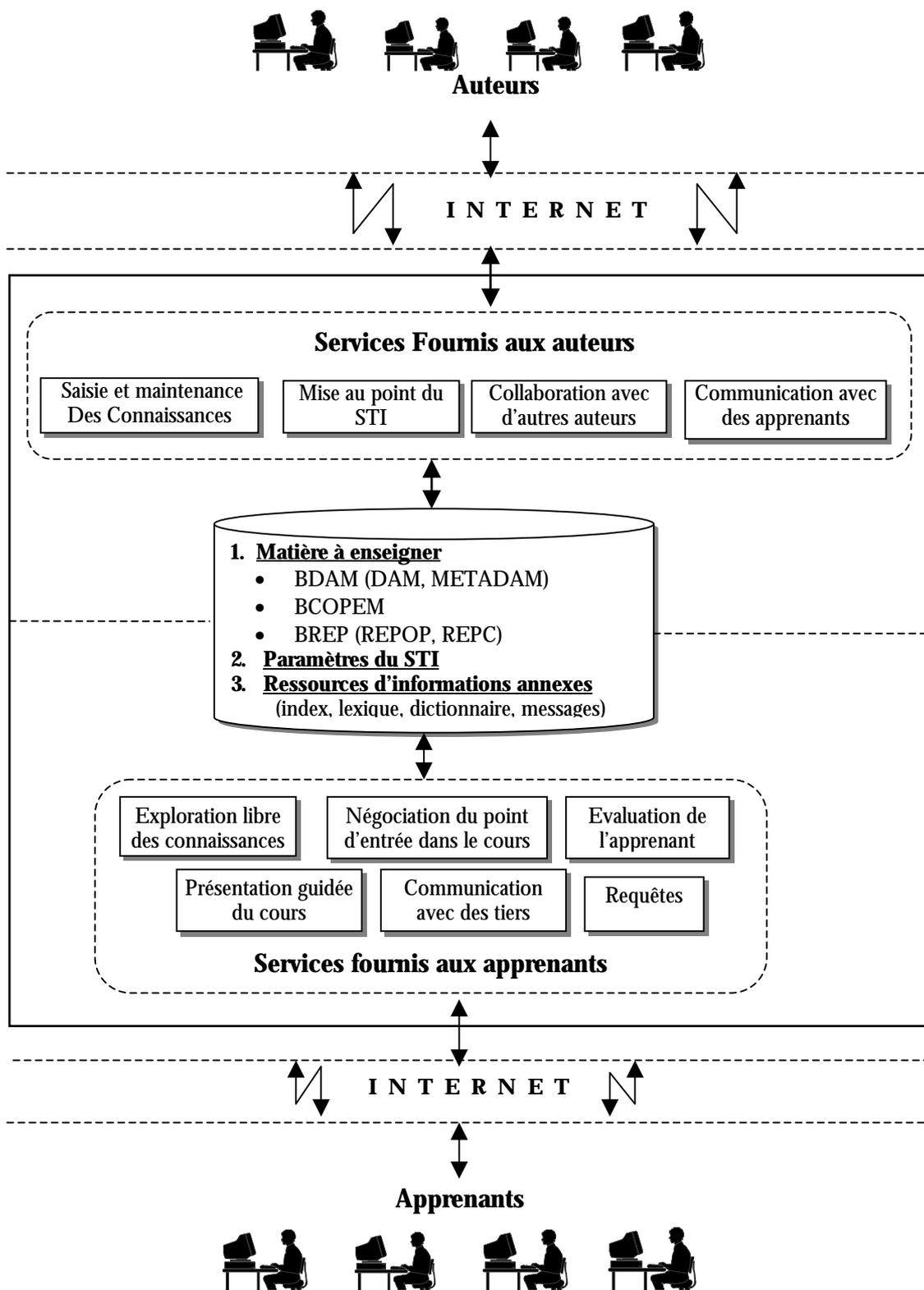


Figure 4.2. Modèle conceptuel du système auteur

4.5 Modèle de STI produit

Conformément aux résultats des recherches atteints dans les domaines de la psychologie cognitive et des STI, il s'avère qu'un système tuteur informatique est qualifié d'intelligent s'il répond à quatre critères essentiels : être expert dans son domaine (maîtriser la matière à enseigner), être pédagogue (maîtriser la manière d'enseigner), être psychologue (savoir adapter son cours en fonction des aptitudes de l'apprenant) et être ordonné et agréable dans sa présentation (avoir une bonne ergonomie). Ce sont ces quatre préoccupations sur lesquels est basé l'architecture logicielle d'un STI avec ses quatre fameux composants (module expert, module pédagogue, modèle de l'apprenant et interface-apprenant).

Un tuteur intelligent devrait pouvoir résoudre les mêmes problèmes que ceux posés à l'apprenant. Il pourrait ainsi suivre le raisonnement de l'apprenant et comprendre ses erreurs. Un des objectifs du développement des systèmes tuteurs intelligents, est de les rendre au moins aussi performants que les «tuteurs humains».

Dans le contexte du tutorat sur le web, il nous a fallu ajouter à ces préoccupations d'autres préoccupations inhérentes au contexte du réseau Internet, du world wide web (www), et des protocoles qui les régissent. La technologie des Systèmes Hypermédias Adaptatifs [Brusilovsky 01], que nous avons adopté pour la conception de notre système, est actuellement l'axe de recherche qui essaye de conjuguer les avantages de la technologie des STI avec ceux de la technologie des hypermédias et du web.

En conséquence, notre système tuteur est conçu sur la base de deux paradigmes : celui des hypermédias et celui des systèmes experts. L'analyse critique que nous avons réalisé sur ces deux paradigmes nous a révélé un certain nombre d'insuffisances pour chacun d'eux mais aussi des points forts.

- Un système expert est très limité dans son niveau d'expertise, long à développer, rigide et difficile à modifier, alors qu'un hypermédia est théoriquement illimité dans son expertise, développé rapidement et ses modifications et mises à jour sont faciles ;
- Un système hypermédia, au vu de sa liberté et sa flexibilité, est réputé par les deux problèmes de désorientation et de surcharge cognitive qui fait que l'utilisateur se perd souvent dans la chaîne de liens qui lui sont proposés et part souvent dans des liens inutiles en perdant de vue son objectif initial, alors qu'un système expert est réputé par le fait qu'il guide pertinemment l'utilisateur vers son objectif.

Ainsi, nous avons conjugué les bénéfices des deux paradigmes dans le but d'adapter le cours aux besoins et aptitudes de chaque apprenant. Ceci permettra par conséquent, à ce dernier, d'atteindre les objectifs d'apprentissage fixés par l'auteur du cours. Dans ce qui suit nous présentons l'approche conceptuelle adoptée puis la réalisation du système Hits

Pour structurer la matière à enseigner, nous utilisons une hiérarchie à trois niveaux d'objectifs pédagogiques [Hameline 90] et [Tagliante 91] : *les objectifs généraux*, *les objectifs spécifiques* et *les objectifs opérationnels*. Cette hiérarchie a permis de considérer trois niveaux d'abstraction du contenu : *les parties* (satisfaisant aux objectifs généraux), *les chapitres* (satisfaisant aux objectifs spécifiques) et *les uah* (unités d'apprentissage hypermédias) (satisfaisant aux objectifs opérationnels). Ces derniers sont les unités élémentaires de transfert évaluables. Deux modes d'apprentissage sont offerts aux apprenants : le *mode formation* (i.e. avec évaluation) et le *mode exploration libre*.

Notre système s'insère dans le courant des systèmes qui organisent le processus d'enseignement autour de composants hypermédias. La gestion des composants dans le canevas de STI, est assurée par un système multi-expert basé sur un ensemble de règles de production. Ces règles complètement paramétrables, dites règles maîtresses, décrivent les différents plans de tutorat relatifs aux différentes situations dans lesquelles peut se trouver l'apprenant. Elles constituent à cet effet une base de connaissances générique qu'il convient d'instancier pour chaque STI créé.

L'opération d'instanciation, produisant des règles instances, est effectuée automatiquement par le système en se basant sur des *paramètres* saisis impérativement par l'auteur. Ces paramètres du STI, représentés sous forme de prédicats, décrivent l'aspect quantitatif de la matière à enseigner (*nombre de parties*, *nombre de chapitres*, *nombre d'uah*, *nombre de questions*, *nombre d'exercices*, etc.).

Pour rester indépendantes de tout domaine, les règles maîtresses invoquent des structures abstraites appelées unités d'apprentissage hypermédias (uah). Ces uah étant dénuées de toute connaissance du domaine dans le canevas de STI prédéfini. Elles sont censées recevoir, par instanciation, toutes sortes de connaissances du domaine, sous toutes les formes de médias permises par les langages hypermédias (texte normal, texte formaté, image fixe, image animé, son, vidéo, activex, script, applet). En somme, en terme de curriculum [Lesgold 88], deux niveaux de connaissances sont donc utilisés :

1. Un niveau supérieur correspondant aux plans de tutorat — Ces derniers consistent en cinq paquets de règles qui invoquent des uah du niveau inférieur. Chaque paquet de règles possède une fonction précise, ces fonctions sont respectivement les suivantes :

- négociation avec l'apprenant du point d'entrée dans le cours ;
- déduction des acquis à l'issue de la phase de négociation ;
- planification d' enchaînement des uah ;
- recherche et affichage des uah ;
- évaluation de l'apprenant.

2. Un niveau inférieur correspondant à l'univers des uah — Cet univers consiste en un réseau de prérequis formé de six niveaux d'uah. Les quatre premiers niveaux correspondent à des uah de cours (sommaire, résumé de partie, résumé de chapitre, uah de cours) et les deux derniers sous-niveaux correspondent à des uah d'évaluation (questions théoriques et exercices). Selon la logique « dire, montrer, faire », les uah de cours présentent la théorie sur le thème à enseigner (dire) et les exemples permettent de (montrer) à l'apprenant comment appliquer la théorie sur des exemples concrets. Les uah d'évaluation quant à elles, permettent de mesurer l'atteinte des objectifs opérationnels par l'apprenant, et ce, en le poussant à (faire) soit même des exercices et applications de la théorie qui lui a été présentée dans l'uah de cours.

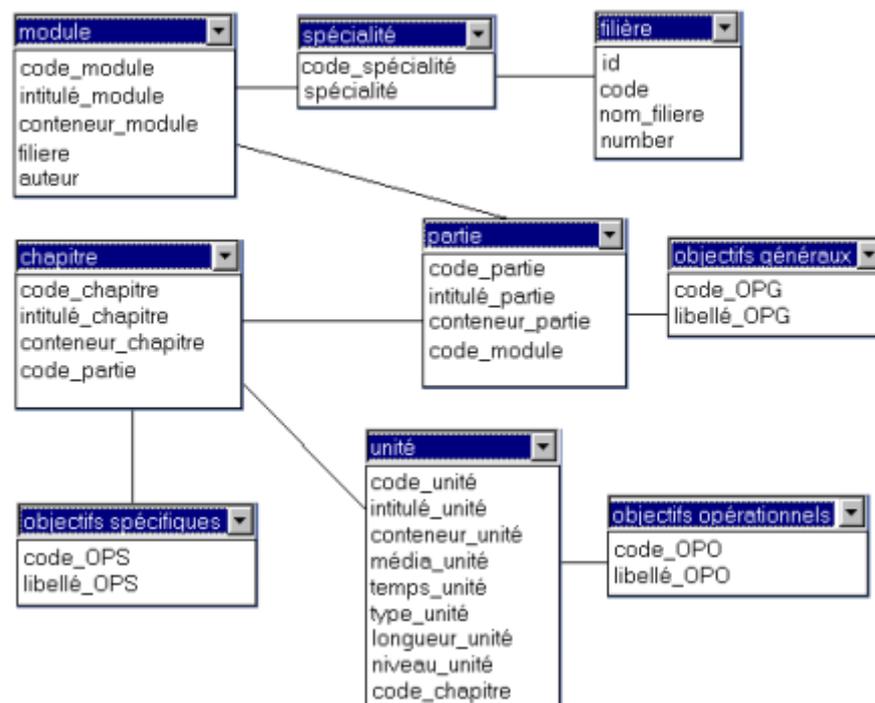


Figure 4.3. Une partie du modèle conceptuel des données de la plate-forme Ibn Sina

4.6 L'édition coopérative du STI

Littéralement, l'expression « travail coopératif » désigne un travail effectué en commun par plusieurs personnes aboutissant à une œuvre commune. Les personnes interagissent pour se donner ou accomplir un objectif commun, lequel implique une répartition des tâches (coordination) et une convergence des efforts (coopération) des membres du groupe. Nous précisons que l'expression « travail coopératif » est toujours orientée vers la réalisation d'un produit final tangible, alors que ce n'est pas toujours le cas pour « travail en équipe » qui peut être une finalité – d'apprentissage, par exemple – sans nécessairement donner lieu à un résultat tangible.

En vue d'améliorer la productivité dans le domaine des STI et permettre à une communauté plus large de s'impliquer, des systèmes auteurs (voir Chapitre 3) sont apparus et ont permis de développer plus ou moins aisément des STI. Certains de ces systèmes auteurs sont discutés dans [Murray 99], cependant, ils ont tous été conçus pour fonctionner en mono-usager.

De nos jours, nous pensons que la disponibilité de ces systèmes ne résout pas totalement le problème. En effet, avec la complexité et l'interdépendance des sciences actuelles, la difficulté de construction d'un STI par un seul auteur reste toujours à l'ordre du jour. Ainsi donc, en faisant abstraction des autres types de problèmes, même de point de vue acquisition de connaissances, cette construction nécessite souvent la coopération de divers experts du domaine en question. Etant évidemment géographiquement éloignés dans la plupart du temps, il est par conséquent nécessaire de mettre à leur disposition des supports coopératifs leur permettant de communiquer et coordonner leurs activités.

Aujourd'hui, grâce aux technologies de réseaux et aux collecticiels, des rencontres virtuelles à large échelle sont rendues possibles. Plusieurs travaux dans ce sens ont déjà porté sur des domaines tels que l'édition coopérative de documents [Decouchant 95] [Zidani 00], la conception coopérative d'objets, etc. Le point commun entre tous ces systèmes est qu'ils permettent à plusieurs participants de travailler ensemble de manière synchrone ou asynchrone pour réaliser une tâche commune.

D'autant plus que la technologie le permet donc et est expérimentée avec succès dans beaucoup de domaines, ceci nous a conduit à proposer que notre système auteur (spécialement son mode auteur) intègre l'aspect coopératif à travers un réseau informatique.

Notre système, offrant des fonctionnalités de collaboration aux auteurs, génère des STI hypermédias assurant l'enchaînement et la planification de curriculum. Les systèmes auteurs de cette catégorie structurent généralement la matière à enseigner sous forme d'un réseau d'unités d'apprentissage hypermédia (uah) et où chaque uah possède certains objectifs pédagogiques.

Les uah à présenter à l'apprenant sont alors déterminées dynamiquement en se basant sur les performances de ce dernier, les objectifs pédagogiques de la leçon et les relations qui existent entre les différentes uah. Notre modèle de système auteur est l'un des modèles de systèmes qui, dans le souci de faciliter la tâche aux auteurs, ne demande que d'instancier les uah. L'auteur devra ensuite introduire les réseaux de prérequis et les paramètres nécessaires au fonctionnement des trois autres composants.

L'architecture logicielle du mode auteur permet aux auteurs de coopérer pour la production du STI hypermédia. Toutefois, il est à remarquer que la partie logicielle ne constitue pas tout dans les collecticiels. La prise en compte des facteurs humains impliqués par les activités de groupe constituent également une condition capitale quant au succès de ces collecticiels [Greenberg & al 92]. Ainsi, pour éviter les conflits inhérents à la nature humaine, nous avons proposé une organisation qui permet de mener la conduite du projet de construction collective du STI de manière rationnelle et optimale. Nous définissons alors quatre rôles à travers lesquels les participants peuvent intervenir au cours du processus de construction du STI : auteur principal (chef de projet), coauteur superviseur (supervisant un sous-groupe d'auteurs), coauteur constructeur et coauteur lecteur/commentateur.

Afin de bien comprendre le fonctionnement de notre système par la suite (Chapitre 6), nous présentons ci-après les fonctionnalités essentielles du modèle client-serveur selon lequel fonctionne le world wide web.

4.7 Architecture client serveur du web

Selon Bernard Martin [Martin 01], le web est construit autour d'une architecture client-serveur (**Figure 4.4**). Le client envoie des messages aux serveurs Web qui sont appelés démons HTTPD (HTTP Daemon). Les serveurs Web sont des processus dont la fonction essentielle est de répondre à des requêtes émises par les clients exécutant un navigateur (browser). Ces requêtes peuvent être aussi bien la demande de transfert d'un fichier que le résultat de l'exécution d'un programme sur la machine serveur (par exemple, l'interrogation d'une base de données).

Un client web est un processus qui exécute un programme permettant à un utilisateur de soumettre des requêtes à un serveur web et d'en visualiser le résultat. Le protocole de communication ne maintient pas de session permanente entre le client et le serveur web. Le dialogue entre un client et un serveur web est schématiquement le suivant :

- établissement de la connexion et envoi de la requête ;
- le serveur fournit ou non l'information demandée ;
- fermeture de la connexion.

Le protocole de communication utilisé entre un client web et un serveur web est appelé HTTP. Le protocole HTTP est un protocole question-réponse et sans état, construit sur TCP/IP. Le dialogue dans le cas d'une requête pour un document HTML se déroule schématiquement de la façon suivante (**Figure 4.4**) :

- le client établit une connexion TCP sur le port du serveur qui accepte la connexion ;
- le client émet sa requête vers le serveur qui se compose de la méthode GET, de l'URL du document demandé, de la version du protocole utilisé (et éventuellement d'un message de type MIME contenant des modificateurs pour la requête et des informations du client) ;
- le client donne la liste des types MIME qu'il peut accepter ;
- le serveur répond avec une ligne d'état, incluant la version du protocole et un code d'état suivi d'un message de type MIME contenant des informations du serveur et le corps du document HTML demandé ;
- le serveur coupe la connexion qui matérialise ainsi la fin du document demandé.



Figure 4.4. Dialogue entre un client et un serveur selon le protocole HTTP

Le Common Gateway Interface (CGI) [Maurice 00] est le mécanisme qui permet à un client de faire exécuter un programme sur le serveur. Ces programmes sont appelés scripts de passerelles, scripts CGI ou scripts CGI-BIN et peuvent être des scripts ou des programmes exécutables. C'est le mécanisme de base pour les documents interactifs ou dynamiques, à l'opposé des pages statiques HTML. Ils peuvent par exemple permettre de traiter les données enregistrées dans un formulaire, de se connecter à une base de données pour des extractions ou des mises à jour, de lancer une autre application. Pour utiliser l'interface CGI, il faut contacter l'administrateur du serveur Web (Webmaster). L'exécution des scripts CGI doit être explicitement autorisée.

Afin d'améliorer les performances des communications entre le client et le serveur web, sont utilisées des techniques de cache pour les clients (en particulier pour les navigateurs) et de proxy, ou mandataire, pour les serveurs. Le mandataire joue le rôle d'un intermédiaire entre le client et le serveur web. Le client adresse alors sa requête au mandataire qui peut soit la satisfaire si le document est dans son cache, soit la transmettre au serveur si ce n'est pas le cas. Le mandataire agit pour un ensemble de clients, généralement connectés sur le même réseau. Les navigateurs ont une option de configuration qui permet de spécifier leur mandataire par son adresse TCP/IP [Pujole 97].

4.7.1 Serveurs

Les premiers logiciels serveurs du web furent développés par le CERN et le NCSA. Il en existe maintenant un grand nombre, dont plus de 80 %, sont des serveurs sous UNIX. On note depuis 1999 la montée en puissance du serveur Apache qui représente plus de 60 % des serveurs web en 2000. Nous rappelons que le client web peut aussi bien demander l'accès à une page HTML que l'exécution d'un programme sur le serveur en utilisant la passerelle CGI, qui fait partie intégrante du serveur ou l'exécution de scripts comme asp ou php par exemple.

Apache (« A PAtCHy server ») est un serveur web pour le domaine public, qui utilise la même syntaxe que NCSA 1.3 pour les fichiers de configuration. Le serveur Apache introduit un API pour l'ajout de nouvelles fonctions au serveur, la personnalisation des messages d'erreurs et la gestion automatique du multilingage en traitant les entêtes http.

4.7.2 Navigateurs

Les premiers navigateurs furent ceux développés au CERN. L'importance du web a réellement commencé avec la mise sur le réseau Internet du navigateur Mosaic pour un grand nombre de plates-formes. Le marché actuel des navigateurs html est largement dominé par MS Internet

Explorer et Mozilla Firefox, dont les dernières versions permettent la visualisation de fichiers XML [Michard 99] avec un support limité des feuilles de styles. Mais, citons aussi le navigateur Amaya du consortium W3C qui est un logiciel libre.

Les navigateurs offrent fonctions de navigation qui permettent aux utilisateurs de suivre les hyperliens dans les pages html et, par les commandes ou boutons du panneau de contrôle, d'aller à la page précédente (Back), à la page suivante (Forward), à une des pages mémorisées dans l'historique (Go), à une page définie par un URL explicitement affiché («file/open location») par l'utilisateur, à une page définie par un nom de fichier local («file/open file»).

La commande «actualiser» permet de rafraîchir la page courante en forçant son rechargement. Les bookmarks permettent à l'utilisateur de constituer une liste structurée, classée par thèmes par exemple, de sites identifiés par leur URL.

Les premiers navigateurs tels que Mosaic du NCSA et les premières versions de Netscape étaient des logiciels monolithiques pour lesquels l'adjonction de nouveaux formats et de nouveaux protocoles étaient impossibles. L'ajout de nouvelles fonctions a été rendu possible par différents mécanismes :

- **Connexion de visualisateurs externes**

Quand un navigateur reçoit des données d'un format qu'il ne sait pas traiter, il appelle un programme externe pour afficher les données. Des données en format RTF de Microsoft, par exemple, conduiront le navigateur à appeler le logiciel de traitement de texte Word. L'utilisateur peut décrire dans le fichier de configuration de son navigateur les différentes applications à associer aux différents types de données qu'il souhaite pouvoir traiter. Le format des types de données est décrit en utilisant un type MIME.

- **Interface client à client**

Cette interface, appelée CCI (Client-to-Client Interface), définit un API que les applications externes pourront utiliser pour communiquer avec le navigateur. Elle permet à une application de faire réaliser un traitement par le navigateur (ouvrir une nouvelle fenêtre, charger un document défini par un URL donné...) ou de signaler l'occurrence d'un événement.

- **API propriétaires**

Les exemples en sont les plug-Ins de Netscape et ActiveX d'Internet Explorer qui permettent d'étendre les fonctions offertes initialement par le navigateur.

4.8 Architecture générale de l'environnement TalHits

Le système TalHits est structuré selon une architecture client-serveur centralisée. Il est composé de trois composants en interaction répartis entre les postes client et serveur (**Figure 4.6**). Plus de détails concernant cette architecture seront données dans le chapitre suivant.

- Du côté serveur : Nous trouvons, outre le serveur web, les deux applications qui constituent les deux modes du système auteur à savoir le mode auteur (CamHits) et le mode apprenant (Hits).
 1. CAMHITS (Cooperative Authoring Module for Hypermedia Intelligent Tutoring System) qui offre aux auteurs des outils de saisie des connaissances du domaine, des outils de maintenance du STI, des outils de coopération, etc.
 2. HITS (Hypermedia Intelligent Tutoring System) qui fournit à l'apprenant des outils lui permettant de mener sa session d'apprentissage, tels que : exploration libre des connaissances, négociation du point d'entrée dans le cours, collaboration avec des tiers, etc.
- Du côté client : nous trouvons un navigateur Web (MS Internet Explorer ou Mozilla Firefox) mettant à la disposition de ses usagers (apprenants et auteurs) les fonctionnalités essentielles de navigation à travers le web.

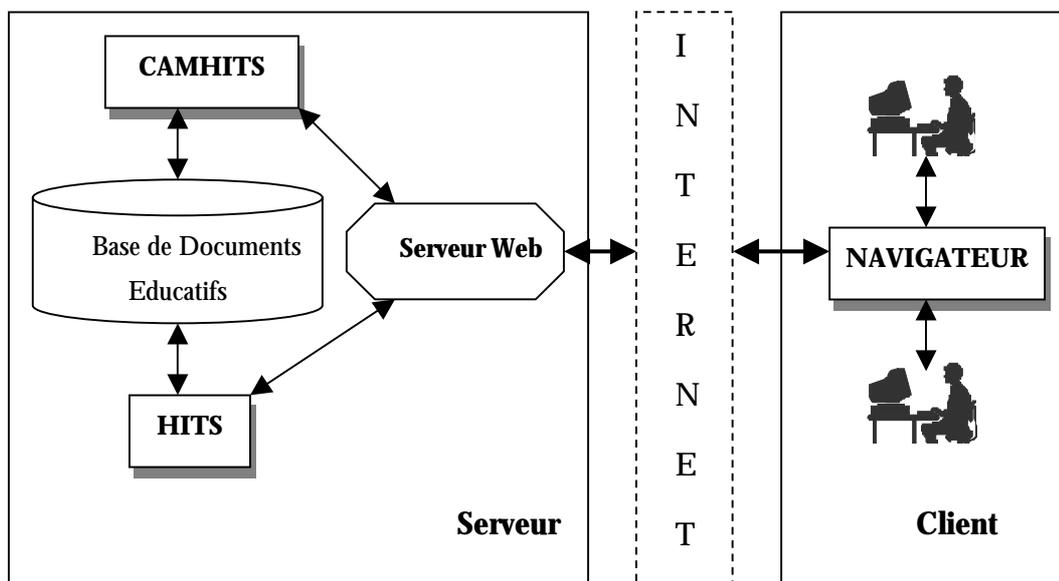


Figure 4.5. Architecture générale du système auteur TalHits

4.8.1 Le mode apprenant Hits

L'architecture logicielle du STI hypermédia comporte au moins tous les composants constituant un STI traditionnel, elle comporte :

- Un module d'exploration libre qui permet à l'apprenant de naviguer librement à travers les différentes unités d'apprentissage hypermédiées (uah).
- Trois modules représentant le mode d'apprentissage formation avec évaluation :
 1. Un module expert du domaine qui permet, en utilisant les règles instances de recherche des uah, de chercher et d'afficher l'uah sollicitée par le système à un moment donné.
 2. Un module tutoriel qui permet de négocier avec l'apprenant le point d'entrée dans le cours (concept ou objectif pédagogique ou uah) et de planifier l'enchaînement des uah sur la base du résultat de cette négociation. Ces deux fonctions étant assurées par deux sous-modules : le négociateur utilisant les règles instances de négociation et le planificateur utilisant les règles instances de planification.
 3. Un module de diagnostic de l'apprenant qui permet d'évaluer l'utilisateur et d'assurer la maintenance d'un modèle-apprenant de type overlay. Ce module comporte à son tour trois sous-modules : un évaluateur utilisant les règles instances d'évaluation, un déducteur des acquis utilisant les règles instances de détermination des acquis et un gestionnaire du modèle de l'apprenant manipulant le contenu de ce dernier.
 4. Un module coordonnateur qui permet d'une part de communiquer avec l'apprenant et d'autre part, de coordonner la communication entre les trois modules : expert, tutoriel et celui du diagnostic de l'apprenant. La communication entre ces trois modules étant assurée par envoi de message.
 5. un module de collaboration permettant aux apprenants de communiquer entre eux, ainsi qu'avec leurs enseignants dans le but de collaborer entre eux pendant les phases d'apprentissages.

4.8.2 Le mode auteur CamHits

L'éditeur coopératif de connaissances est organisé selon une architecture client/serveur centralisée [Orfali & al 97]. Par conséquent, toutes les communications transitent automatiquement par le site central (ou serveur).

L'éditeur coopératif présente aux auteurs tous les outils nécessaires à l'élaboration collaborative d'un STI Hypermédia. Du point de vue d'un auteur, construire un STI avec CamHits consiste à introduire, via cet éditeur, un ensemble d'objets qui seront manipulés par le mode apprenant. Ces objets sont constitués de la matière à enseigner sous formes d'uah, des réseaux de prérequis sous forme de graphes orientés, des paramètres du STI sous forme de prédicats, et des connaissances pédagogiques sous forme de règles de production.

La coopération dans CamHits est introduite au niveau de l'édition de la matière à enseigner et au niveau de l'édition du réseau de prérequis. Ces deux composants sont bien structurés : la matière étant hiérarchisée en parties, chapitres et uah, et le réseau de prérequis en sous-réseaux (prérequis-parties, prérequis-chapitres et prérequis-uah, prérequis-concepts, prérequis-objectifs). Cette structure s'adapte bien pour la fragmentation de ces deux composants et constitue de ce fait la base de notre approche d'édition coopérative comme dans JamEdit [Zidani & al 00].

Le principe d'édition coopérative que nous avons exploité repose en fait sur les deux concepts clés utilisés dans la plupart des éditeurs coopératifs : la fragmentation et l'attribution de rôles d'édition sur les différents fragments. Nous avons défini quatre rôles pour les auteurs dans CamHits : auteur principal, superviseur, constructeur et lecteur/commentateur.

4.8.3 Le navigateur

L'activité de navigation fait partie intégrante du processus d'apprentissage. Le processus de navigation se trouve généralement confronté à un problème majeur. Les apprenants en situation d'exploration libre de contenus sur le web trouvent la plupart du temps des difficultés à construire un schéma mental de navigation et posent des problèmes de désorientation et de surcharge cognitive. Les systèmes d'aide à la navigation proposés par les différents logiciels classiques ne répondent pas aux besoins des usagers. Les navigateurs les plus utilisés tels que Microsoft Internet Explorer et Mozilla FireFox par exemple proposent des fonctionnalités telles que les signets et les favoris. Cependant, elles sont insuffisantes car l'utilisateur ne dispose pas dans ces navigateurs d'un outil de personnalisation des documents [Zeiliger & al 97].

Ces problèmes ont motivé notre intérêt à développer un navigateur basé sur Internet Explorer [Ouchen & al 02] doté de fonctionnalités qui permettent d'imiter les activités qu'un étudiant a généralement l'habitude de faire lors de la lecture d'un document papier, à savoir : le marquage et l'annotation [Denouet & Vignolet 00a] [Denouet & Vignolet 00b] .

Annoter permet de mettre en valeur des passages dans le document qui sont jugés importants par l'utilisateur, c'est donc un outil de personnalisation. Les signets représentent un document uniquement par son URL et son titre. Les annotations quant à elles permettent une représentation plus riche du document. Lors d'un nouvel accès, les passages annotés aident l'apprenant à se repérer dans le document. Un outil d'annotation devrait permettre à la manière des signets la création d'annotations stockées sur la machine de l'usager et laisser à sa volonté de partager ces annotations avec d'autres utilisateurs.

4.9 Conclusion

Nous venons de décrire brièvement notre modèle de système auteur TalHits. Nous avons décrit, en premier lieu, la méthode pédagogique sur laquelle repose ce système, à savoir la pédagogie par objectifs. Nous avons retenu essentiellement trois niveaux hiérarchiques d'objectifs: les objectifs généraux, les objectifs spécifiques et les objectifs opérationnels.

Sur la base de cette pédagogie, nous avons défini, par la suite, les besoins fonctionnels du système pour montrer les services qui sont offerts aux deux utilisateurs potentiels du système, les auteurs d'un côté et les apprenants de l'autre côté.

Du côté de l'apprenant particulièrement, nous avons proposé deux modes d'apprentissage, le « mode exploration » qui consiste en une exploration libre de la matière à enseigner et le « mode formation » qui consiste en une présentation guidée de la matière, sur la base d'un diagnostic continu de l'apprenant.

Aussi, des échanges de message avec des tiers via des outils de communication sera d'un grand intérêt pour la progression de l'apprentissage de l'apprenant.

Du côté de l'auteur, nous avons proposé des outils coopératifs de création de tutoriels qui soient faciles à utiliser d'une part, et qui d'autre part, permettent de développer des STI conformes aux principes que nous avons retenus de la pédagogie par objectifs. Nous avons, par exemple, proposé pour la structuration du cours à enseigner, une organisation qui ne diffère pas beaucoup de celle adoptée dans l'enseignement traditionnel, une organisation qui consiste en trois niveaux hiérarchiques : les parties (satisfaisants aux objectifs généraux), les chapitres (satisfaisant aux objectifs spécifiques) et les unités d'apprentissage hypermédias (satisfaisant aux objectifs opérationnels).

L'expression des besoins a été le point de départ pour une modélisation du système. L'approche que nous avons adoptée consiste à conjuguer les deux technologies de système expert et d'hypermédia pour mener la tâche d'enseignement/apprentissage. Afin de conserver la structure des tuteurs intelligents traditionnels, nous avons divisé la base de connaissances du système expert en cinq paquets de règles que nous avons répartis entre les trois modules (expert, pédagogue et modèle de l'apprenant). Le système expert est alors conçu autour de cinq systèmes experts qui coopèrent entre eux, par envoi de message à travers un module coordonnateur.. Chaque système expert possède sa propre base de connaissance et son propre moteur d'inférence.

Dans le chapitre suivant, nous détaillerons l'architecture logicielle ainsi que la représentation des connaissances et les mécanismes de raisonnement adoptés dans le mode apprenant Hits.



Chapitre 5

Mode apprenant – architecture logicielle

5.1 Introduction

Dans le chapitre précédent, nous avons présenté une vue générale de notre système auteur (TalHits) ainsi que le contexte dans lequel il s'insère. Sur la base de la pédagogie par objectifs, nous avons établi un ensemble de besoins de ce système par rapport à l'environnement global où il devrait s'intégrer. Une ébauche du système répondant à ces besoins a été alors réalisée par la suite. Deux grands composants ont alors émergé dans l'architecture globale du système : le mode auteur CamHits [Talhi & al 06] et le mode apprenant Hits [Talhi & al 05].

Dans ce chapitre, nous décrivons plus en détail le mode apprenant Hits du point de vue architecture logicielle de tous ses composants ainsi que la manière d'interaction entre eux, d'un côté, et avec les apprenants d'un autre côté. Nous décrivons particulièrement les différents types de connaissances intégrées dans le système : les connaissances du domaine, les connaissances de l'apprenant et les connaissances tutorielles. Les méthodes de raisonnement utilisées par les différents modules de Hits, sont décrites au fur et à mesure que les structures de ces connaissances sont définies.

5.2 Architecture du STI Hypermédia (Hits)

Nous rappelons que dans le but de faciliter la création de tutoriels par des enseignants, nous avons proposé de définir un canevas de STI une fois pour toute, que les enseignants ne font qu'instancier à travers le mode auteur CamHits, puis de fournir tous les paramètres nécessaires à son fonctionnement.

La gestion des uah n'est plus assurée par un algorithme rigide comme dans les didacticiels classiques, mais par un système multi-expert basé sur un ensemble de cinq paquets de règles maîtresses. Cet ensemble de règles constitue une base de connaissances générique prédéfinie qu'il convient d'instancier pour chaque STI créé, en se basant sur les paramètres du STI saisis par l'auteur.

Les uah invoquées par les règles maîtresses sont dénuées de toutes connaissances du domaine dans le canevas de STI prédéfini. Elles sont sensées recevoir, par instanciation, toute sorte de connaissances du domaine sous toutes les formes de médias autorisées par les hypermédias (texte, image fixe, image animé, son, vidéo, etc.) (**Figure 5.1**).

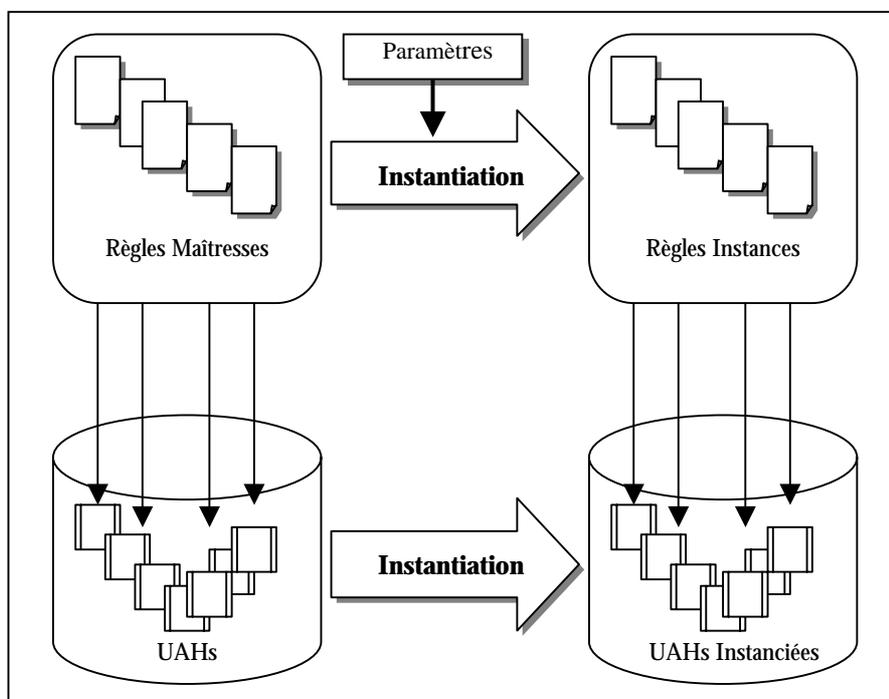


Figure 5.1. Création d'un STI par TalHits

Quant à l'architecture de Hits, elle comporte plusieurs modules en interaction (**Figure 5.2**) :

- Un « **module d'exploration libre** » qui permet à l'apprenant de naviguer librement à travers les uah et les ressources d'information annexes constituant le STI ;
- Trois modules représentant le mode d'apprentissage « Formation » :
 - Un « **module expert du domaine** » qui permet, de rechercher et d'afficher l'uah sollicitée par le système à un moment donné ;
 - Un « **module tutoriel** » qui permet de négocier avec l'apprenant et de planifier l'enchaînement des uah sur la base du résultat de cette négociation. Ces deux fonctions étant assurées par deux sous-modules : le « négociateur » et le « planificateur » ;

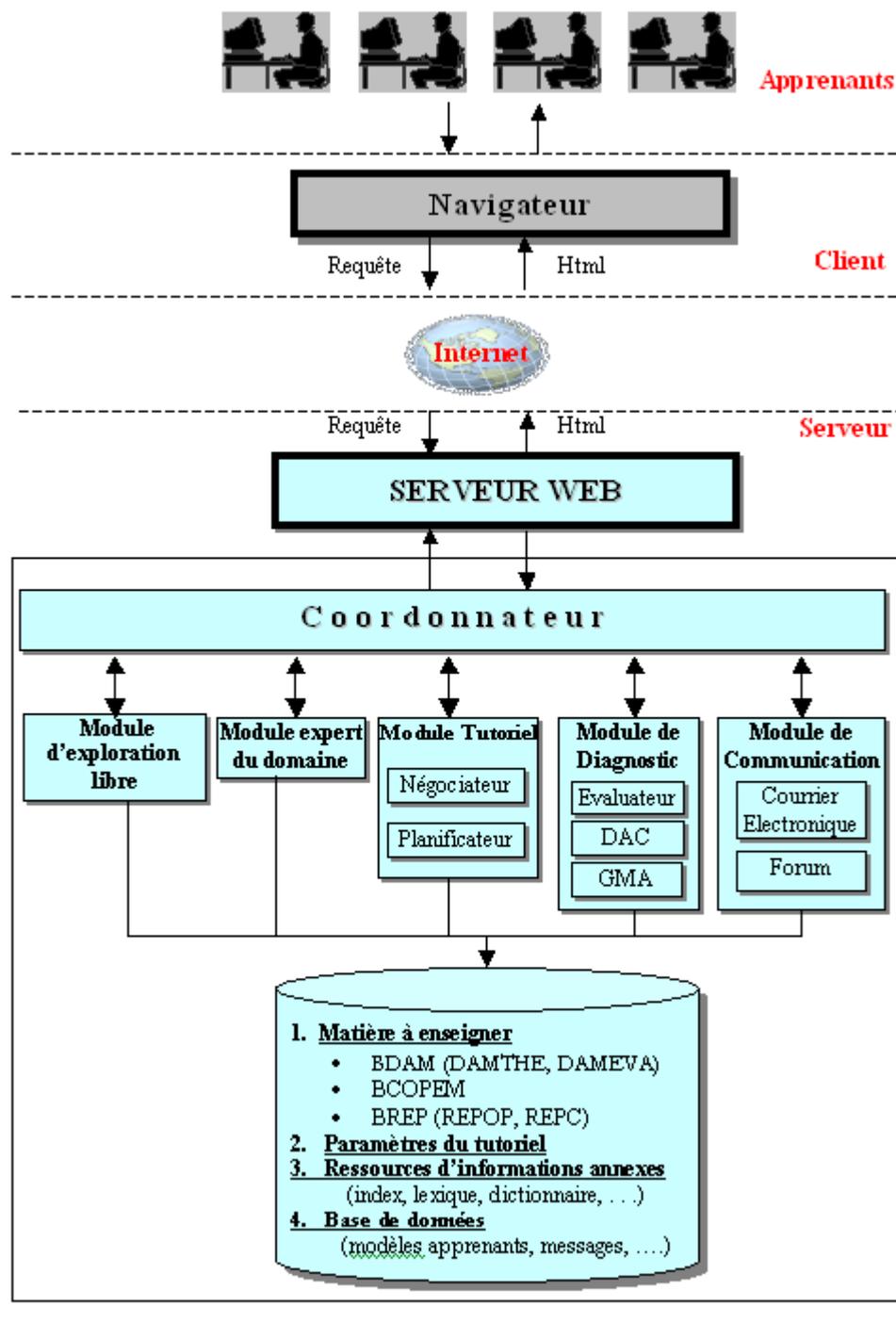


Figure 5.2. Architecture logicielle de Hits

- Un « **module de diagnostic de l'apprenant** » qui permet d'évaluer l'apprenant et d'assurer la maintenance du modèle de ce dernier. Ce module comporte à son tour trois sous-modules : un « évaluateur », un « déducteur des acquis » (DAC) et un « gestionnaire du modèle de l'apprenant » (GMA) manipulant le contenu de ce dernier ;

- Un « **module de communication** » qui permet aux apprenants de communiquer d'une manière asynchrone avec des tiers (enseignants ou autres apprenants).
- Un « **module coordonnateur** » qui permet d'une part de communiquer avec l'apprenant (via un serveur web), et d'autre part, de coordonner la communication entre tous les autres modules. La communication entre ces modules étant assurée par « envoi de message ».

5.2.1 Coopération inter-composants

Rappelons que le système Hits offre deux modes d'apprentissage à l'apprenant : le « mode exploration libre » et le « mode formation guidée ». Alors que le premier, assuré par le module (« module d'exploration libre ») est simple, le second, par contre, est plus complexe. Il est assuré par une coopération entre cinq systèmes experts (l'expert du domaine, le négociateur, le planificateur, l'évaluateur et le déducteur des acquis). Chaque système expert consiste en un moteur d'inférence, une base de règles spécialisée et une base de faits. La coopération entre les systèmes experts est effectuée par « envoi de message » via le module coordonnateur (**Figure 5.3**). Un message envoyé par un « expéditeur » à un « destinataire » est représenté par un « fait » déduit par l'expéditeur qui bloque la suite du raisonnement dans ce dernier et qui est censé déclencher une règle dans le destinataire. Par exemple, à un moment donné d'une session, le planificateur déduit le fait « présenter-uah(2,3,2) ». Etant donné que le planificateur n'est pas spécialisé dans l'affichage des uah, ce « fait » est alors envoyé au coordonnateur qui, à son tour, l'achemine vers l'expert du domaine qui contient certainement une règle de la forme :

***Si** présenter-uah(2,3,2) **alors** afficher-uah(2,3,2) **et** présenté-uah (2,3,2)*

Le message envoyé déclenche donc cette règle, ce qui aura pour conséquence l'affichage de l'uah(2,3,2) et le renvoi du message « *présenté-uah(2,3,2)* » au coordonnateur qui choisira à son tour le prochain destinataire à invoquer.

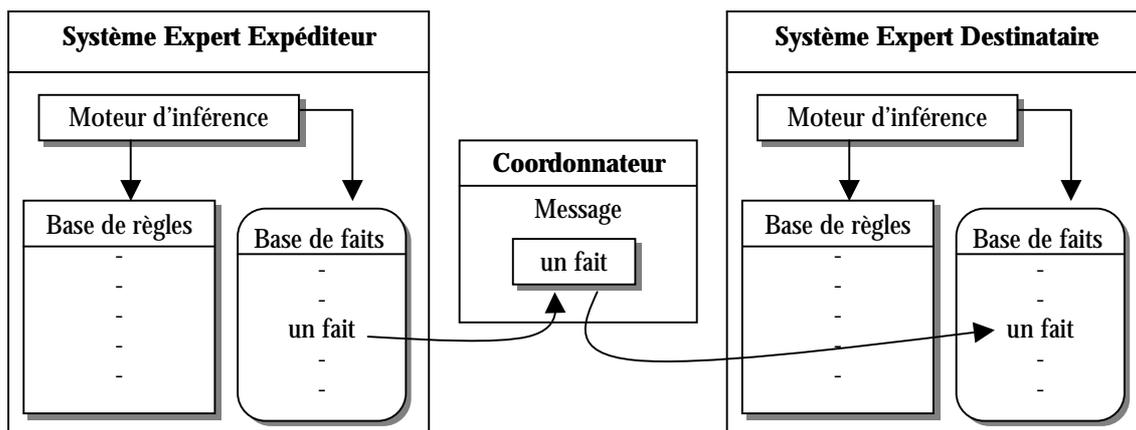


Figure 5.3. Coopération par envoi de message entre les composants

A l'instar de la plupart des systèmes tuteurs intelligents, le système Hits comprend trois sortes de connaissances : les connaissances relatives au domaine d'apprentissage (quoi ?), les connaissances relatives à l'apprenant (à qui ?) et les connaissances tutorielles chargées de la gestion des sessions d'apprentissage (comment ?). Représenter les connaissances en informatique consiste à trouver des structures de données appropriées au stockage et à la manipulation des informations relatives à une matière donnée. C'est ce que nous allons décrire dans les sections suivantes.

5.2.2 Les connaissances relatives au domaine d'apprentissage

La constatation faite pour la plupart des STI dédiés à des domaines spécifiques est la non réutilisabilité des structures de connaissances pour d'autres domaines. Pour essayer de répondre à ce besoin de réutilisation, le canevas de STI que nous proposons doit, dans la mesure du possible, rester indépendant du domaine d'application. Pour ce faire, il exploite une représentation abstraite de ce domaine. Autrement dit, il propose un modèle de connaissances qui, pour obtenir un STI dédié à un domaine particulier, doit être instantié sur les connaissances de ce dernier.

Le type de représentation des connaissances du domaine dépend de la nature des savoirs et/ou savoir-faire dans ce domaine. Deux approches essentielles sont considérées pour faire face à cette problématique. La première concerne des représentations assez fines des connaissances qui s'adaptent bien à des apprentissages de savoir-faire et de résolution de problèmes par exemple, dans un champs de compétences bien ciblé. La deuxième approche concerne des

connaissances plus macroscopiques dans le cadre de gestion de cursus individualisé d'apprentissage dans des univers de composants pédagogiques (ou frames).

Notre modèle se situe à mi-chemin entre ces deux approches, empruntant à chacune d'entre-elles des éléments du possible. Nous rappelons que le module d'enseignement est constitué de deux composants : la matière théorique formée des documents d'apprentissage THEoriques (DAMTHE) et la matière d'évaluation formée des documents d'EVALuation (DAMEVA). Chaque document d'apprentissage que ce soit dans DAMTHE ou dans DAMEVA est conçu sous forme d'une page hypermédia et doit satisfaire un ou plusieurs objectifs pédagogiques opérationnels fixés par l'auteur.

Pour un module d'enseignement donné du cursus, la matière théorique d'apprentissage (DAMTHE) est structurée en trois niveaux hiérarchiques :

- les *parties* (satisfaisants aux objectifs généraux) et comportant un ou plusieurs chapitres, un module devant comporter au moins une partie;
- les *chapitres* (satisfaisant aux objectifs spécifiques) et comportant une ou plusieurs unités d'apprentissage hypermédiées (uah), une partie devant comporter au moins un chapitre ;
- les *uah* (satisfaisant aux objectifs opérationnels) représentant les unités élémentaires d'apprentissage hypermédiées qui peuvent être évaluées, un chapitre devant comporter au moins une uah.

Quant à la matière d'évaluation (DAMEVA), elle est liée à la matière théorique d'apprentissage et permet au système d'adapter le tutorat au niveau de progression de l'apprenant, elle contient des uahs d'évaluation réparties à travers les entités suivantes :

- *un pretest-module* : permettant une *évaluation pronostique* du module, s'il existe il comporte au moins une épreuve ;
- *des tests-uahs* : permettant une *évaluation formative* des unités d'apprentissage hypermédiées théoriques, s'ils existent ils comportent chacun au moins une question et/ou un exercice;
- *des posttests-chapitres* : permettant une *évaluation inventaire* des chapitres, s'ils existent ils comportent chacun au moins une épreuve ;
- *des posttests-parties* : permettant une *évaluation inventaire* des parties, s'ils existent ils comportent chacun au moins une épreuve ;
- *et un posttest-module* : permettant une *évaluation inventaire* du module, s'il existe il comporte au moins une épreuve.

Avant de décrire le modèle conceptuel du module d'enseignement, nous allons d'abord décrire tous ses composants de base à savoir : les uahs, les objectifs pédagogiques, les concepts. Et les paramètres quantitatifs.

5.2.2.1 Les unités d'apprentissage hypermédias du module d'enseignement

Une unité d'apprentissage hypermédia (uah) est une structure exécutable par l'apprenant. C'est une petite unité de contenu centrée sur les objectifs à atteindre pour apprendre un certain nombre de concepts donnés. Une uah peut être constituée de plusieurs activités (qui aident à atteindre les objectifs assignés à l'uah), on peut citer des problèmes à résoudre, des transmissions de connaissances, des simulations, etc. Sur le plan pratique, une uah est donc constituée d'un ensemble d'éléments tels que des textes, des images, des animations, des vidéos et des programmes interactifs par exemple.

Nous espérons que l'exécution d'une uah par l'apprenant, la mise en application des activités qu'elle lui destine et son interaction avec elle, favorisent l'atteinte de ou des objectifs pédagogiques pour lesquels elle a été conçue et par là, l'apprentissage des concepts sous jacents du domaine.

Pour faire un rapprochement avec ce qui existe dans la littérature, nous pouvons apparenter notre concept d'uah avec celui « d'objet pédagogique » (Learning Object) [Bourda 01]. Les objets pédagogiques sont depuis déjà un certain temps au coeur de nombreux travaux tant dans les institutions d'enseignement, les sociétés de formation que dans les organismes de standardisation. Ces travaux étant assez coûteux en temps et visant à permettre l'échange et la réutilisation des objets pédagogiques.

Le groupe Ltsc de l'IEEE, (Institute of Electrical and Electronics Engineers) définit un « objet pédagogique » comme étant « une entité numérique ou non, qui peut être utilisée, réutilisée ou référencée dans toute activité liée à l'enseignement ou à l'apprentissage » [Bourda 01]. On entend ici par «activité», les systèmes d'enseignement assistés par ordinateur, les environnements interactifs d'apprentissage, les systèmes tuteurs intelligents, les systèmes hypermédias adaptatifs et les environnements d'apprentissage collaboratif (**Figure 5.4**).



Figure 5.4. Spécification 1484. 12. 1. du groupe Ltsc de l'IEEE

Un « cours » de formation à distance dans notre contexte est réalisé par agencement cohérent de petites activités d'apprentissage, par rapport à des objectifs à atteindre. Ces petites activités doivent être précises et imbriquées les unes par rapport aux autres, sur lesquelles il est possible de construire sans cesse de nouvelles connaissances. Cela revient à dire que l'apprentissage se fait grâce à un ensemble de « petits cours » imbriqués formant un tout [Behaz & Djoudi 05].

Cette approche est souvent appelée la « granulation » des apprentissages. Elle vise à segmenter les apprentissages à acquérir en petites unités cohérentes (éléments textuels, graphiques, des images, des sons et des vidéos) qui, une fois réorganisées, formeront un tout cohérent plus structurant que l'ensemble des unités prises individuellement. La difficulté est comment peut-on définir le niveau adéquat de granularité de ces unités élémentaires ?

S'il est clair que plus un objet pédagogique est petit, plus il est réutilisable dans des contextes d'apprentissage différents, il est très probable que définir des grains trop petits risque de leur ôter tout sens [Bourda 01].

Nous proposons donc que nos uahs soient de granularité moyenne (pas trop grande et pas trop petite) afin qu’elles puissent constituer une source effective aux diverses activités proposées aux apprenants. Nous affirmons que si le contenu du module est fragmenté en petites unités traitant un à deux concepts au maximum à la fois, ces unités seront plus partageables entre les acteurs du système, donc plus réutilisables ce qui va augmenter et favoriser énormément l’adaptation aux apprenants.

D’autres travaux menés au sein de notre groupe et qui ont pour objectif de faire de l’adaptation dynamique en apprentissage, s’occupent de décomposer encore nos uahs en plusieurs fragments plus fins de contenus appelées : unités d’apprentissage élémentaire (UPE) [Behaz & Djoudi 05]. Ces UPE, spécifiées dans le modèle Modap des deux auteurs [Behaz & Djoudi 04], peuvent être de type : Introduction, Théorème, Formule, Illustration, Définition, Conclusion, Exemple, Explication, etc. (**Fig. 5.5**).

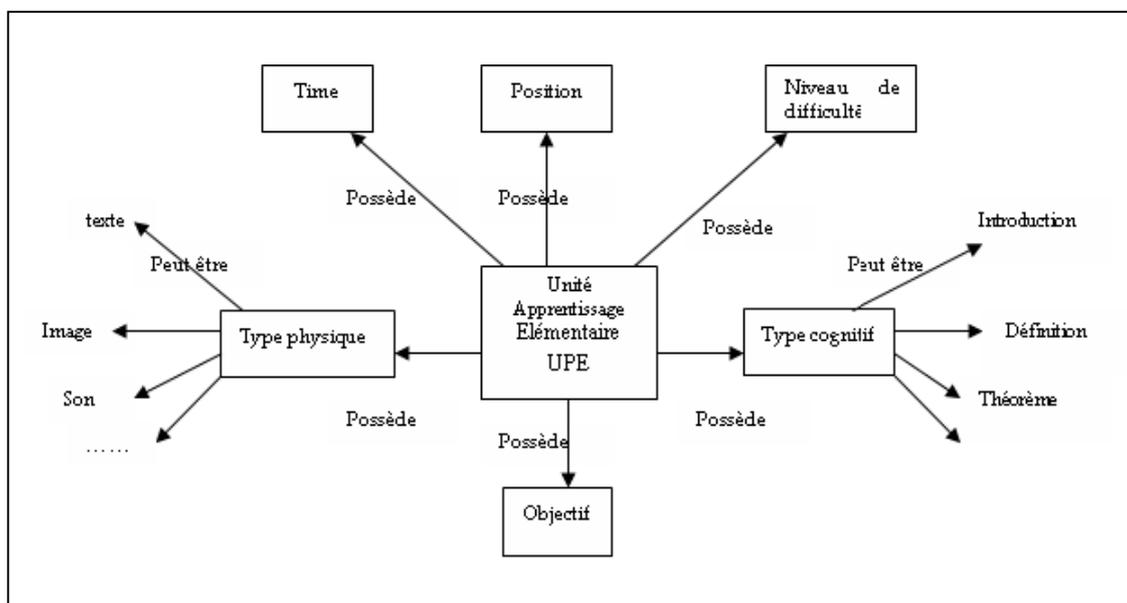


Figure 5.5. Représentation du concept UPE dans le modèle Modap

a) Caractéristiques des unités d’apprentissage hypermédias

Aussi, il faut ajouter des informations de nature sémantique aux uahs de manière à en obtenir une description plus précise. Les métadonnées sont appropriées pour atteindre ce but. Qu’entend-on exactement par « métadonnées »? Nous pouvons définir les métadonnées comme des « données relatives à des données » [Bourda 01]. Il s’agit d’informations décrivant les ressources dans le but de rendre plus performants la recherche de contenus, leur gestion et leur utilisation.

Cet ensemble de métadonnées doit être partageable et reconnaissable entre les auteurs et les apprenants. La norme LOM (Learning Object Metadata) qui signifie « métadonnées des objets pédagogiques », est issu des travaux du groupe de travail Ltsc [Ltsc 04] des IEEE auquel participent de nombreux consortiums dont ARIADNE [Ariadne 04] et IMS [Imms 04]. Cette norme, convergeant vers un standard unique et général permettant d'indexer les objets pédagogiques, est applicable à un très grand nombre de situations éducatives. Elle spécifie la syntaxe et la sémantique des métadonnées pédagogiques et définit les attributs nécessaires pour une description complète des objets pédagogiques afin de pouvoir les gérer et les rechercher.

La version 6 du LOM est plus complète, elle décrit tout ce qui caractérise un objet pédagogique. Elle définit une soixantaine de descripteurs regroupés en neuf catégories [Ltsc 04] dont la description du contenu, des objectifs visés, du public visé, de la configuration matérielle nécessaire, etc. Ces descripteurs ne sont pas évidemment tous obligatoires, chaque concepteur d'objets pédagogiques peut en choisir ceux qui conviennent à ses besoins.

Pour ces raisons nous avons repris la normalisation LOM, mais pour les besoins de notre application, nous avons défini un sous-ensemble de vocabulaire plus simple. Après avoir typé les uahs de cours et d'évaluation relativement aux types d'objectifs pédagogiques auxquels elles sont assignées (uah-but, uah-objectif-général, uah-objectif-spécifique, uah-objectif-opérationnel, uah-question, uah-réponse, uah-exercice, uah-solution), nous avons intégré un certain nombre de métadonnées pouvant faciliter la caractérisation de ces uahs (**Table 5.1**). Tous ces éléments intégrés dans les uahs vont servir à l'individualisation de l'apprentissage et à l'adaptation du cours selon les spécificités caractérisant les apprenants. Nous caractérisons les uah par les métadonnées (METADAM) suivantes :

- **Générales** (identifiant, type-uah, auteur, date de création, langue, temps moyen d'apprentissage, couple niveaux de formation (niveau de départ, niveau cible)) ;
- **Educatives** (titre, concepts, granularité, type de média, type de compétence, objectifs pédagogiques, mode d'évaluation, couple (seuil d'échec, seuil de réussite)).

Où chaque descripteur des « métadonnées générales » exprime :

- 1- Identifiant : un numéro permettant d'identifier sans ambiguïté l'uah dans la hiérarchie de la structure du module d'enseignement, par exemple uah(2,3,1) ;
- 2- Type-uah : un type parmi les quatre types d'uah de cours (uah-but, uah-objectif-général, uah-objectif-spécifique, uah-objectif-opérationnel) ou bien un type parmi les quatre types d'uah d'évaluation (uah-question, uah-réponse, uah-exercice, uah-solution) ;

- 3- Auteur : l'auteur de l'uah ;
- 4- Date de création : la date de création de l'uah ;
- 5- Langue : la langue avec laquelle est rédigé le contenu de l'uah (arabe, français, anglais) ;
- 6- Temps moyen d'apprentissage : la durée moyenne que mettra un apprenant moyen pour parcourir et comprendre le contenu de l'uah ;
- 7- Couple niveaux de formation : le « niveau de départ » de l'apprenant pouvant accéder à l'uah et le « niveau cible » souhaité. Les valeurs possibles de ce descripteur sont (débutant, formé), (moyen, formé), (formé, expert) [Tenachi 94] ;

Et chaque descripteur des « métadonnées éducatives » exprime :

- 8- Titre : le titre de l'uah, une combinaison de mots qui désigne l'intention de l'uah ;
- 9- Concepts : la liste du ou des concepts à apprendre dans le contenu de l'uah. Pour un but de réutilisation de l'uah, il est souhaitable que la liste de valeurs de ce descripteur soit la plus petite possible ;
- 10- Granularité : le nombre d'objectifs pédagogiques à satisfaire par l'uah. Pour un but de réutilisation de l'uah, il est souhaitable que la valeur de ce descripteur soit la plus petite possible ;
- 11- Type de média : le type de média utilisé pour décrire le contenu de l'uah (texte, image, vidéo, animation, etc.) ;
- 12- Type de compétence : le type de compétence à atteindre chez l'apprenant (savoir, savoir-faire) ;
- 13- Objectifs pédagogiques : le ou les objectifs pédagogiques à atteindre une fois le contenu de l'uah maîtrisé par l'apprenant. Nous rappelons que pour un module d'enseignement ayant un « BUT » donné, nous utilisons trois niveaux d'objectifs (objectifs généraux, objectifs spécifiques, objectifs opérationnels). Pour chaque uah l'auteur doit préciser l'objectif ou les objectifs opérationnels visés par l'uah ;
- 14- Mode d'évaluation : le mode d'évaluation de la maîtrise de l'uah par l'apprenant. Les uahs ne sont pas toutes évaluables et donc ce descripteur peut avoir la valeur « indéfini ». Si dans le cas contraire, une évaluation est envisagée pour une uah, ce descripteur peut avoir les valeurs (pronostic-module, formative-uah, inventaire-chapitre, inventaire-partie, inventaire-module) ;
- 15- Couple (seuil d'échec, seuil de réussite) : référentiels auxquels sont confrontés les résultats obtenus par l'apprenant lors de l'évaluation des objectifs visés par l'uah.

Unité d'Apprentissage Hypermédia (UAH)	
Générales	Educatives
Identifiant	Titre
Type-uah	Concepts
Auteur	Granularité
Date de création	Type de média
Langue	Type de compétences
Temps moyen d'apprentissage	Objectifs pédagogiques
Couple (niveau de départ, niveau d'arrivée)	Mode d'évaluation
	Couple (seuil d'échec, seuil de réussite)

Table 5.1. Métadonnées d'une unité d'apprentissage hypermédia

b) Typologie des unités d'apprentissage hypermédi

Outre les caractéristiques des uahs présentés ci-dessus, nous définissons des types distincts d'uah. Ils nous permettent également de les exploiter pour adopter un enseignement adaptatif.

b1) Les uahs de présentation de connaissances sans évaluation

Ces uahs se limitent à la « présentation » des notions et concepts du module d'enseignement sans essayer de mesurer leur acquisition par l'apprenant. Elles peuvent consister en un résumé de module (uah-but) » présentant le *but* du module, des résumés de parties (uah-objectifs-généraux) présentant les objectifs généraux des parties, des résumés de chapitres (uah-objectifs-spécifiques) présentant les objectifs spécifiques des chapitres, mais aussi des uah de type « introduction », « conclusion », « étude de cas », « simulation », etc. Elles ne sont pas forcément statiques et peuvent interagir avec l'apprenant et lui donner l'occasion de pratiquer. En voici un exemple de tels types d'uah (**Figure 5.6**).

b2) Les uahs de fixation des connaissances avec évaluation

En principe, un apprenant est appelé à apprendre, autrement dit, il est appelé pour s'approprier des savoirs et des savoir-faire. Le rôle de l'enseignant est de bannir l'erreur et d'éliminer les fausses représentations pour leur substituer les vérités scientifiques. La recherche des erreurs et des fausses représentations en vue de guider l'apprenant dans la bonne voie fait partie de la tâche d'évaluation. Nous ne pouvons donc imaginer un système d'apprentissage sans

la fonction d'évaluation. Pour cette raison, nous avons tenu compte dans notre système d'assurer cette fonction au début, pendant et après le processus d'apprentissage. Les « uahs de fixation des connaissances » sont présents pour assurer une évaluation formative de l'apprenant.

Contrairement aux « uahs de présentation de connaissances sans évaluation », ces « uahs de fixation des connaissances » (ou uahs évaluables) tentent de juger du degré d'atteinte de leurs objectifs pédagogiques. A ces uahs évaluables sont associées des « uahs d'évaluation » de type questions théoriques et/ou de type exercices permettant une évaluation formative de ces uahs. Un exemple de ce type d'uah évaluable est donné en **(Figure 5.7)**.

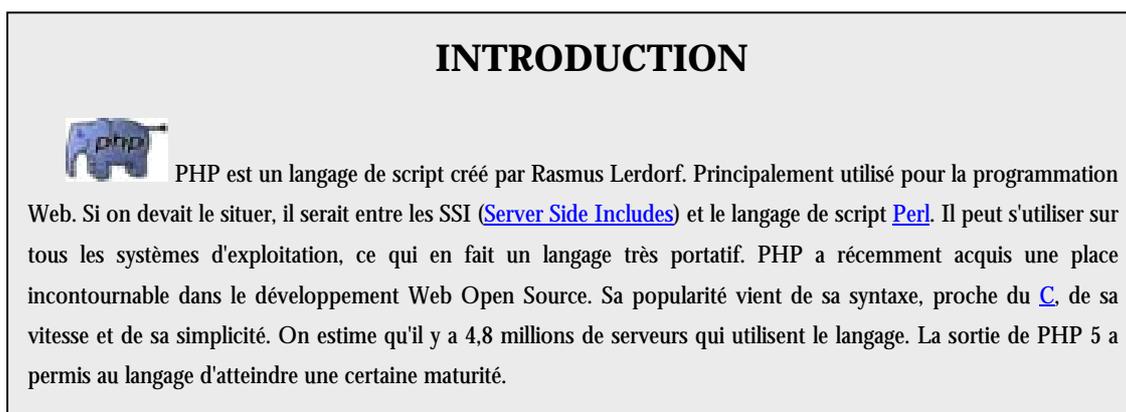


Figure 5.6. Un exemple d'uah sans évaluation : une introduction au langage PHP

b3) Les uahs d'évaluation

Afin de mesurer l'atteinte des objectifs pédagogiques fixés dans les uah évaluables de cours, par l'apprenant, nous avons prévu des uahs d'évaluation qui leur sont associés. Ces uahs consistent en des Questions aux Choix Multiples (QCM) sous forme de questions théoriques et/ou exercices pratiques. L'exécution de ces uahs par le système retourne des valeurs numériques permettant de mesurer d'atteinte des objectifs pédagogiques et ainsi permettre de suivre de près la progression de l'apprenant et le modéliser le plus objectivement possible. Aussi, le degré de finesse des évaluations est primordial pour l'action de remédiation. Les remédiations consistent en des apports de connaissances ciblés sur les erreurs commises par l'apprenant. Suivant l'endroit où l'évaluation est faite, les stratégies tutorielles choisissent les uahs de cours qui conviennent le mieux à remédier à l'erreur de l'apprenant. Un exemple d'une uah d'évaluation **(Figure 5.8)**.

UN PREMIER PROGRAMME EN PHP

Le programme : « Bonjour tout le monde ! »

```
<!DOCTYPE html PUBLIC //W3C//DTD XHTML 1.0 >
<html>
  <head>
    <title>Mon premier script PHP !</title>
  </head>
  <body>
    <?php
      echo 'Bonjour tout le monde !';
    ?>
  </body>
</html>
```

Présentation des premiers concepts

Un programme en php est en général inséré à l'intérieur d'un code écrit en [HTML](#). Le programme débute par la balise `<?php` et se termine par la balise `?>`. Pour que le code en php soit exécuté, il faut :

- que la page porte une extension particulière par exemple `.php` : la page pour donc s'appeler par exemple `index.php`.
- que le [serveur](#) soit configuré pour accepter le PHP. La plupart des hébergeurs proposent en standard des plateformes PHP.

Ce premier programme comporte la commande « écho » qui permet d'afficher une chaîne de caractères écrite entre côtes. Ici le programme affiche le message “Bonjour tout le monde !”. Le programme sera exécuté sur le serveur qui remplacera le programme par le résultat de son exécution. Ainsi le [navigateur](#) recevra la page HTML suivante :

```
<!DOCTYPE html PUBLIC //W3C//DTD XHTML 1.0 >
<html>
  <head>
    <title>Mon premier script PHP !</title>
  </head>
  <body>
    Bonjour tout le monde !
  </body>
</html>
```

Figure 5.7. Un exemple d'uah évaluable : initiation à la programmation en langage PHP

b4) Les uahs de remédiation

Les uah de remédiation sont conçus par l'auteur en corrélation étroite avec les uahs d'évaluation. Elles consistent en des apports ciblés de connaissances pour faire face aux erreurs commises par l'apprenant en cours des activités d'évaluation. Par exemple, si l'apprenant a choisi la deuxième réponse dans l'exemple (**Figure 5.8**), l'uah de remédiation suivante peut lui être présentée (**Figure 5.9**).

TEST D’EVALUATION FORMATIVE N° 1

Soit le programme PHP suivant :

```
<!DOCTYPE html PUBLIC //W3C//DTD XHTML 1.0 >
<html>
  <head>
    <title>Mon premier test </title>
  </head>

  <body>
    <?php
      echo 'Bonjour Html et Php !';
    ?>
  </body>
</html>
```

Question : Que manque t-il à ce programme ? Cocher la ou les bonnes réponses.

1. la balise <?php

2. la balise ?>

3. la balise <body>

Figure 5.8. Un exemple d’une uah d’évaluation

Erreur de mémorisation !

Un programme en php est en général inséré à l’intérieur d’un code écrit en **HTML** entre les balises **<body>** et **</body>**. Le programme php débute par la balise **<?php** ou **< ?** et se termine par la balise **?>**.

Figure 5.9. Un exemple d’une uah de remédiation

5.2.2.2 Les objectifs pédagogiques du module d’enseignement

Un module d’enseignement est conçu pour atteindre un certain nombre d’objectifs pédagogiques répartis à travers les uahs. Dans le contexte de notre système, un objectif pédagogique (OP) est un identificateur instancié par l’auteur. Il est rattaché à ces uahs qui visent à induire l’apprentissage par l’apprenant d’une partie du savoir ou savoir-faire du domaine identifié par cet OP. Un OP est en même temps une notion abstraite de modélisation et une caractéristique de spécification des uahs.

Comme nous l’avons décrit dans le chapitre précédent, nous avons adopté la hiérarchie de trois niveaux d’objectifs pédagogiques pour décrire un module d’enseignement. L’objectif global du module appelé « BUT » est alors décomposé en objectifs généraux (OPG), à leur tour décomposés en objectifs spécifiques (OPS) qui eux aussi à leur tour décomposés en objectifs

opérationnels (OPO). En termes mathématiques, BUT est un ensemble dont les éléments sont les OPG, chaque OPG est à son tour un ensemble dont les éléments sont les OPS, et chaque OPS est à son tour un ensemble dont les éléments sont les OPO.

Alors que le BUT, les OPG et les OPS sont difficilement mesurables, les OPO, eux, sont en principe plus faciles à mesurer si l'auteur arrive à bien les formuler. Toutefois, nous considérons les règles suivantes quant à l'atteinte des OPS, OPG, BUT :

- Un OPS est atteint **Si** tous ses OPO sont atteints **Et** l'apprenant réussit au test inventaire posttest-chapitre.
- Un OPG est atteint **Si** tous ses OPS sont atteints **Et** l'apprenant réussit au test inventaire posttest-partie.
- Le BUT pédagogique du module est atteint **Si** tous ses OPG sont atteints **Et** l'apprenant réussit au test inventaire posttest-module.

a) Evaluation de l'atteinte des objectifs pédagogiques

Un OP est mesurable si l'auteur a conçu une ou plusieurs uahs d'évaluation portant sur cet OP et éventuellement sur d'autres OP et que ces uahs d'évaluation sont capables de retourner au système des résultats d'évaluation des OP concernés. Ces résultats consistent alors en des valeurs numériques représentant le jugement que fait l'uah sur l'atteinte des OP sur lesquels elle porte. Nous avons défini deux seuils de références sur l'intervalle de notation [0,20] pour la mesure de l'atteinte des objectifs visés par une uah : un seuil d'échec **Se** et un seuil de réussite **Sr** qui sont fournis par l'auteur et qui vérifient la relation $\{Se(uah_{i,j,k}) \leq Sr(uah_{i,j,k})\}$ (**Figure 5.10**) :

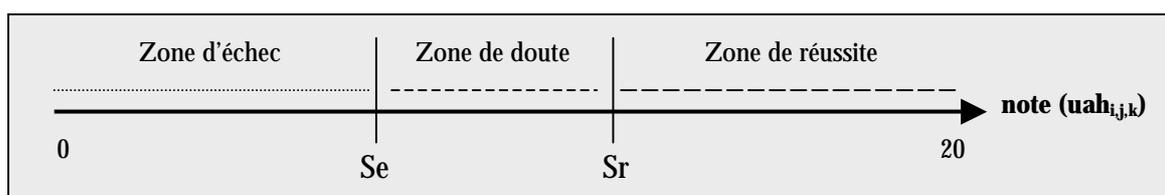


Figure 5.10. Mesure de l'atteinte des objectifs d'une uah

Le système de notation classique utilisé dans la plupart des universités est relativement bien maîtrisé par l'ensemble des enseignants universitaires. Ce système consiste à faire le rapport entre un cumul de points obtenus par l'apprenant aux différentes épreuves d'évaluation et le maximum de points qu'on peut obtenir à ces épreuves (20 points dans les universités algériennes par exemple).

L'étude d'une uah évaluable est toujours suivie par l'exécution de l'uah d'évaluation qui lui est associée pour vérifier l'atteinte des OPO visés. L'évaluation des OPO consiste alors à porter un jugement de valeur sur le degré de réussite de l'apprenant. Selon la **note** obtenue on parle du « degré d'acquisition » de l'OPO ou des OPO de l'uah. Pour pouvoir décider de l'action suivante à effectuer, le système interprète cette note obtenue par l'apprenant. Il lui propose alors de passer aux objectifs suivants ou au contraire les lui déconseille et lui propose alors des objectifs plus adéquats pour renforcer les notions de l'uah courante. L'interprétation de la note est faite selon les trois règles suivantes. :

Si $note(uah_{i,j,k}) \geq Sr(uah_{i,j,k})$ **alors** acquise ($uah_{i,j,k}$) avec une mesure note ($uah_{i,j,k}$) ;

Si $note(uah_{i,j,k}) \leq Se(uah_{i,j,k})$ **alors** non-acquise ($uah_{i,j,k}$) avec une mesure note ($uah_{i,j,k}$) ;

Si $Se(uah_{i,j,k}) \leq note(uah_{i,j,k}) \leq Sr(uah_{i,j,k})$ **alors** douteuse ($uah_{i,j,k}$) avec une mesure note ($uah_{i,j,k}$) ;

La notion de doute sur l'atteinte des objectifs d'une uah permet d'une part une analyse plus fine des résultats de l'évaluation et d'autre part, permet-elle le choix d'objectifs à renforcer lors par exemple d'un échec sur un autre objectif dont un objectif prérequis est douteux. En dépit de ces avantages, la notion de doute peut être ressentie comme une contrainte dans certains contextes de formation. Dans ce cas on peut éliminer la zone de doute simplement en définissant un seul seuil qui serait à la fois seuil d'échec et de réussite [Tenachi 94]. Dans les universités algériennes ce seuil est généralement fixé à la note 10 sur l'échelle [0,20].

b) Le réseau de prérequis des objectifs pédagogiques (REPOP)

Un lien de prérequis entre deux objectifs OP1 et OP2 (dans le sens OP1 vers OP2) définit d'une part une précédence souhaitée par l'auteur entre les deux objectifs, en proposant que l'apprentissage du deuxième objectif OP2 ne peut être effectué qu'après l'atteinte (ou la réussite) du premier objectif OP1, et d'autre part, un lien de progression et aussi un lien potentiel de remédiation. Cette dernière fonctionnalité signifie que le système peut choisir un OP qui est prérequis à un OP sur lequel l'apprenant n'a pas réussi, afin de proposer à ce dernier un apport de connaissances qui portent sur l'OP prérequis.

Dans la structure de REPOP, les prérequis directs d'un objectif OP_i sont une liste d'OP fournie par l'auteur et dont les liens de prérequis aboutissent à cet OP_i . Par transitivité, nous considérons que tous les prérequis des prérequis directs de OP_i sont des prérequis indirects de OP_i . En tenant compte de la hiérarchie des OP, le réseau de prérequis est composé de trois

niveaux de sous réseaux d'OP. Le sous réseau des OPGs, les sous réseaux des OPSs et les sous réseaux des OPOs (**Figure 5.11**).

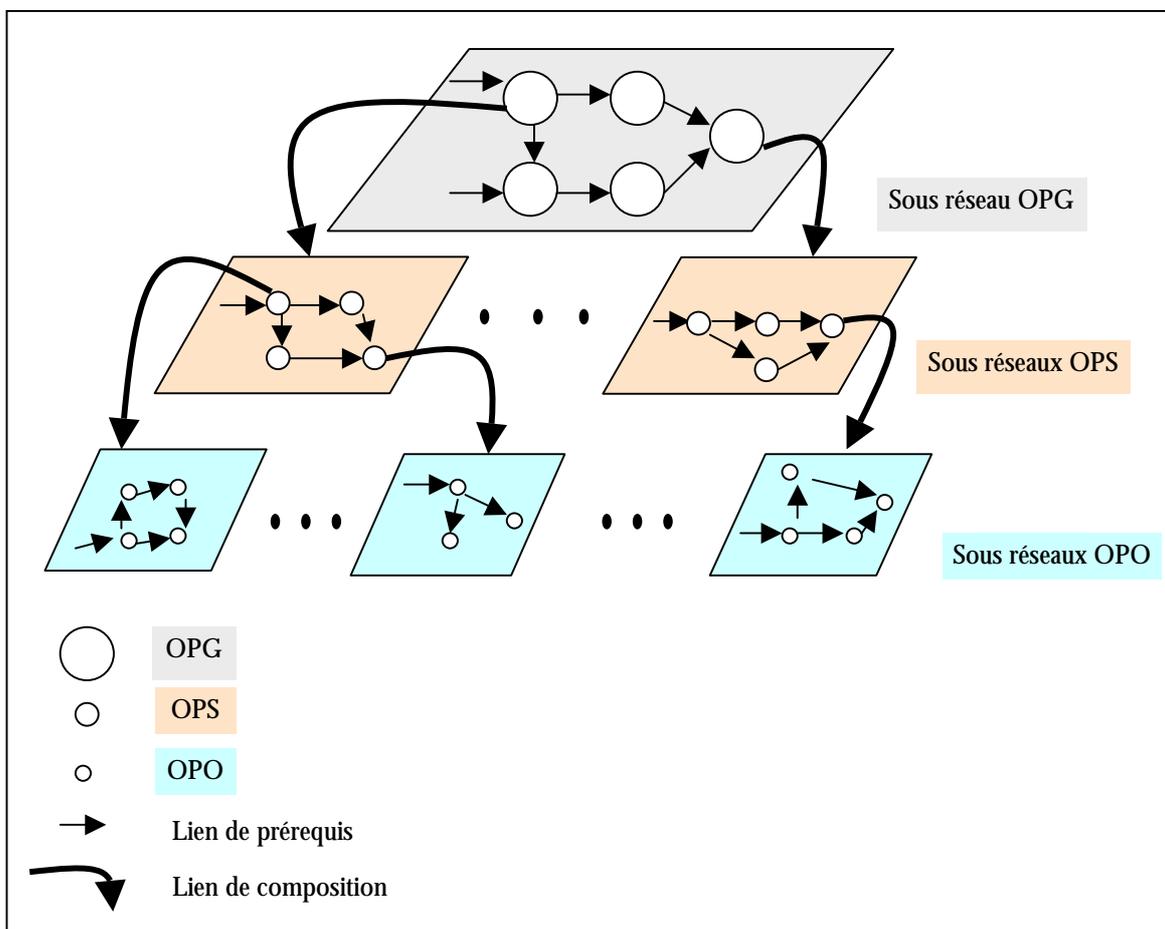


Figure 5.11. Les trois niveaux du réseau de prérequis des OPs

5.2.2.3 Les « concepts » à apprendre dans un module d'enseignement

En principe, un module d'enseignement est conçu pour inculquer un certain savoir et/ou savoir faire chez l'apprenant. Si nous reprenons la taxonomie de Bloom [Bloom 56] discutée au chapitre 4 (**Table 5.1**), nous remarquons que les trois premiers niveaux concernent les savoirs alors que les trois autres niveaux concernent les savoir faire [Muller 06].

Le premier niveau consiste à demander à l'apprenant de « retenir et d'avoir des connaissances » sur (des éléments de connaissances, des façons de mettre en relation des éléments, des systèmes de relations abstraites, etc.). Le deuxième niveau consiste à leur demander de « comprendre » le sens du message en le prouvant par des (traductions, interprétations, extrapolation, etc.). Le troisième niveau consiste à leur demander d'« appliquer d'une manière concrète » un principe.

Le quatrième niveau consiste à lui demander d'« analyser » un ensemble complexe de connaissances, c'est à dire y découvrir des (éléments, relations entre les éléments, principes organisateurs, etc.). Quant au cinquième niveau, il consiste à lui demander de « faire une production » personnelle ou une synthèse dans l'ordre de (l'expression personnelle, l'élaboration d'un plan d'action, dérivation d'un ensemble de relations abstraites, etc.). Et enfin, le sixième niveau consiste à demander à l'apprenant de « juger ou d'évaluer » sa production ou celle d'autrui relativement en elle même ou en référence à autre chose.

Niveau	Verbes d'action à utiliser pour formuler un objectif pédagogique
CONNAISSANCE	abrévier choisir citer cocher copier couper désigner dire donner donner la définition encercler épeler esquisser exclure identifier inscrire insérer localiser marquer montrer nommer noter placer prononcer réciter reconnaître répéter sélectionner séparer situer souligner
COMPREHENSION	construire critiquer démonter différencier discriminer distinguer estimer formuler intégrer utiliser
APPLICATION	adapter appliquer compléter employer illustrer interpréter poser un problème pratiquer reporter traduire transférer transposer utiliser vulgariser
ANALYSE	décomposer désassembler disséquer diviser examiner expliquer extraire prendre partie rechercher séparer simplifier
SYNTHESE	assembler compiler composer construire créer édifier expliquer façonner former un tout intégrer mettre ensemble produire rassembler recombinaison reconstruire regrouper remettre en ordre réorganiser structurer systématiser
JUGEMENT (EVALUATION)	apprécier donner selon l'ordre estimer évaluer évaluer selon expertiser juger vérifier par des tests

Table 5.2. Les six niveaux de la taxonomie de Bloom et les verbes d'action envisagés

a) La notion de savoir

Le « savoir » est défini habituellement comme un ensemble de connaissances ou d'aptitudes reproductibles, acquises par l'étude ou l'expérience. En français, le terme de savoir a un sens qui ne coïncide pas exactement avec celui de « connaissances » alors que par exemple l'anglais utilise le terme « knowledge » dans tous les cas.

Le savoir se distingue par divers traits d'un ensemble de connaissances en particulier par la dimension qualitative : l'acquisition d'un savoir véritable suppose un processus continu d'assimilation et d'organisation de connaissances par le sujet concerné, qui s'oppose à une simple accumulation et rétention hors de toute volonté d'application. Au niveau individuel, le savoir

intègre donc une valeur ajoutée en rapport avec l'expérience vécue et de multiples informations contextuelles.

Chaque personne organise et élabore son savoir en fonction de ses intérêts et besoins ; la composante consciente et volontaire de cette élaboration s'appelle la métacognition. La plupart des « savoirs » individuels sont naturellement utiles à l'action, à sa performance, sa réussite : «savoir, c'est pouvoir !». C'est aussi sur des mises en situation que reposent les meilleures évaluations du savoir alors que des tests basés sur la seule restitution d'informations ne garantissent pas sa qualité et par conséquent sa valeur.

De même, le savoir se rend plus visible et pratique sous le nom de « savoir-faire », « savoir-vivre », etc. Les savoirs les plus intellectuels reposent sur l'appropriation ou création de la notion de « **concepts** », en parallèle avec le développement de la philosophie [Mendelsohn 94].

Chaque communauté repose sur un savoir commun ; c'est une composante de son identité. Le poids et la reconnaissance de ce savoir et donc du savoir présentent des formes variables, mais le sort de la communauté est généralement lié à la conservation de ce patrimoine immatériel. Au sein des sociétés et cultures, l'éducation a pour mission d'aider à l'appropriation du savoir collectif élémentaire, on parle ainsi d'acquisition d'un socle commun, l'enseignement complétant l'acquisition de connaissances et savoir-faire disciplinaires, pendant que la formation professionnelle est chargée de la transmission des savoirs professionnels.

Dans toute société, le savoir prend donc l'allure d'un bien et même d'un « bien économique » à gérer. On parle alors de : acquisition du savoir, accès au savoir, appropriation des savoirs, transmission du savoir, échanges de savoirs, partage du savoir, circulation du savoir , gestion du savoir, maîtrise des savoirs, valorisation des savoirs, etc.

Selon les époques et les cultures, la conservation du savoir et la transmission des connaissances s'appuient sur la communication orale et l'expression écrite et imagée. L'efficacité de la transmission étant pour une part fonction de la plasticité mentale de l'apprenant, elle-même fonction de son âge en particulier, la pédagogie étudie les conditions de ces transmissions entre apprenants et leurs enseignants plus expérimentés ou plus savants.

En plus de la pédagogie de transmission (enseignant vers apprenant), une autre pédagogie considère qu'au lieu d'ingurgiter des savoirs formels dont ils ne perçoivent pas clairement la fonction mais qu'il s'agit tout simplement de mémoriser puis de restituer pour satisfaire aux exigences d'une institution peu ambitieuse, les apprenants d'une classe coopérative en concevant

puis en réalisant coopérativement des projets auront l'occasion de se construire de véritables savoirs qui ont un sens pour eux, puisque acquis pour pouvoir poursuivre leur projet.

De notre point de vue, nous trouvons que les deux pédagogies ne doivent pas être exclusives mais complémentaires. Quoiqu'on dise sur les capacités individuelles de l'apprenant, nous pensons qu'il a souvent besoin d'être stimulé, guidé et corrigé par quelqu'un qui est plus connaisseur et expérimenté que lui et qu'on désigne habituellement par enseignant. C'est ce qui fait que dans notre système nous avons tenu compte des deux aspects en offrant des fonctionnalités de transmission de connaissances et des fonctionnalités de collaboration entre apprenants et enseignants. Quant au savoir nous proposons qu'il soit organisé autour de la notion de « concepts », terme que nous expliquons par la suite.

b) La notion de « savoir-faire »

Le savoir-faire : compétence, habileté, acquise par l'expérience dans des problèmes pratiques, dans l'exercice d'un métier. C'est aussi un ensemble de connaissances, d'expériences et de techniques accumulées par un individu ou une institution. Les savoir-faire ont été appris et acquis par l'exercice et par l'expérience qui les confirment, les assurent, les développent [Muller 06]. Le champ conceptuel des savoir-faire est loin d'être stabilisé. On peut, en effet, combiner le verbe savoir avec bien d'autres verbes d'action : dire, penser, vivre, être, devenir, etc. Le savoir-faire s'est stabilisé car il englobe, sans doute, tous les autres.

Le terme « savoir-faire » unit en une seule expression deux termes qui demeurent, malgré l'usage du trait d'union, dissociés. En effet il est commun de penser que posséder un savoir est une chose, et qu'agir en est une autre [Mendelsohn 94]. Il n'est pas toujours facile de distinguer les différents types de savoir-faire. Cependant un certain nombre de spécialistes décomposent les savoir-faire en sous-catégories spécifiques [Muller 06] :

- les savoir-faire méthodologiques qui permettent d'utiliser des savoirs pour synthétiser, analyser, appliquer, évaluer, etc.
- les savoir-faire relationnels qui reposent sur des micro-comportements : accueillir, négocier, argumenter, etc.

On oppose souvent savoirs et savoir-faire, or la tension entre les deux est fondatrice de toute démarche pédagogique. Cette opposition se résume aujourd'hui dans la distinction entre:

- les connaissances procédurales qui sont prescriptives et qui indiquent comment il faut procéder pour acquérir une compétence. Elles se constatent dans l'activité ;

- les connaissances déclaratives qui représentent des faits et des principes que l'on peut formuler et énoncer. On parlait autrefois de culture générale et culture professionnelle. Ces connaissances ont trois fonctions principales : communiquer des informations, contrôler le déroulement de l'action, procurer un cadre pour généraliser.

Dans l'université traditionnelle, le savoir théorique est censé toujours précéder le savoir pratique : il faut savoir pour savoir faire. Cette situation ne plait pas évidemment à certains pédagogues qui considèrent que les savoir-faire sont souvent méprisés par l'université qui a tendance à dénigrer les situations authentiques d'apprentissage telles qu'on peut les rencontrer dans la vie. Freinet a très bien illustré cette conception avec la métaphore du vélo. Heureusement, disait-il, qu'on n'apprend pas à faire du vélo à l'école sinon avant d'avoir le droit de faire des essais, il faudrait d'abord apprendre les lois de l'équilibre et le nom des différentes pièces ! Les enseignants affirment qu'ils ont pour objectifs le développement de compétences ou de savoir-faire mais ils interviennent d'abord sur les savoirs. Or dans les apprentissages "naturels" les exemples sont nombreux où les savoir-faire précèdent les savoirs : langage, habiletés motrices et sportives, savoir-faire implicites ou savoirs d'expérience des professionnels...

En résumé, nous pouvons dire que la question des savoir-faire est au centre des débats qui opposent partisans des méthodes actives ou de la pédagogie nouvelle et partisans des méthodes traditionnelles « qui ont fait leurs preuves » (?).

Pour la pédagogie traditionnelle qui refuse de prendre en compte la façon dont on apprend naturellement dans la vie, il suffit de savoir pour savoir-faire. L'école traditionnelle met incontestablement l'accent sur les savoirs : règles, résumés, tables, etc., qu'il faut souvent apprendre par cœur sans s'interroger ni sur leur pertinence, ni sur leur efficacité.

La classe coopérative, qui s'inscrit dans le courant de la pédagogie nouvelle, prend davantage en compte la complémentarité : savoirs/savoir-faire dans la mesure où elle propose des tâches finalisées, en particulier sous la forme de projets qui impliquent et favorisent la maîtrise de connaissances procédurales et de connaissances déclaratives.

De plus, la classe coopérative, par l'importance qu'elle accorde à la réflexion sur les pratiques, permet de « s'interroger sur la manière dont les savoirs sont transformés en savoir-faire et dont

les savoir-faire sont interrogés par les savoirs » et favorise ainsi la construction de véritables compétences.

De notre point de vue, nous optons pour une approche contextualiste qui met en valeur la complémentarité savoirs/savoir-faire et les liens profonds qui existent entre ces concepts. L'apprenant apprend en utilisant des savoirs et des savoir-faire intimement associés aux contextes qui leur donnent du sens. Savoirs et savoir-faire sont à notre avis de véritables partenaires impliqués dans des tâches finalisées. Apprendre c'est en permanence réduire l'écart entre le déroulement d'une action et ce qu'on sait de cette situation pour l'avoir déjà rencontrée ou pour avoir rencontré des situations analogues.

L'apprentissage est un processus de gestion du flux d'informations qui traverse l'apprenant au fur et à mesure qu'il exécute une tâche. les savoirs sont transformés en savoir-faire et les savoir-faire sont interrogés par les savoirs.

De cette manière, le savoir se rend plus visible et pratique sous le nom de « savoir-faire ». Les savoirs et savoir-faire les plus intellectuels et ainsi unis reposent alors sur l'appropriation ou création de la notion de « **concepts** », notion sur laquelle repose notre modèle du domaine et que nous développons ci-dessous.

c) La notion de « concept »

Comme nous venons de l'expliquer ci dessus, nous proposons que les savoirs et savoir-faire qui font le support du module d'enseignement soient organisés autour de la notion de « concepts ». Nous allons donc décrire cette notion.

c1) Qu'est-ce qu'un concept ?

Il est presque banal d'utiliser aujourd'hui le terme de concept. A tel point que son usage passe-partout ou tout-terrain finit par faire écran à un minimum de clarté. Les concepts sont partout, ils prolifèrent dans tous les domaines, même les plus étranges, mais ils ne sont définis que rarement en tant que tels. Traditionnellement, deux tendances s'opposent lorsqu'il s'agit d'aborder la notion de concept. La première, « idéaliste/réaliste », pose une vision dite réaliste du concept : ce dernier préexiste, qu'il soit immanent ou issu d'un monde intelligible, et est ensuite perçu par l'individu. La seconde, « nominaliste », suppose que le concept est une construction personnelle, et qu'il n'existe qu'au sein d'une culture, voire d'un individu donné. Chacune de ces visions tire ses présupposés d'idées philosophiques que nous allons aborder, avant de présenter leurs conséquences respectives au niveau pédagogique.

- **La tendance Idéaliste/Réaliste**

D'après le courant idéaliste, qui remonte à Platon, il existe deux mondes distincts. Le monde matériel n'est qu'une projection, nécessairement imparfaite, du monde des Idées (ou monde intelligible). Ces Idées, que l'on peut identifier aux Concepts, ont donc une existence en elles-mêmes. La connaissance la plus pure est relative à ces Idées, non aux phénomènes concrets qui n'en sont que des manifestations, comme les ombres sur la paroi d'une grotte ne sont qu'une manifestation des objets projetés.

Proche de cette tendance idéaliste, existe également un courant réaliste. Ce dernier présuppose lui aussi que les concepts existent par eux-mêmes, mais pas dans un autre monde : ils sont immanents, par opposition aux Idées qui, elles, sont issues d'un autre monde. Le précurseur de ce courant est Aristote.

Dans tous les cas, que l'on considère que les Idées viennent d'un monde dual ou non, on considère néanmoins que le concept préexiste à l'individu, ce dernier se contentant de découvrir son existence lors de ses rencontres avec divers phénomènes. Ces deux approches ont eu particulièrement le vent en poupe au cours de l'histoire de la philosophie.

Conséquence éducative: pédagogie transmissive et modélisation par recouvrement

Une telle vision du savoir a de nombreuses applications traditionnelles dans le domaine pédagogique. La principale est l'idée qu'il existe une réalité objective que l'apprenant doit découvrir. On parlera ainsi, dans ce cadre, du « savoir » comme un idéal à atteindre. L'enseignant est celui qui dispose de ce savoir : même s'il ne le maîtrise pas en totalité, il en est le garant. La voie d'enseignement privilégiée sera alors la pédagogie transmissive, par le biais du cours magistral, l'enseignant est alors le maître.

Cette vision suppose que l'apprenant s'approche petit à petit d'un savoir idéal, que peu à peu ses connaissances recouvrent les connaissances de l'expert (l'enseignant). La technique de modélisation de l'apprenant dans les STI dite modélisation « par recouvrement » repose globalement sur une telle vision. On considère ainsi que l'objectif d'un enseignement est de faire passer les connaissances d'un apprenant, pour un ensemble de notions déterminées a priori, d'un état à un autre [Talhi et al 05].

- **La tendance Nominaliste**

À l'idée d'une existence a priori des concepts s'oppose, une idée qui sera appelée « nominalisme ». Cette idée part en fait du principe opposé aux conceptions exposées précédemment. Chez ses tenants, les concepts sont des constructions individuelles et sont donc créés par déduction à partir de l'observation de régularités dans les phénomènes observés. Ainsi, à force de voir des hommes, et en observant les points communs existant entre eux, un individu va construire le concept d'homme, selon une conception qui lui est propre.

Sont donc considérés comme universaux non pas les concepts, mais les noms qui les désignent, d'où l'appellation de « nominalisme ». Par exemple, dans l'affirmation « Tous les hommes sont mortels », les nominalistes considèrent que le concept d'homme n'a pas d'existence propre et universelle ; seul le nom « homme » en a une.

Conséquence éducative : apprentissage par découverte

De telles idées sont notamment à la base de tous les travaux autour du constructivisme et de ses variantes [Vygotski 85], [Varela 96]. Puisque le savoir n'existe que chez l'individu, et n'a pas d'existence en dehors de lui, l'enseignant ne peut pas l'inscrire dans l'apprenant si ce n'est en ce qui concerne certains faits ponctuels, qui ne seront de toute façon retenus que s'ils font sens chez l'apprenant.

Selon donc ce point de vue, c'est l'apprenant qui crée du sens à partir de sa propre expérience sensible. Pour un ensemble de phénomènes donné, il crée donc un modèle explicatif qui donne un sens à cet ensemble de phénomènes ; on parle également de « conception » [Vergnaud 91]. Il est important de noter que, d'un certain point de vue, une conception est nécessairement juste. En effet, si une conception est construite par un individu, c'est qu'elle lui permet de répondre à un problème donné [Giordan 98].

Un exemple simple de conception correcte d'un certain point de vue, bien qu'incorrecte scientifiquement, concerne l'utilisation des nombres décimaux. Un enfant peut posséder, au début de son apprentissage, la conception suivante : « un nombre décimal, c'est deux nombres entiers séparés par une virgule ». Cette conception coïncide avec ce que l'enfant voit, et fonctionne même dans certains cas avec l'addition : additionner deux nombres décimaux, c'est additionner les deux nombres à gauche et les deux nombres à droite. Soit par exemple l'opération $(14,55 + 5,44)$. On calcule $(14 + 5 = 19)$ et $(55 + 44 = 99)$, on insère la virgule, ce qui donnera

(14,55 + 5, 44 = 19,99). Le résultat est correct et si tous les exemples donnés par l'enseignant sont de cette nature, l'enfant pourra donc construire une telle conception.

Le problème se pose bien entendu lorsque l'on souhaite additionner, par exemple, « 14,55 » et « 5,55 ». Dans ce cas le résultat proposé par l'enfant est 19, 110. Le rôle de l'enseignant sera alors non pas seulement de corriger en surface l'erreur mais bien de remettre en cause la conception de l'apprenant et de l'amener à en construire une nouvelle, plus adaptée à la réalité et lui permettant de résoudre correctement les problèmes posés. Il est donc important de noter que le savoir n'évolue pas de manière continue et localisée, mais par ruptures successives de conceptions insatisfaisantes.

- **Mode opératoire du concept**

Dans la définition donnée par l'académie française [Barth 93], les concepts sont des créations, voire des procréations (créations en vue d'un projet). Ils s'ajoutent en quelque sorte à ce qui est perçu immédiatement et sont les briques élémentaires de l'univers intellectuel de l'être humain.

Ces constructions de l'esprit consistent en l'utilisation d'un **signe verbal**, d'un **mot**, d'une **expression**, pour représenter un ensemble de caractères communs et de propriétés communes. Un concept se définit donc tant par ces propriétés (définition en compréhension) que par la série d'objets auxquelles elle s'applique (définition en extension).

D'après [Barth 93], un concept est constitué de trois composantes interdépendantes :

- l'étiquette (ou le nom) du concept, en termes linguistiques le **signifiant**,
- l'ensemble des attributs nécessaires et suffisants pour qu'un phénomène relève du concept, en termes linguistiques le **signifié**, en termes ensemblistes la **définition en intension**,
- un ensemble d'exemples, aussi vaste que possible, d'instances concrètes du concept, en termes ensemblistes sa **définition en extension**.

Selon Barth, ces trois composantes sont aussi importantes les unes que les autres, et un enseignement conceptuel ne peut être efficace s'il ne s'intéresse pas d'une manière ou d'une autre à ces trois aspects du savoir. En effet, connaître un concept, c'est :

1. être capable, lorsqu'un phénomène quelconque est présenté, de savoir à quel(s) concept(s) connu(s) il fait référence, même s'il n'a jamais été rencontré auparavant,
2. savoir associer, à un nom de concept donné, l'ensemble des attributs qui le définissent.

Dans notre contexte, nous retenons qu'un concept est un couple (**signifiant, signifié**) représentant une connaissance descriptive de l'uah, où le **signifiant** est un terme ou une expression formée d'un mot ou d'une juxtaposition de mots et le **signifié** est une partie du savoir et/ou savoir-faire qui recouvre l'apprentissage du concept selon l'auteur du module d'enseignement (définition, règle, théorème, exemple, exercice, activité, etc.). Il peut s'agir de la description d'un ensemble d'objets, d'une tâche, d'une fonction, d'une stratégie d'un processus de raisonnement, etc., le tout réparti à travers un ensemble d'uahs. Le travail de Behaz [Behaz et Djoudi 05] s'intéresse à une granularité plus fine de la notion de concept en vue de faire de l'adaptation dynamique dans le processus d'apprentissage.

Voici quelques exemples de concepts que nous pouvons retenir dans le module d'enseignement « Systèmes d'information » : base de données, modèle de données, entité, relation, attribut, dépendance fonctionnelle, normalisation, modèle de traitement, événement, diagramme de flux, unité fonctionnelle, unité organisationnelle, modèle de communication, etc.

c2) Réseau de prérequis des concepts (REPC)

Le module d'enseignement, ayant un BUT pédagogique à atteindre, est donc conçu pour permettre à l'apprenant de comprendre un certain nombre de concepts désignés par des identifiants (les signifiants). De toute évidence, un concept n'est défini qu'en prenant appui et en utilisant des termes (et donc des concepts) déjà connus. Autrement dit, toute construction intellectuelle ne peut se faire qu'à partir de briques élémentaires déjà certifiées et correctement estampillées dans le domaine d'enseignement.

L'apprentissage d'un concept ne peut donc se concrétiser que par l'apprentissage d'un ou de plusieurs concepts prérequis. Ceci exige donc, comme pour les objectifs pédagogiques, un réseau de prérequis des concepts (REPC). Un lien de prérequis entre deux concepts C1 et C2 (dans le sens C1 vers C2) définit d'une part une « précédence » souhaitée par l'auteur entre les deux concepts, en proposant que l'apprentissage du deuxième concept C2 ne peut être effectué qu'après l'apprentissage du premier concept C1, et d'autre part, un « lien de progression » et aussi un lien potentiel de « remédiation » (**Figure 5.12**).

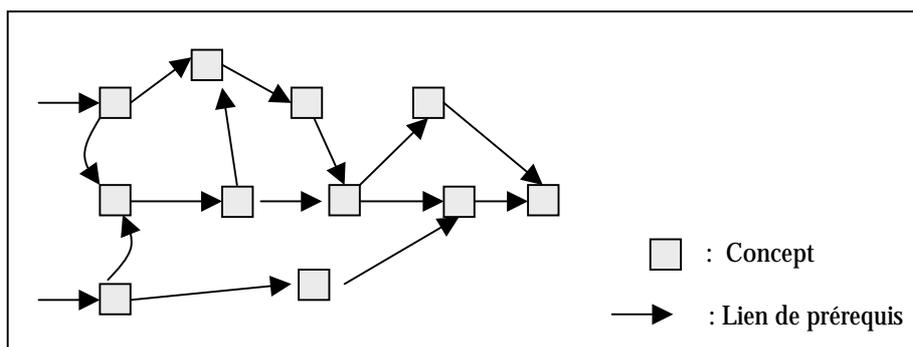


Figure 5.12. Réseau de prérequis des concepts (REPC)

5.2.3 Les paramètres quantitatifs du module d'enseignement

Ces paramètres décrivent l'aspect quantitatif concernant le module d'enseignement. Les paramètres consistent essentiellement en trois prédicats relatives à la matière théorique et douze prédicats relatives à la matière d'évaluation. Nous rappelons qu'un prédicat est une fonction prenant un ou plusieurs arguments [$f(a_1, a_2, \dots)$] et retournant l'une des valeurs vrai ou faux. Les quinze classes de prédicats constituant les paramètres du module sont les suivantes :

- 1) nbrepar (nbpar) : nombre de parties du module d'enseignement ;
- 2) nbrechap(numpar, nbchap) : nombre de chapitres d'une partie donnée de numéro numpar) ;
- 3) nbreuah(numpar, numchap, nbpar) : nombre d'uah d'un chapitre donné (de numéro numchap) d'une partie donnée (de numéro numpar) ;
- 4) nbreques(numpar, numchap, numuah, nbques) : nombre de questions associées à une uah donnée avec toute la hiérarchie de l'uah ;
- 5) nbrerep(numpar, numchap, numuah, numques, nbrep) : nombre de réponses associées à une question donnée avec toute la hiérarchie de la question ;
- 6) nbreexo(numpar, numchap, numuah, nbexo): nombre d'exercices associés à une uah donné avec toute la hiérarchie de l'uah ;
- 7) nbresol(numpar, numchap, numuah, numexo, nbsol): nombre de solutions associées à un exercice donné avec toute la hiérarchie de l'exercice ;
- 8) nbreepret(nbepret): nombre d'épreuves du pretest ;
- 9) nbresolpret(numepret, nbsolpret): nombre de solutions-de-pretest associées à une épreuve-de-pretest donnée de numéro numepret ;
- 10) nbreepostc(numpar, numchap, nbepostm) : nombre d'épreuves du posttest d'un chapitre de numéro numchap ;

- 11) `nbresolpostm(numpar, numchap, numepostc, nbresolpostm)` : nombre de solutions-de-posttest associées à une épreuve-de-posttest-chapitre donnée de numéro `numepostc`.
- 12) `nbreepostp(numpar, nbepostp)` : nombre d'épreuves du posttest d'une partie de numéro `numpar` ;
- 13) `nbresolpostp(numpar, numepostp, nbresolpostp)` : nombre de solutions-de-posttest associées à une épreuve-de-posttest-partie donnée de numéro `numepostp` ;
- 14) `nbreepostm(nbepostm)` : nombre d'épreuves du posttest du module d'enseignement;
- 15) `nbresolpostm(numepostm, nbresolpostm)` : nombre de solutions-de-posttest associées à une épreuve-de-posttest-module donnée de numéro `numepostm`.

Lors de la création d'un STI, toutes ces classes de prédicats sont instanciées par les auteurs en utilisant le « module d'acquisition des paramètres » du mode auteur. Le nombre d'instances varie d'un STI à un autre. Il est fonction du volume de la matière à enseigner et des découpages effectués par l'auteur. A titre d'exemple, si un module est divisé en deux parties comportant trois chapitres chacune, alors la deuxième classe de prédicat donne naissance à deux instances: `nbrechap (1,3)` et `nbrechap (2,3)`.

5.2.4 Le module d'enseignement

Dans tout ce qui précède, nous avons donné une description détaillée de tous les objets qui constituent un module d'enseignement dans le système Hits. Pour résumer, nous en donnons enfin le modèle conceptuel dans la (**Figure 5.13**). Les noeuds qui ne sont pas étiquetés dans la figure sont les instances de « paramètres » (ce sont les quinze prédicats listés ci dessus) et les « métadonnées » de l'uah (ce sont les quinze propriétés qui sont listés dans la table 5.1).

Plus concrètement, un module d'enseignement est composé d'un ensemble d'uahs de différents types (`uah-but`, `uah-objectif-général`, `uah-objectif-spécifique`, `uah-objectif-opérationnel`, `uah-question`, `uah-réponse`, `uah-exercice`, `uah-solution`). Nous appelons cet ensemble « univers des uah ». Cet univers est structuré sous forme d'un arbre de prérequis formé de six niveaux d'uah (les quatre premiers concernent les uahs de cours, et les deux autres niveaux concernent les uahs d'évaluation). Chaque nœud de l'arbre désigne un identifiant d'une uah.

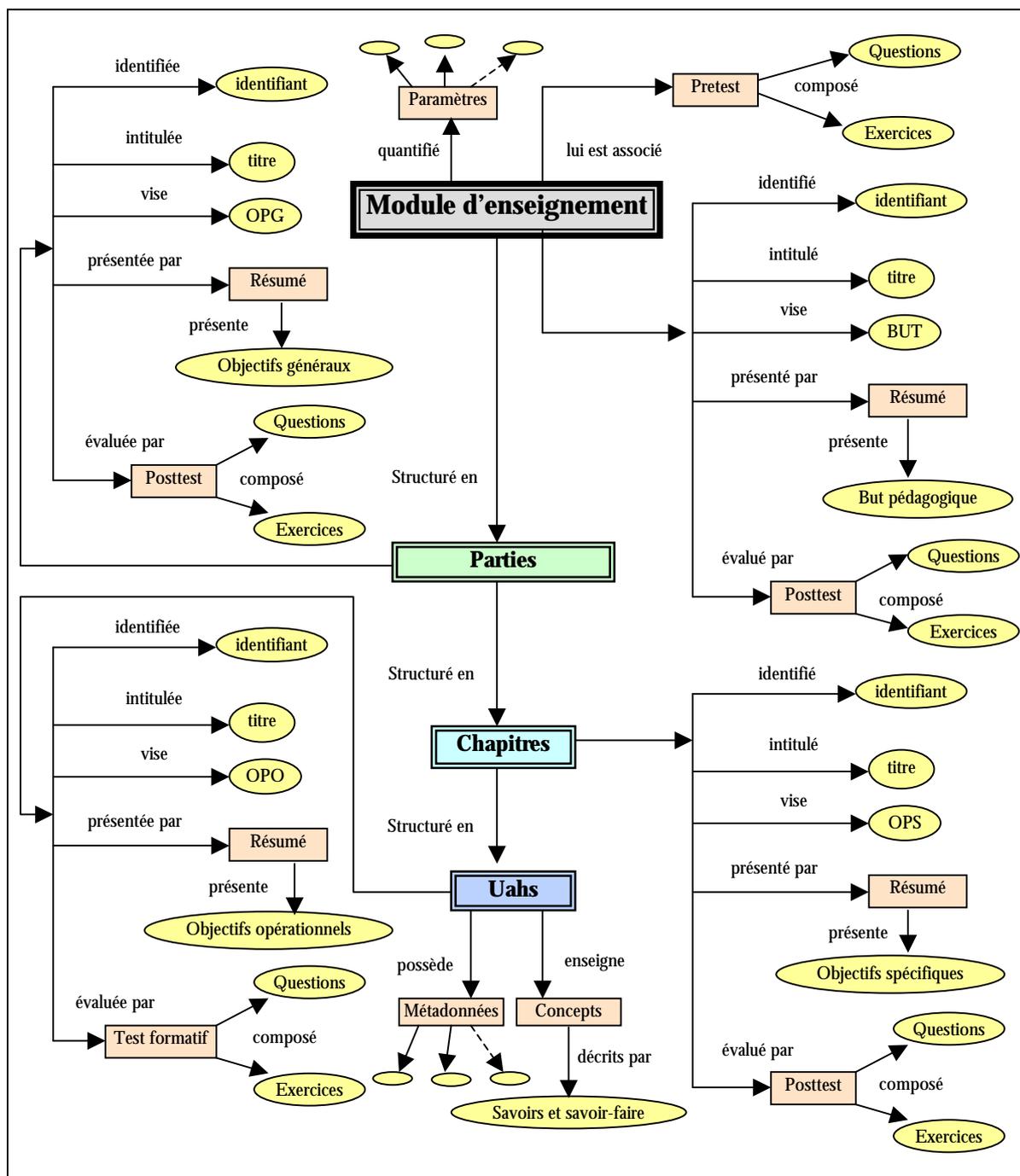


Figure 5.13. Modèle conceptuel du module d'enseignement

A la racine, nous trouvons le résumé du module qui décrit le « but du module » suivi au deuxième niveau des résumés des parties qui décrivent les OPG suivis au troisième niveau des résumés des chapitres qui décrivent les OPS suivis au quatrième niveau des uahs qui visent à atteindre les OPO suivis au cinquième niveau par les « uah-question » et les « uah-exercice » suivis enfin au sixième niveau par les « uah-réponses » et les « uah-solutions ». Toutes les uahs sont

explicitement ou implicitement reliées entre elles par un lien de prérequis. Ces liens s'expriment comme suit :

- les liens explicites sont les liens « père-fils » qui indiquent que le père est un prérequis du fils.
- les liens implicites sont les liens correspondants à un parcours « en profondeur d'abord » de l'arbre de prérequis.

Chaque nœud du deuxième niveau avec toute sa descendance représente une « partie » du module d'enseignement. Chaque nœud du troisième niveau avec toute sa descendance représente un chapitre d'une partie.

5.2.5 Les règles de recherche des uahs

En plus de toutes les structures de connaissances citées, le « Module expert » du domaine contient aussi les méthodes permettant la recherche des uahs pour les mettre à la disposition de l'apprenant, quand il est sollicité par le pédagogue « Module tutoriel » ou le « Module de diagnostic ». Cette recherche est effectuée par un système expert basé sur des règles de production et fonctionnant en chaînage avant. A chaque uah est alors associée une règle permettant la recherche et l'affichage de cette uah. L'ensemble de ces règles, dites « règles instances de recherche des uahs », forme le paquet de règles (RIREC). Ces règles sont obtenues par instanciation des « règles maîtresses de recherche des uahs » (RMREC) et elles ont, toutes, la même forme suivante :

Si *présenter-<identificateur d'uah>* **Alors** *afficher-<identificateur d'uah>* **Et** *présenté-<identificateur d'uah>*

et où les faits ont la signification suivante :

- *présenter-<identificateur d'uah>* est un message envoyé par le « Module tutoriel » ou le « module de diagnostic » pour réveiller le « Module expert » et lui demander d'afficher l'uah en question (uah de cours pour le « Module tutoriel » et uah d'évaluation pour le « Module de diagnostic ») ;
- *afficher-<identificateur d'uah>* est un sélecteur d'une procédure (i.e. nom d'une procédure). Cette procédure est chargée de chercher et d'afficher l'uah en question et de rester en contact avec l'apprenant jusqu'à ce que ce dernier interrompe la liaison avec elle en appuyant un bouton « lecture terminée » par exemple, l'informant par ce geste qu'il a terminé l'apprentissage de l'uah et qu'il veut passer à autre chose ;

- *présenté- \langle identificateur d'uah \rangle* est le message d'acquittement renvoyé par le « Module expert » au module appelant l'informant que sa mission est accomplie.

A titre d'exemple, une règle instance affichant l'uah(1,2,3) se présente comme suit:

RIR_i **Si** *présenter-uah(1,2,3)* **Alors** *afficher-uah(1,2,3)* **Et** *présenté-uah(1,2,3)*

Cette règle instance est obtenue par instanciation d'une règle maîtresse permettant de générer toutes les règles instances affichant les uah d'un module d'enseignement donné. Cette règle maîtresse est formulée comme suit :

RMR_j **Si** *présenter-uah(i,j,k)* **Alors** *afficher-uah(i,j,k)* **Et** *présenté-uah(i,j,k)*

L'instanciation des règles maîtresses est effectuée automatiquement par le système après que l'auteur ait saisi tous les paramètres quantitatifs du module d'enseignement.

5.3 Les connaissances relatives à l'apprenant

Après avoir présenté les connaissances relatives au module d'enseignement, nous nous intéressons dans cette section aux connaissances relatives à l'apprenant. Afin de pouvoir adapter le cours à un apprenant, le système doit disposer d'une représentation interne des connaissances de cet apprenant. Cette représentation est appelée « modèle de l'apprenant » et le processus permettant sa création et sa mise à jour est dit « modélisation de l'apprenant » ou « diagnostic de l'apprenant » [Wenger 87].

Cette activité de modélisation des « connaissances de l'apprenant » est assurée, dans Hits, par le « Module de diagnostic de l'apprenant ». Ce module (Cf. **Figure 5.3**) utilise trois types de connaissances pour faire son diagnostic :

1. Les « règles instances d'évaluation » (RIEV) utilisées par « l'Évaluateur » et obtenues par instanciation des « règles maîtresses d'évaluation » (RMEV) ;
2. Les « règles instances d'acquisition » (RIAC) utilisées par le « Déducteur des acquis » (DAC) et obtenues par instanciation des « règles maîtresses d'acquisition » (RMAC) ;
3. Les connaissances mêmes de l'apprenant organisées selon une structure d'arbre, (le « modèle de l'apprenant ») gérée par le « Gestionnaire du Modèle de l'Apprenant » (GMA).

En commençant par le « modèle de l'apprenant », nous détaillerons ci-dessous, ces trois types de connaissances.

5.3.1 Le modèle de l'apprenant

Dans le chapitre 2, nous avons discuté les différentes techniques et théories de modélisation de l'apprenant adoptées dans les STI. La méthode à recouvrement (overlay method) est une des méthodes les plus utilisées dans les systèmes auteurs à cause de son indépendance du domaine et sa facilité de mise en oeuvre. C'est la méthode que nous avons choisie, nous aussi, pour modéliser l'apprenant dans notre système. En effet, Hits considère les connaissances d'un apprenant (ce qu'il sait) comme un sous-ensemble de la connaissance de l'expert (l'auteur).

Pour notre système, les connaissances à transmettre à l'apprenant sont réparties à travers les uahs. Parmi ces uahs, certaines sont évaluables alors que d'autres ne le sont pas. Les uahs consistent en des objets pédagogiques repérés par des « identifiants » indiquant la position de l'uah dans la structure du « module d'enseignement ». L'apprenant est donc modélisé en utilisant ces « identifiants ».

Chaque fois que l'apprenant réussit à atteindre des objectifs opérationnels visés par une uah, l'identificateur de cette uah est ajoutée au modèle de l'apprenant. La représentation des connaissances dans ce modèle repose sur une structure d'arbre. Ce dernier, appelé « arbre d'acquisition » est initialement vide. Au fur et à mesure que des uahs sont acquises, des noeuds sont créés. L'arbre consiste en quatre niveaux de noeuds. En commençant par les feuilles, nous trouvons : les noeuds « identifiants-uah » puis les noeuds « identifiants-chapitre » puis les noeuds « identifiants-parties » et enfin le noeud racine désigné par l'identifiant « module ».

Ces noeuds ne sont d'ailleurs pas les seuls. Nous trouvons aussi les noeuds « pretest » et « posttest-module », « posttest-partie », « posttest-chapitre » qui se situent au même niveau que les noeuds « identifiants-uah ». Ces quatre noeuds sont créés quand l'apprenant réussit à ces différents tests (de pronostic et d'inventaire) prévus par l'auteur. Avec ces quatre autres noeuds, le système suppose donc que : « Un module d'enseignement n'est considéré totalement acquis que si le « pretest » est réussi, toutes les uahs sont acquises et les trois « posttest » sont réussis ».

Nous considérons qu'un chapitre est acquis (et donc les objectifs spécifiques sont atteints) si toutes les uahs du chapitre sont acquises et le « posttest-chapitre » est réussi. Nous considérons qu'une partie est acquise (et donc les objectifs généraux sont atteints) si tous les chapitres sont acquis et le « posttest-partie » est réussi. Les différentes possibilités de création des noeuds sont définies par les règles suivantes :

- Quand le pretest est réussi alors trois noeuds sont créés : le noeud « pretest » et deux noeuds vides, l'un se situe dans le niveau correspondant aux « identifiants-chapitres », l'autre dans le niveau correspondant aux « identifiants-parties » ;
- Quand une uah est acquise alors un noeud de type « identifiant-uah » est créé ;
- Quand toutes les uahs d'un chapitre sont acquises alors un noeud de type « identifiant-chapitre » est créé, il représente le père de tous ces uahs ;
- Quand tous les chapitres d'une partie sont acquis alors un noeud de type « identifiant-partie » est créé, il représente le père de tous ces chapitres ;
- Quand un posttest de module ou de partie ou de chapitre est réussi alors trois noeuds sont créés : le noeud de type « posttest-* » et deux noeuds vides comme pour le pretest ;
- Quand le pretest est réussi, toutes les parties du module sont acquises et les trois posttests sont réussis alors le noeud racine « module » est alors créé signifiant que tout le module d'enseignement est acquis par l'apprenant.

La (**Figure 5.14**) montre l'évolution du « modèle de l'apprenant » en prenant un exemple de module très réduit (un pretest, une seule partie formée d'un seul chapitre formé lui de deux uahs évaluables, un posttest-chapitre, un posttest-partie et un posttest-module).

En plus de la gestion du « modèle de l'apprenant » en mémoire, le GMA (Gestionnaire du Modèle de l'Apprenant) est aussi chargé de la gestion des échanges entre la mémoire centrale et la base de données de la plate-forme Ibn Sina sur le disque, lors des opérations de sauvegarde et de chargement de ce modèle. Deux fichiers sont alors utilisés pour mémoriser et restituer les connaissances des différents apprenants utilisant le STI : le fichier APRENANT contenant les informations signalétiques des apprenants et le fichier MODELAPR contenant l'image du modèle de l'apprenant en mémoire (i.e. les noeuds de l'arbre d'acquisition).

- APRENANT (nom-utilisateur, mot-de-passe, nom-et-prénom, date-du-premier-cours, niveau-de-départ, niveau-arrivée, etc.).
- MODELAPR (nom-utilisateur, mot-de-passe, liste-des-noeuds, date-dernière-session).

Lors de sa première session d'apprentissage, un nouvel apprenant se voit insérer dans ces deux fichiers. A la fin de cette session, son modèle (l'arbre d'acquisition) est sauvegardé dans l'enregistrement qui lui correspond dans MODELAPR. Dans une prochaine session du même apprenant, cet enregistrement est rechargé en mémoire, l'arbre d'acquisition est reconstruit et

l'identifiant du dernier niveau acquis est envoyé au « Module tutoriel ». Ce dernier, recevant ce message, se charge alors de planifier le reste de la formation.

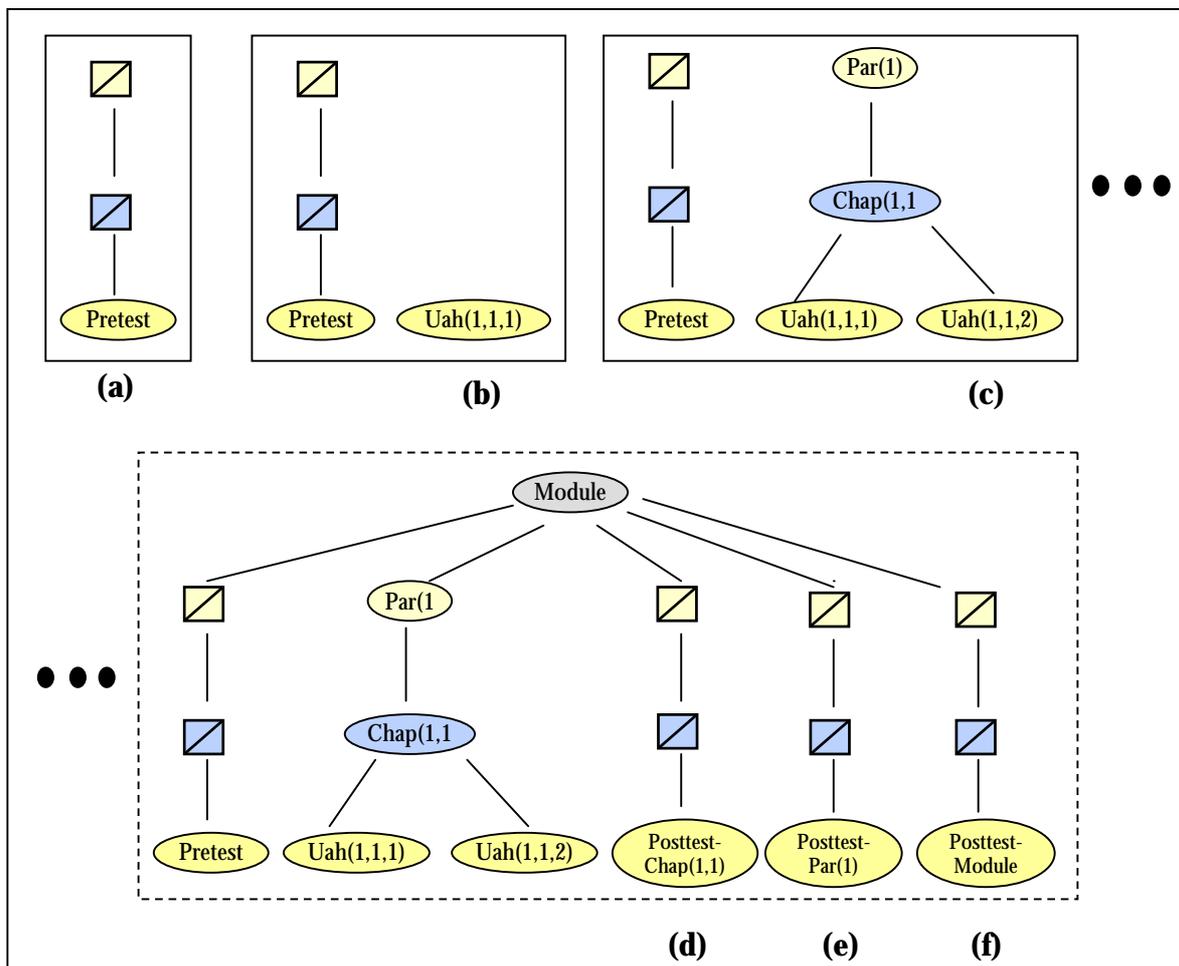
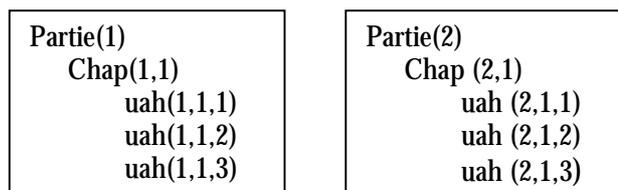


Figure 5.14. Exemple d'évolution d'un modèle de l'apprenant

5.3.2 Les règles d'acquisition (RIAC)

Ces règles complètent les règles de négociation du point d'entrée dans le cours (Cf. section concernant les connaissances tutorielles ci dessus). Elles ne sont utilisées que dans le cas où le système a accepté la sollicitation d'un apprenant de commencer son étude à partir d'une uah autre que la première. Leur but est de mettre à jour le modèle de l'apprenant, c'est à dire construire l'arbre d'acquisition décrit plus haut. Cette mise à jour est faite en mettant à « acquis » toutes les uahs jugées acquises qui précèdent l'uah de commencement.

Supposons, à titre d'exemple, que le module d'enseignement est composé de deux Parties contenant un Chapitre chacune et chaque Chapitre contenant trois uahs chacun :



Le choix du point d'entrée ne pouvant se faire au maximum que par l'uah(2,1,3), les « règles instances d'acquisition » qui seront générées par le DAC pour ce cours seront alors les suivantes :

RIAC1	:	Si	acquis-uah (2,1,2)	Alors	acquis-uah (2,1,1)
RIAC2	:	Si	acquis-uah (2,1,1)	Alors	acquis-uah (1,1,3)
RIAC3	:	Si	acquis-uah (1,1,3)	Alors	acquis-uah (1,1,2)
RIAC4	:	Si	acquis-uah (1,1,2)	Alors	acquis-uah (1,1,1)
RIAC5	:	Si	acquis-uah (1,1,3)	Alors	acquis-chap (1,1)

Cette base de règles, générée par instanciation des « règles maîtresses d'acquisition » (RMAC), est exploitée en chaînage avant par le DAC, donnant ainsi tous les noeuds de l'arbre d'acquisition. Supposons par exemple qu'un nouveau apprenant souhaite commencer son étude à partir de l'uah(2,1,2) et il a réussi au test de l'uah (2,1,1). Le DAC reçoit alors le message « acquis-uah(2,1,1) » envoyé par le « Négociateur » Ce message porte déjà en lui le premier noeud de l'arbre d'acquisition : « uah(2,1,1) ». A partir de ce message, le DAC déduit les faits suivants :

acquis-uah(1,1,3), acquis-uah(1,1,2), acquis-uah(1,1,1), acquis-chap(1,1), acquis-par(1), réussi-pretest.
--

Les « identifiants » qui seront alors envoyées au GMA pour construire l'arbre d'acquisition comme expliqué ci dessus sont les suivantes :

uah(2,1,1), uah(1,1,3), uah(1,1,2), uah(1,1,1), chap(1,1), par(1) et pretest
--

5.3.3 Les règles d'évaluation (RIEV)

Ces règles ont pour but d'évaluer l'apprenant. Avec leur interpréteur, elles forment ce que nous avons appelé l'«Evalueur». Ce dernier est invoqué par le « Module tutoriel » dans trois cas :

- Pour effectuer le pretest du module d'enseignement ;
- Pour tester la compréhension d'une uah évaluable ;
- Pour effectuer les trois posttests : posttest-chapitre, posttest-partie et posttest-module.

Un test d'évaluation, dans chacun de ces trois cas, consiste en un certain nombre de questions et/ou d'exercices formulés sous forme de QCM soigneusement élaboré par les auteurs. Devant chaque épreuve du test (i.e. question/exercice), l'apprenant est alors sollicité pour choisir des réponses/solutions qui lui semblent correctes. A chaque épreuve sont associées au moins deux réponses/solutions pondérées par l'auteur sur l'intervalle [0,20] pour exprimer différents types d'erreurs. Aussi, un « message d'appréciation » est associé à chaque réponse/solution.

Ceci diffère légèrement des QCM traditionnels dans lesquels l'enseignant donne une ou plusieurs réponses justes accompagnée d'un certain nombre de distracteurs (i.e. réponses complètement fausses). Ces QCM traditionnels restent toutefois compatibles avec notre approche puisque l'auteur est libre de pondérer les réponses. Il peut donc attribuer une note de 20 pour une réponse et une note de 0 pour tous les autres distracteurs.

Notre approche de pondération des réponses/solutions est donc plus générale dans la mesure où elle permet à l'auteur de varier les types d'erreurs à travers des réponses/solutions partiellement correctes. Une pondération de 15/20 associée à une réponse/solution, par exemple, signifie que la réponse/solution n'est pas totalement juste. Elle contient certainement, certains types d'erreurs qui, selon l'appréciation de l'auteur, ne sont pas aussi graves que ceux qu'il a inclus dans une réponse/solution pondérée 07/20.

Par le biais du « message d'appréciation » que l'auteur associe également à une réponse/solution, l'apprenant est informé sur les types d'erreurs qu'il a commis en ayant choisi une réponse/solution qui n'est pas totalement juste. Comme remédiation, il est orienté pour revoir certaines uahs du cours acquises auparavant.

Du moment que les réponses/solutions sont pondérées, l'« Evalueur » peut donc par simple calcul de moyenne des notes associées aux différentes réponses/solutions choisies par l'apprenant, décider si le test est réussi ou non. Nous rappelons que dans TalHits, l'auteur est appelé à définir deux seuils : un seuil d'échec et un seuil de réussite pour chaque uah évaluable. Pour affirmer la réussite à un test, il faut que la moyenne des notes obtenues aux différentes épreuves soit \geq au seuil de réussite de l'uah.

Le déroulement d'un test par l'« Evalueur » est effectué en général en deux étapes :

1. La première étape, nécessitant la collaboration du « Module expert », est répété autant de fois qu'il y a d'épreuves (i.e. question, exercice, épreuve de pretest ou épreuve de posttest) constituant le test. Chaque épreuve est traitée en deux sous-étapes :

- Recherche et présentation des uahs d'évaluation concernée par l'épreuve;
 - Ouverture d'un dialogue avec l'apprenant en vue de lui permettre d'effectuer son choix de la réponse/solution qu'il lui semble convenir à l'épreuve. Une fois le choix effectué, un « message d'appréciation » associé à la réponse/solution est alors affiché, et le couple (identifiant-réponse/identifiant-solution, note obtenue) est enregistré dans les feuilles d'une structure d'arbre (l'« arbre d'évaluation ») qui va être complété dans la deuxième étape.
2. La deuxième étape consiste en l'opération d'évaluation proprement dite, c'est à dire la prise de décision concernant la réussite ou la non réussite au test. Cette décision est prise à la suite d'une complétion de l'« arbre d'évaluation » issu de la première étape. La racine de l'arbre représente la moyenne des notes obtenues à toutes les épreuves du test. La (Figure 5.15) montre un exemple de l'arbre d'évaluation d'une uah contenant deux questions accompagnées de quatre réponses chacune et deux exercices accompagnés de trois solutions chacun. Dans le cas d'un pretest ou d'un posttest, l'arbre d'évaluation n'aurait que deux niveaux, celui des solutions choisies des épreuves (les feuilles) et celui de « moy-pretest/moy-posttest » (la racine).

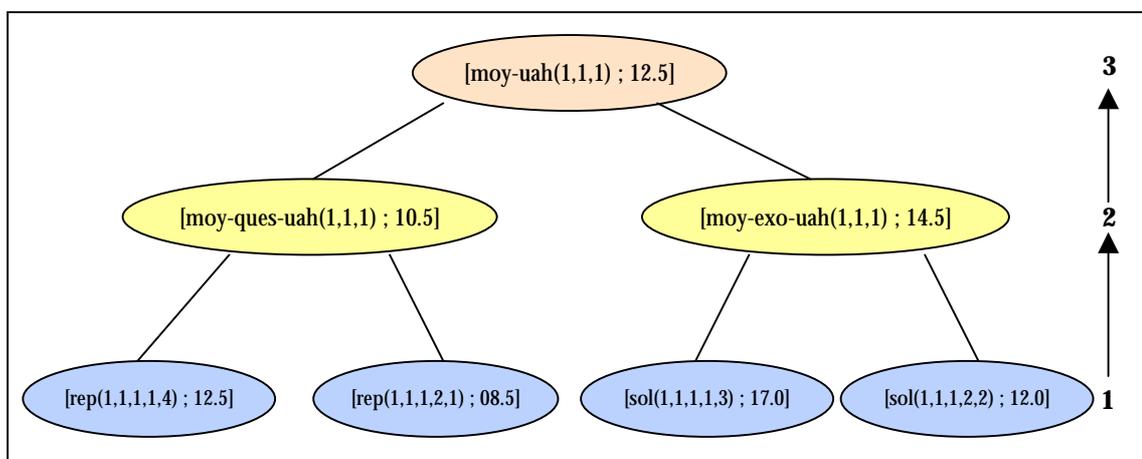


Figure 5.15. Un exemple d'arbre d'évaluation

Tous ces raisonnements effectués par l'« Evalueur » sont basés sur les règles instances d'évaluation (RIEV). Ces règles sont invoquées en chaînage arrière, avec la stratégie « en profondeur d'abord ». Le but à résoudre est le message reçu par l'« Evalueur ». D'après tout ce qui précède, ce message, envoyé par le « Module tutoriel », peut être l'un des cinq suivants: effectuer-pretest, tester-uah(i,j,k), effectuer-posttest-chap(i,j), effectuer-posttest-par(i), effectuer-

posttest-mod. Pour montrer comment est décrit le test de compréhension d'une uah évaluable par des règles, nous prenons l'exemple suivant :

Supposons qu'à l'uah(1,1,1) sont associés deux questions et un exercice. Supposons aussi que chacune des questions est accompagnée de trois réponses et que l'exercice est accompagné de deux solutions. Les règles nécessaires pour résoudre le but « tester-uah(1,1,1) » (en suivant la convention suivante) sont les suivantes :

Convention : La règle (exprimées en notation Prolog) (a :- b,c) se lit : résoudre le but 'a' revient à résoudre le sous but 'b' puis le sous but 'c'.

RIEV1:	tester-uah(1,1,1) :- présenter-épreuves-uah(1,1,1), évaluer-uah(1,1,1).
RIEV2:	présenter-épreuves-uah(1,1,1) :- poser-questions-uah(1,1,1), proposer-exercices-uah(1,1,1).
RIEV3:	poser-questions-uah(1,1,1) :- poser-ques(1,1,1,1), poser-ques(1,1,1,2).
RIEV4 :	poser-ques(1,1,1,1) :- présenter-ques(1,1,1,1), présenté-ques(1,1,1,1), suggérer-rep-ques(1,1,1,1), demander-choixrep-ques(1,1,1,1).
RIEV5:	suggérer-rep-ques(1,1,1,1) :- présenter-rep(1,1,1,1,1), présenté-rep(1,1,1,1,1), présenter-rep (1,1,1,1,2), présenté-rep (1,1,1,1,2), présenter-rep(1,1,1,1,3), présenté-rep(1,1,1,1,3).
RIEV6:	poser-ques(1,1,1,2) :- présenter-ques(1,1,1,2), présenté-ques(1,1,1,2), suggérer-rep-ques (1,1,1,2), demander-choixrep-ques(1,1,1,2).
RIEV7:	suggérer-rep-ques(1,1,1,2) :- présenter-rep(1,1,1,2,1), présenté-rep(1,1,1,2,1), présenter-rep(1,1,1,2,2), présenté-rep(1,1,1,2,2), présenter-rep(1,1,1,2,3), présenté-rep(1,1,1,2,3).
RIEV8:	proposer-exercices-uah(1,1,1) :- proposer-exo(1,1,1,1).
RIEV9:	proposer-exo(1,1,1,1) :- présenter-exo (1,1,1,1), présenté-exo(1,1,1,1), suggérer-sol-exo(1,1,1,1), demander-choixsol-exo (1,1,1,1).
RIEV10:	suggérer-sol-exo(1,1,1,1) :- présenter-sol(1,1,1,1,1), présenté-sol(1,1,1,1,1), présenter-sol(1,1,1,1,2), présenté-sol(1,1,1,1,2).

Selon la stratégie en « profondeur-d'abord », la résolution du but « tester-uah(1,1,1) » par le moteur d'inférence, devrait s'effectuer en résolvant un ensemble de sous-buts hiérarchisés. Sans qu'il soit exhaustif, le chemin de résolution est comme suit:

[tester-uah(1,1,1), présenter-épreuves-uah(1,1,1), poser-questions-uah(1,1,1), poser-ques(1,1,1,1), présenter-ques(1,1,1,1), présenté-ques(1,1,1,1), suggérer-rep-ques(1,1,1,1), présenter-rep(1,1,1,1), . . . , présenté-rep(1,1,1,1,3), demander-choixrep-ques(1,1,1,1), poser-ques(1,1,1,2), . . . , évaluer-uah(1,1,1)]

Les faits terminaux (i.e. qui n'apparaissent dans aucune partie gauche d'une règle) ont la signification suivante :

- le fait « présenter-<identifiant-uah> » est le message envoyé par l'« Evalueur » au « Module expert » pour afficher l'uah en question ;

- le fait « présenté-<identifiant-uah> » est le message renvoyé par le « Module expert » après l'affichage de l'uah ;
- le fait « demander-choix... » est le sélecteur d'une procédure qui dialogue avec l'apprenant en vue de choisir une réponse/solution parmi celles proposées ;
- le fait « évaluer-uah(1,1,1) » est le sélecteur d'une procédure qui évalue l'apprenant et retourne au « Module tutoriel ». Ce retour est supporté par l'un des deux messages : « acquis-uah(1,1,1) » ou « non(acquis-uah(1,1,1)) ». Dans le premier cas, le message est aussi envoyé au GMA pour mettre à jour le modèle de l'apprenant.

La même chose peut être obtenue pour résoudre le but « effectuer-pretest » ou le but « effectuer-posttest-* » où le signe * désigne (*module, partie ou chapitre*). La seule différence réside dans le fait que ces tests ne contiennent qu'un seul type d'épreuves. Ceci donne lieu à un nombre plus réduit de faits intermédiaires, comme celles de (RIEV2) dans l'exemple ci-dessus, et un nombre plus réduit de règles en général.

5.4 Les connaissances tutorielles

Les connaissances tutorielles forment la base de connaissances du « module tutoriel ». Elles manipulent les connaissances relatives au domaine d'enseignement et celles relatives à l'apprenant pour la gestion de son apprentissage. Comme pour les connaissances relatives au domaine et celles relatives à l'apprenant, notre objectif ici est de représenter des connaissances tutorielles indépendantes des contenus de la matière à enseigner..

Pendant les sessions d'apprentissage, le « module tutoriel » construit dynamiquement le cursus en déterminant au fur et à mesure les meilleurs OPs à proposer à l'apprenant. Il s'appuie pour cela sur les résultats de négociation avec l'apprenant, le niveau de départ représentant les acquis de ce dernier et un niveau cible représenté par les OPs ciblés dans la session d'apprentissage. Les uah sont choisies en fonction de caractéristiques décrivant les besoins à une étape donnée.

Dans le chapitre 1, nous avons décrit trois stratégies de conduite de dialogue qui sont implantées dans la plupart des tuteurs intelligents :

- le « dialogue socratique » qui consiste à mener l'activité d'enseignement sous forme de questions-réponses entre le système et l'apprenant;
- le « guidage discret » qui consiste à laisser l'apprenant agir, observer ce qu'il fait et n'intervenir qu'à certains moments bien choisis pour enseigner une notion;

- la « planification pédagogique » qui consiste à planifier l'enseignement à l'aide d'objectifs pédagogiques, c'est à dire établir une séquence d'étapes que le STI doit suivre pour satisfaire les besoins de l'apprenant pendant la session d'apprentissage.

Les deux premières approches nécessitent généralement une représentation explicite des connaissances du domaine pour mener le dialogue. La dernière approche cependant, peut très bien convenir aux systèmes manipulant des uah, comme le nôtre. C'est cette méthode que nous avons donc adoptée dans Hits. Nous représentons les connaissances tutorielles sous forme de « plans d'action » à exécuter par le système. Ces plans tiennent compte du point d'entrée dans le cours, des liens de prérequis entre les uah, les OPs et les concepts ainsi que du niveau d'instruction atteint par l'apprenant. Ils sont formalisés sous forme de règles dans lesquelles :

- la partie « condition » représente une situation dans laquelle peut se trouver l'apprenant (par exemple : acquis-uah (2,1,1)) ;
- la partie « conclusion » représente l'action à exécuter par le système dans une telle situation (par exemple : présenter-uah(2,1,2)).

Deux types de connaissances sont utilisés par le « module tutoriel » :

1. les « règles instances de négociation » (RINEG) utilisées par le « négociateur » ;
2. les « règles instances de séquençement » (RISEQ) utilisées par le « planificateur » ;

5.4.1 Les connaissances de négociation

Afin de permettre au système de générer automatiquement une leçon adaptée à chaque apprenant, une session d'apprentissage débute toujours par une négociation du « *type de parcours* » et le « *point d'entrée* » dans le cours. L'apprenant possède trois choix de *type de parcours* : parcours basé sur les composants, parcours basé sur les objectifs pédagogiques et parcours basé sur les concepts. Cependant, la décision de démarrer la formation par un certain niveau (partie, chapitre, uah, objectif pédagogique, concept) est conditionnée par l'acquisition (i.e. compréhension) du niveau qui lui est prérequis.

Des évaluations de prérequis doivent donc être effectuées avant de satisfaire le désir de l'apprenant. Aussi le système permet à l'apprenant de négocier les endroits où il veut subir les tests au cours de sa formation : en fin du module seulement, en fin des parties seulement, en fin des chapitres seulement, en fin des uah seulement ou n'importe quelle combinaison de ces endroits. Le déroulement d'une négociation entre l'apprenant et le « négociateur » est effectué en général en trois étapes :

1. Le « négociateur » invite l'apprenant à choisir une uah, un OP ou un concept à partir de laquelle (ou duquel) il souhaite commencer sa formation ;

2. le « négociateur » vérifie la maîtrise des prérequis de l'uah, de l'OP ou du concept demandé(e). Cette tâche nécessitant la collaboration de l'« évaluateur » et du « module expert », consiste à effectuer des tests sur les prérequis de l'uah choisie ou des uah qui portent sur l'OP ou le concept choisi. Si le résultat est positif alors l'étude commencera à partir du niveau demandé sinon le « négociateur » remonte de plus en plus en arrière (jusqu'à la première uah de module s'il le faut) pour déterminer le vrai niveau de compréhension de l'apprenant. Le résultat du processus de négociation est un « fait » qui indique la référence de l'uah acquise dans la série de tests effectués. Ce fait constitue le message à envoyer au DAC et au « planificateur ». ;

3. le message issu de l'étape 2 est d'abord envoyé au DAC pour mettre à jour le modèle de l'apprenant, puis, au « planificateur » pour planifier le séquençement des uah à présenter à l'apprenant pendant la session d'apprentissage.

Tous ces raisonnements effectués par le « négociateur » sont basés sur les règles instances de négociation qu'il invoque en chaînage avant. Afin de montrer un exemple de telles règles et comment s'effectue le raisonnement sur ces règles, nous supposons la structure d'un module d'enseignement très réduit (**Figure 5.16**) et nous considérons un parcours basé sur les uah. Nous supposons en outre que l'uah (2,1,1) n'est pas évaluable (celle marquée par **) :

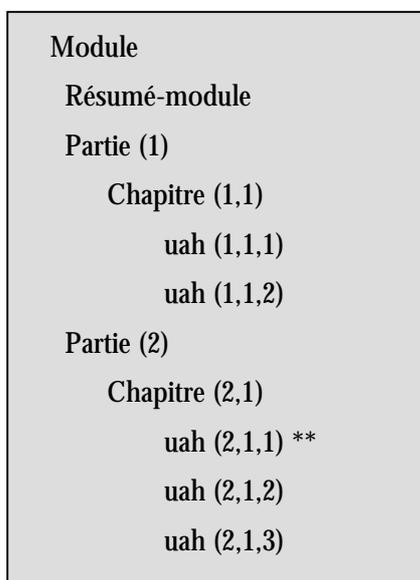


Figure 5.16. Un exemple de structure d'un cours

Les règles de négociation qui seront générées par le système pour un cours structuré de cette manière sont alors comme suit :

RINEG1	<u>Si</u> commencer-uah(2,1,3)	<u>Alors</u>	tester-uah(2,1,2)
RINEG2	<u>Si</u> commencer-uah(2,1,2)	<u>Alors</u>	tester-uah(1,1,2)
RINEG3	<u>Si</u> commencer-uah(2,1,1)	<u>Alors</u>	tester-uah(1,1,2)
RINEG4	<u>Si</u> commencer-uah(1,1,2)	<u>Alors</u>	tester-uah(1,1,1)
RINEG5	<u>Si</u> commencer-uah(1,1,1)	<u>Alors</u>	démarrage-normal
RINEG6	<u>Si</u> non(acquis-uah(2,1,2))	<u>Alors</u>	tester-uah(1,1,2)
RINEG7	<u>Si</u> non(acquis-uah(1,1,2))	<u>Alors</u>	tester-uah(1,1,1)
RINEG8	<u>Si</u> non(acquis-uah(1,1,1))	<u>Alors</u>	démarrage-normal

A titre d'exemple, si un apprenant souhaite commencer par l'uah(2,1,2), alors un test lui sera effectué sur l'uah(1,1,2) puisque l'uah(2,1,1) n'est pas évaluable (règle 2). Le « négociateur » envoie à « l'évaluateur » le message « tester-uah(1,1,2) » et attend la réponse. Si le test est négatif, « l'évaluateur » envoie le message « non(acquis-uah(1,1,2)) », ce qui permet au « négociateur » de continuer le raisonnement en utilisant la règle 7. Celle-ci propose alors un test sur l'uah (1,1,1). Si par contre le test est positif, le « négociateur » ne reçoit aucune réponse et donc s'arrête de fonctionner. Grâce au message « acquis-uah(1,1,2) », « l'évaluateur » donne la main dans ce dernier cas, d'abord au DAC pour mettre à jour le modèle de l'apprenant puis au « planificateur » pour décider de l'action suivante à exécuter.

Dans le cas où l'apprenant choisit un parcours basé sur les OPs ou bien sur les concepts le même raisonnement est effectué pour déterminer l'uah de démarrage en se basant sur les liens de rattachement des OPs/concepts aux uah.

5.4.2 Les connaissances de gestion du séquençement des uah

Une fois l'étape de positionnement effectuée, le système passe à la gestion du séquençement des uah pour atteindre les objectifs fixés pour la session d'apprentissage. Nous rappelons que nous avons prévu trois types de parcours de la matière d'enseignement. Le parcours basé sur les composants utilise les liens implicites de prérequis entre uah, c'est à dire le parcours d'arbre. Le parcours basé sur les OPs utilise le réseau de prérequis des OPs pour déterminer la prochaine uah à présenter à l'apprenant. Le parcours basé sur les concepts utilise, lui, le réseau de prérequis des concepts.

Le déroulement de la session d'apprentissage est assuré par le « planificateur » qui se base sur un ensemble de règles. Ce système expert est invoqué par le système pour décider de l'action suivante à entreprendre après qu'une phase d'apprentissage vient d'être franchie. Par exemple décider de ce qui doit être fait après que l'uah(3,2,1) vient d'être testée avec succès. Une phase d'apprentissage consiste à enchaîner une ou plusieurs uah appropriées en réponse à une situation

pédagogique donnée. Le « mode formation » que nous avons prévu dans Hits s’articule autour de trois phases d’apprentissage, classiques en enseignement, à savoir : la « phase de présentation de connaissances », la « phase d’évaluation de connaissances » et la « phase de remédiation ».

- **La phase de présentation de connaissances** consiste en la recherche et l’affichage d’une uah de cours pour apporter de nouvelles connaissances à l’apprenant. Elle est assurée par le « module expert ». Cette phase est déclenchée par le « planificateur » en envoyant au moment voulu le message « présenter-<identifiant d’uah> » au « module expert ». Une fois la présentation terminée, ce dernier retourne au « planificateur » le message « présenté-<identifiant d’uah> ». Ce message représente alors une nouvelle situation pédagogique à laquelle le « planificateur » doit réagir : présenter d’autres connaissances ou passer à une évaluation.
- **La phase d’évaluation** est assurée par une coordination entre « l’évaluateur » et le « module expert ». Elle consiste en la recherche et l’affichage des uah d’évaluation formatives relatives aux uah de cours évaluables ou bien dans le cas de pretest et des posttests. Dans cette phase l’apprenant est actif, il peut s’exercer et essayer de s’approprier et fixer les connaissances théoriques tant qu’il le désire. La phase d’évaluation est déclenchée par le « planificateur » en envoyant l’un des messages suivant à « l’évaluateur » : « effectuer-pretest », « tester-uah(i,j,k) » ou « effectuer-posttest-* ». Une fois l’évaluation terminée en collaboration avec le « module expert », ce dernier renvoie au « planificateur » l’un des messages suivants : « pretest-réussi », « non(pretest-réussi) », « acquis-uah(i,j,k) », « non(acquis-uah(i,j,k)) », « posttest-mod-réussi », « non(posttest-mod-réussi) », « posttest-par(i)-réussi », « non(posttest-par(i)-réussi) », « posttest-chap(i,j)-réussi », « non(posttest-chap(i,j)-réussi) ». Le message retourné représente alors pour le « planificateur » une nouvelle situation pédagogique qui doit déclencher une réaction qui peut être une remédiation ou une présentation de nouvelles connaissances.
- **La phase de remédiation** est activée suite à la phase d’évaluation et si les résultats obtenus par l’apprenant ne sont pas satisfaisants. L’objectif de cette phase est d’amener l’apprenant à améliorer ses résultats et essaye de chercher dans la mesure du possible les causes de la non réussite de l’apprenant et de la traiter en exploitant les différents types de liens qui existent entre les OP, les uah et les concepts. Elle active généralement les uah de remédiation qui sont conçus en liaison étroite avec les uah d’évaluation mais elle peut aussi activer les phases de présentation de connaissances sous forme d’uah ou bien à choisir des

extraits de granularités plus faibles des uah de type « définition », « théorème », « exemple », etc. pour combler les lacunes de l'apprenant. Pour cette raison, nous travaillons en collaboration avec notre collègue Behaz Amel qui, elle, traite les unités pédagogiques d'apprentissage à un niveau de granularité plus petit [Behaz et Djoudi 05]. La phase de remédiation est déclenchée par le « planificateur » en envoyant le message suivant au « module expert » : `remédier-uah(i,j,k)` où `uah(i,j,k)` est l'uah dont le système a jugé que ses objectifs ne sont pas atteints. Selon les types d'erreurs commis et les différents types de liens, le « module expert » choisira les uah à revoir. Le processus de remédiation s'arrête quand les objectifs de l'uah de cours sont jugés atteints. Le « module expert » retourne alors au « planificateur » le message « `remédié-uah(i,j,k)` » et ce dernier se chargera alors de présenter les connaissances appropriées selon le plan adopté.

Finalement, une session d'enseignement/apprentissage est donc un processus se déroulant progressivement selon les trois phases de présentation de connaissances, d'évaluation et de remédiation. Dans la première phase, le système se charge uniquement de la présentation des uah de cours en explorant les quatre premiers niveaux de l'univers des uah. Dans la deuxième phase, il se charge de la présentation des uah d'évaluation associées aux uah évaluables, en explorant les deux derniers niveaux de l'univers des uah. Dans la troisième phase, il se charge de la présentation des uah de remédiation dans le cas où les objectifs fixés pour une uah de cours ne sont pas atteints. Il se pose alors le problème de la manière d'explorer cet univers, c'est à dire comment choisir l'uah suivante à présenter après qu'une uah vient d'être présentée. Dans Hits, cette décision de choix est prise tout simplement en suivant les liens de prérequis explicites entre les OP et entre les concepts et les liens implicites entre les uah dans l'univers des uah.

Pour une formation complète d'un apprenant, et si ce dernier choisira par exemple le type de parcours basé sur les composants, cela revient à explorer tout l'univers des uah avec la stratégie « en profondeur d'abord ». Les « règles instances de séquençement » (RISEQ) et les "règles d'évaluation" (RIEV) sont conçues spécialement pour assurer ce type d'exploration.

Les premières explorent les quatre premiers niveaux de l'arbre, les secondes, quant à elles, explorent ses deux derniers niveaux. Les « règles d'évaluation » sont déjà décrites au (paragraphe 5.3.3). Les « règles de séquençement » sur lesquelles repose le raisonnement du « planificateur », sont invoquées en chaînage avant. Afin de montrer quelques exemples de telles règles et comment est effectué le raisonnement à travers ces règles, nous reprenons la structure du cours de la (**Figure 5.16**) ci dessus et nous supposons qu'il y a un pretest et un posttest. Les règles

instances de séquençement qui seront générées par le système pour cette univers d'uah ont la forme suivante :

RISEQ1	Si	(démarrage-normal) ou (acquis-rien)	Alors	présenter-résumé-module
RISEQ2	Si	présenté- résumé-module	Alors	effectuer-pretest
RISEQ3	Si	réussi-pretest	Alors	présenter-résumé-par(1)
RISEQ4	Si	non(réussi-pretest)	Alors	arrêt-avec-regret
RISEQ5	Si	présenté-résumé-par(1)	Alors	présenter-résumé-chap(1,1)
RISEQ6	Si	présenté-résumé-chap(1,1)	Alors	présenter-uah(1,1,1)
RISEQ7	Si	présenté-uah(1,1,1)	Alors	tester-uah(1,1,1)
RISEQ8	Si	acquis-uah(1,1,1)	Alors	présenter-uah(1,1,2)
RISEQ9	Si	non(acquis-uah(1,1,1))	Alors	remedier-uah(1,1,1)
RISEQ10	Si	présenté-uah(1,1,2)	Alors	tester-uah(1,1,2)
RISEQ11	Si	acquis-uah(1,1,2)	Alors	acquis-chap(1,1)
RISEQ12:	Si	non(acquis-uah(1,1,2))	Alors	remedier-uah(1,1,2)
RISEQ13:	Si	acquis-chap(1,1)	Alors	acquis-par(1)
RISEQ14:	Si	acquis-par(1)	Alors	présenter-résumé-par(2)
...				
RISEQ17	Si	présenté-uah(2,1,1)	Alors	acquis-uah(2,1,1)
...				
RISEQ23	Si	acquis-uah(2,1,3)	Alors	acquis-chap(2,1)
RISEQ24	Si	acquis-chap(2,1)	Alors	acquis-par(2)
RISEQ25	Si	acquis-par(2)	Alors	effectuer-posttest
RISEQ26	Si	réussi-posttest	Alors	acquis-module
RISEQ27	Si	non(réussi-posttest)	Alors	effectuer-posttest
RISEQ28	Si	acquis-module	Alors	arrêt-avec-félicitation

Avec cet ensemble de règles, nous pouvons imaginer beaucoup de situations lors du démarrage d'une nouvelle session de travail. C'est le premier message reçu par le « planificateur » qui lui indique le point où en est la formation et donc la première règle à déclencher. Ce message peut provenir de deux manières :

1. Dans le cas d'un nouvel apprenant ou un ancien apprenant qui veut choisir à nouveau le point d'entrée dans le module, le message provient du « négociateur » (par exemple « démarrage-normal », « acquis-uah(1,1,2) », etc.).

2. Dans le cas d'un ancien apprenant qui veut poursuivre son apprentissage du point où il l'a arrêtée dans la session précédente, le GMA envoie au « négociateur » un message qui indique la dernière uah acquise par l'apprenant dans la session précédente ou du message « acquis-rien » pour un apprenant qui n'a même pas encore réussi au prétest.

Comme il apparaît dans les règles (RISEQ), la plupart des faits apparaissant dans la partie « conclusion » sont des problèmes à résoudre envoyés aux autres modules pour les traiter. Les faits apparaissant dans la partie « condition » sont les réponses renvoyées par ces modules. Voici par ailleurs la liste de ces échanges :

1. Les messages « présenter-<identifiant d'uah> » et « remédier-<identifiant d'uah> » sont à envoyer au « module expert » ;
2. Les messages « présenté-<identifiant d'uah> » sont les messages renvoyés par le « module expert » ;
3. Les messages « effectuer-prétest », « effectuer-posttest » et « tester-uah(i,j,k) » sont envoyés à « l'évaluateur » ;
4. Les messages « réussi-prétest », « non(réussi-pretest) », « réussi-posttest » « non(réussi-posttest) », « acquis-uah(i,j,k) » et « non(acquis-uah(i,j,k)) » sont les messages renvoyés par « l'évaluateur ». Ceux confirmant la réussite au test peuvent aussi provenir du GMA comme premier message au « planificateur » lors du démarrage d'une nouvelle session de travail ;
5. Les messages « acquis-chap(i,j) », « acquis-par(i) » et « acquis-uah(i,j,k) » sont envoyés au GMA ;
6. Le message « démarrage-normal » est envoyé par le « négociateur » ;
7. Le message « acquis-rien » est envoyé par le GMA ;
8. Les deux faits « arrêt-avec-regret » et « arrêt-avec-félicitation » sont les sélecteurs de deux procédures qui dialoguent avec l'apprenant en vue d'achever la session de travail.

5.4.3 Génération automatique du curriculum

En fonction de ses connaissances précédentes et des différents réseaux de prérequis conçus par l'auteur, chaque apprenant peut suivre un cours adapté à lui. Un cours dans Hits est composé essentiellement d'un ensemble d'objectifs pédagogiques et d'un curriculum.

Le curriculum est une liste ordonnée d'uah construite par le système pour fournir à un apprenant particulier les connaissances nécessaires pour atteindre les objectifs visés. Dans le cas d'un parcours basé sur les composants, la génération du curriculum se fait par parcours de l'univers des uah en profondeur d'abord en commençant par la racine. Dans les deux autres cas, la génération automatique du curriculum est effectuée selon les trois étapes suivantes :

- Une liste ListOp des OPs à atteindre est construite sur la base des choix de l'apprenant lors de la phase de négociation et/ou sur la base du contenu du modèle de l'apprenant. Si l'apprenant a choisi un parcours basé sur les concepts, une liste des concepts à apprendre ListCp est d'abord formée. Pour chaque concept appartenant à cette liste, les OPs qui lui sont attachés sont ajoutés à la liste ListOp.;
- La liste ListOp est ensuite transformée en une liste ordonnée d'uah de cours ListUah contenant la meilleure séquence des uah conçus pour atteindre les objectifs visés. Le

séquencement se base sur le réseau de prérequis des OPs et sur le contenu du modèle de l'apprenant ;

- La liste ListUah est enfin transformée en un curriculum en ajoutant les uah d'évaluation et de remédiation nécessaires à la mise à jour du modèle de l'apprenant.

5.5 Le module de communication interpersonnelle

Dans le contexte de l'enseignement à distance, les apprenants et tuteurs ne se connaissent généralement pas et sont géographiquement isolés pendant presque toute la formation. L'acte d'apprentissage est donc réputé difficile car les personnes sont sans lien social entre elles. L'espoir d'amélioration de cette situation tient dans l'utilisation de dispositifs technologiques de communication pour constituer une situation d'apprentissage coopérative en groupe.

L'apprentissage coopératif assisté par ordinateur (ACAO ou CSCL en anglais) est devenu un domaine de recherche à part entière et beaucoup de conférences s'organisent à travers le monde actuellement pour échanger les idées, les expériences et les résultats auxquels sont arrivés les chercheurs [Uquebec 06]. Nous n'avons donc pas la prétention de se jeter dans cet immense domaine de recherche qui dépasse le cadre de cette thèse, mais nous voulons seulement intégrer dans notre environnement deux outils de communication devenus classiques à ce jour que sont la messagerie électronique et le forum de discussion. Ces deux outils permettent aux usagers de communiquer d'une manière asynchrone et donc de coopérer entre eux et s'échanger des idées concernant la matière d'apprentissage.

5.5.1 Messagerie électronique

La messagerie électronique (email) est un dispositif de communication différé dans le temps entre plusieurs personnes. C'est un système qui permet aux utilisateurs de s'échanger des messages par le biais d'un réseau comme l'Internet. Les messages envoyés sur Internet prennent généralement la forme de textes. Avec un bon kit de messagerie, on peut aussi rattacher d'autres types de fichiers aux messages. Ces fichiers joints peuvent être des images, des vidéos, des fichiers son, des documents et même des logiciels.

Une fois muni d'une adresse e-mail et d'une connexion à l'Internet, on peut envoyer des messages à une personne en utilisant son adresse email. Les programmes de messagerie sont presque tous conçus sur le même modèle que nous avons adopté.

Lorsqu'on ouvre une fenêtre pour composer un nouveau message, on doit fournir certaines informations dans le formulaire affiché pour être certain que le message arrivera à sa bonne destination :

- **A** : C'est la zone dans laquelle on doit taper l'adresse email de la personne à qui on veut envoyer le message.
- **CC (Ou Copie à)** : C'est la zone dans laquelle on doit taper les adresses email des personnes à qui on veut également envoyer une copie du message.
- **Objet** : C'est la zone dans laquelle on doit taper un en-tête qui indiquera au destinataire l'objet du message.

Le reste du formulaire est destiné à contenir le message. Selon la sophistication des programmes de messagerie on trouvera également beaucoup de menus et de boutons ayant des fonctions variées dont le bouton d'attachement d'autres types de fichiers par exemple.

Une fois le message envoyé en appuyant sur le bouton « envoyer », le message est routé de serveur en serveur jusqu'à ce que le destinataire le télécharge sur sa machine pour le lire grâce à son programme de messagerie.

5.5.2 Forum de discussion

Les forums de discussion sont un excellent outil de collaboration asynchrone entre apprenants distants [Poitiers 06]. Les enseignants ont notamment constaté que les étudiants travaillant en réseau sont plus productifs et apprennent les uns des autres en s'échangeant des messages. Dans de nombreuses disciplines, on considère que les forums incitent à une participation accrue de tous les étudiants à des discussions qui vont au-delà d'un cours spécifique.

Un forum est un espace de discussion sur un site web qui permet aux internautes d'échanger des propos, de poser des questions ou des réponses sur un ou plusieurs thèmes définis. Les messages sont postés sur le site web dans l'espace réservé au forum. Certains forums sont libres, d'autres réclament de l'utilisateur à s'identifier (après enregistrement d'un compte) pour avoir le droit de poster un message. Le forum remplit une fonction similaire à celle d'un panneau d'affichage. On peut placer un nouveau message ou bien faire une remarque sur un message affiché. Les messages sont classés par thèmes. Chaque message traitant d'un nouveau thème est placé à la racine et chacun peut le commenter ou répondre aux interrogations de l'auteur.

Commencer un nouveau sujet : Pour commencer un nouveau sujet, l'apprenant choisit le bouton « Rédiger message » et édite son message dans le champ du formulaire destiné à cet effet.

Le sujet du message est tapé dans le champ « sujet ». Une fois envoyé, le message est stocké sur un serveur puis publié de manière à pouvoir être lu par les autres apprenants.

Répondre à un message : Pour donner un avis sur le contenu d'un message qui traite d'un sujet donné, l'apprenant choisit le bouton « Répondre » et édite son message dans le champ du formulaire destiné à cet effet. Une fois envoyée, la réponse sera stockée sur le serveur puis publiée de façon à permettre à tous les apprenants de la lire et donc d'en bénéficier des idées et des expériences des autres.

5.6 Déroulement d'une session d'apprentissage

Le déroulement d'une session d'apprentissage suit les étapes suivantes :

(E1) L'apprenant commence une session d'apprentissage par l'accès à Hits via n'importe quel navigateur standard tel que Internet explorer ou Mozilla FireFox.;

(E2) Le « coordonnateur » ouvre alors un dialogue pour demander à l'apprenant de préciser quel mode d'apprentissage il désire exécuter. Deux cas sont envisageables :

Premier cas (l'apprentissage libre)

Si l'apprenant choisit le mode d'apprentissage libre « exploration libre », le « coordonnateur » ouvre un dialogue pour demander à l'apprenant s'il est déjà inscrit ou non :

- Si l'apprenant n'est pas encore inscrit (i.e. nouvel apprenant), le « coordonnateur » passe le contrôle au GMA pour inscrire le nouvel apprenant puis il passe le contrôle au « module d'exploration libre » qui lui ouvre le chemin pour explorer le contenu du module librement ;
- Si l'apprenant est déjà inscrit dans le module (i.e. ancien apprenant), le « coordonnateur » passe le contrôle au « module d'exploration libre » et ce dernier offre à l'apprenant la possibilité d'explorer librement le contenu du module selon ses besoins.

Deuxième cas (l'apprentissage guidé)

Si l'apprenant choisit le « mode formation », le « coordonnateur » passe à l'étape **(E3)** ;

(E3) Le « coordonnateur » ouvre un autre dialogue pour demander à l'apprenant s'il est déjà inscrit ou non, là aussi, deux cas sont envisageables :

- si l'apprenant n'est pas encore inscrit (i.e. nouvel apprenant), le « coordonnateur » passe à l'étape **(E4)** ;
- si l'apprenant est déjà inscrit (i.e. ancien apprenant), le « coordonnateur » passe à l'étape **(E5)** ;

(E4) Le « coordonnateur » effectue, dans l'ordre, les tâches suivantes :

- i. il passe le contrôle au GMA pour saisir toutes les données nécessaires et inscrire le nouvel apprenant ;
- ii. il ouvre un dialogue pour demander à l'apprenant de préciser le point d'entrée dans le cours (partie, chapitre, uah, concept ou objectif pédagogique) à partir duquel il souhaite commencer sa formation, soit « uah(1,1,5) » par exemple ;
- iii. il passe le contrôle au « négociateur » en lui envoyant le message, dans notre exemple, « commencer-uah(1,1,5) ». Ce message constitue un problème qu'il convient aux modules « négociateur », « évaluateur » et « expert du domaine », de coopérer pour le résoudre. La résolution du problème consiste à effectuer des tests sur les prérequis de cet uah en commençant par tester l'uah(1,1,4)". Si ce dernier n'est pas acquis, le « négociateur » remonte encore de plus en plus en arrière (jusqu'à la première uah du cours s'il le faut) pour déterminer le vrai niveau de l'apprenant. Le résultat du processus de négociation est un « fait » qui indique le numéro de l'uah acquise dans la série de tests effectués. Supposons par exemple que c'est l'uah(1,1,3)". Le message à retourner au « coordonnateur » est donc « acquis-uah(1,1,3) ». La même procédure sera effectuée si l'apprenant a choisi de faire un type de parcours basé sur les concepts ou les objectifs pédagogiques et ce, en exploitant les liens qui les relient aux uah ;
- iv. il passe le contrôle au DAC en lui envoyant le message issu de la phase de négociation acquis-uah(1,1,3). Ce déducteur déduit la liste de toutes les uah qui sont considérées comme acquises selon les liens de prérequis entre concepts et objectifs pédagogiques définis par l'auteur ou bien les liens implicites entre les uah. Il s'agit dans notre exemple des uah(1,1,2) et uah(1,1,1). Par coopération avec le GMA, toutes ces uah ainsi que l'uah(1,1,3) seront rajoutées au modèle de l'apprenant ;
- v. il passe enfin le contrôle au « planificateur » en lui envoyant le même message issu de la phase de négociation. Sur la base de ce message, le « planificateur » planifie le reste de la session en établissant le séquençement des uah de cours à présenter et les uah d'évaluation pour effectuer les tests correspondants. Il travaille alors en collaboration avec les modules « expert du domaine », « évaluateur » et le GMA pour adapter le cours à chaque apprenant. Ce processus continue jusqu'à l'achèvement de « l'arbre des prérequis » ou jusqu'à la rencontre d'une requête de

suspension de la part de l'apprenant. Notons enfin que dans notre exemple, l'apprenant est contraint de commencer sa formation à partir de l'uah(1,1,4) et non pas à partir de l'uah(1,1,5) comme il l'a souhaité ;

- vi. à la fin de la session, il invoque le GMA pour sauvegarder le modèle de l'apprenant dans la base de données.

(E5) Le « coordonnateur » effectue, dans l'ordre, les tâches suivantes :

- i. il ouvre un dialogue pour demander à l'apprenant de préciser s'il continue son apprentissage à partir de la dernière uah acquise ou bien de préciser le point d'entrée dans le cours (partie, chapitre, uah, concept ou objectif pédagogique) à partir duquel il souhaite commencer sa nouvelle session. S'il choisit un autre point d'entrée, le coordonnateur suivra la même procédure que pour un nouvel apprenant mais en mettant à jour d'abord le modèle de l'apprenant.
- ii. Si l'apprenant décide de continuer son apprentissage à partir de la dernière uah acquise auparavant, le « coordonnateur » invoque le GMA pour charger ses données stockées dans le modèle de l'apprenant et déterminer la dernière uah acquise dans la session précédente, soit par exemple l'uah(1,1,2) ;
- iii. il passe le contrôle au « planificateur » en lui envoyant le message, dans notre exemple, « acquis-uah(1,1,2) ». Ce message sera utilisé par le « planificateur » pour déterminer le noeud de reprise dans « l'arbre des prérequis » en suivant soit les liens implicites entre uah soit les liens de prérequis entre les concepts et/ou ceux entre les objectifs pédagogiques. Dans notre exemple, ce noeud sera par exemple uah(1,1,3) et le processus d'apprentissage continue comme dans les étapes E4 (v.) et E4(vi.).

(E6) Pendant une session d'apprentissage un apprenant peut à tout moment, grâce au « module de communication », demander de l'aide à son enseignant ou à ses collègues en leur envoyant des messages. S'ils sont déjà en ligne les réactions peuvent être immédiates sinon les réponses peuvent lui parvenir ultérieurement quand ils se connectent. Le « module de communication » offre deux outils à l'apprenant pour coopérer avec des tiers : la messagerie électronique et le forum de discussion. Il lui offre également la possibilité de voir à tout moment quelles sont les personnes qui sont en ligne.

5.7 Conclusion

Dans ce chapitre, nous avons décrit l'architecture logicielle ainsi que les connaissances utilisées dans le mode apprenant Hits : les connaissances relatives au domaine, les connaissances relatives à l'apprenant et les connaissances tutorielles. Nous avons d'abord proposé un modèle basé sur la pédagogie par objectifs pour la structuration des connaissances du domaine. Ce modèle repose sur la notion d'uah et nous avons distingué quatre types d'uah : uah de présentation de connaissances sans évaluation, uah de fixation de connaissances avec évaluation, uah d'évaluation et uah de remédiation.

Sur la base de cette structure, nous avons défini par la suite, les connaissances permettant de modéliser l'apprenant en utilisant la technique de recouvrement. Les connaissances tutorielles sont enfin présentées. Toutes ces connaissances sont représentées sous forme de règles de production et sont divisées en cinq paquets. Chaque paquet de règles ayant une fonction précise, est invoqué par un moteur d'inférence. Une session d'enseignement/apprentissage est donc gérée par une coopération de cinq systèmes experts s'échangeant des messages à travers le « module coordonnateur ».

L'accès à l'apprentissage peut se faire selon trois modes adaptatifs : le mode de parcours basé sur les composants (l'apprenant sélectionne un composant de la structure d'arbre qui forme le module d'enseignement : partie, chapitre, uah), le mode de parcours basé sur les concepts (l'apprenant sélectionne un concept sur lequel il souhaite travailler et le système lui fournit le séquençement de composants satisfaisant ce concept) et le mode de parcours basé sur les objectifs pédagogiques (l'apprenant sélectionne un OP qu'il souhaite atteindre et le système lui fournit le séquençement de composants satisfaisant cet OP).

L'adaptation est faite via un langage de règles. Cela permet d'inférer des connaissances non explicites dans les différents modèles et de fournir l'adaptation du contenu en fonction des connaissances de l'apprenant et de ses préférences.

Les différents objets constituant le STI Hypermédia Hits étant décrits dans ce chapitre, nous présentons dans le chapitre suivant le mode auteur coopératif CamHits permettant d'introduire ces objets dans le système.



Chapitre 6

Mode auteur coopératif – architecture logicielle

6.1 Introduction

Malgré les efforts consentis ces deux dernières décennies, la réalisation de STI demeure une tâche difficile à entreprendre. Pour réussir, cette tâche exige souvent la collaboration entre équipes pluridisciplinaires constituées généralement de pédagogues, psychologues, experts de domaines et informaticiens. La difficulté de constitution de ces équipes en pratique, a néanmoins eu un effet réducteur quant à la productivité dans ce domaine. En vue d'améliorer la productivité et permettre à une communauté plus large de s'impliquer, les systèmes auteurs (voir chapitre 3) ont permis de développer plus ou moins aisément des STI. La tâche est alors réduite aux experts de domaines à faire couler des connaissances dans un canevas de STI générique prédéterminé par le système. Certains de ces systèmes auteurs sont discutés dans [Murray 99], cependant, ils ont tous été conçus pour fonctionner en mono-usager.

De nos jours, nous pensons que la disponibilité de ces systèmes ne résout pas totalement le problème. En effet, avec la complexité et l'interdépendance des sciences actuelles, la difficulté de construction d'un STI par un seul auteur reste toujours à l'ordre du jour. Ainsi donc, et même de point de vue introduction de connaissances seulement, cette construction nécessite souvent la coopération de divers experts du domaine en question. Etant évidemment géographiquement éloignés dans la plupart du temps, il est par conséquent nécessaire de mettre à leur disposition des supports coopératifs leurs permettant de communiquer et coordonner leurs activités.

Aujourd'hui, grâce aux technologies de réseaux et aux collecticiels, des rencontres virtuelles à large échelle sont rendues possibles [Reynard & al 98]. Plusieurs travaux de recherche dans ce sens ont déjà porté sur des domaines très variés dont celui de l'édition coopérative de documents qui se trouve à un stade plus ou moins réussi [Decouchant 95] [Pacull & al 94]. Le point commun entre tous ces systèmes est qu'ils permettent à plusieurs participants de travailler ensemble de manière synchrone ou asynchrone pour réaliser une tâche commune qui est l'édition de documents.

D'autant plus que la technologie le permet donc et est expérimentée avec succès dans certains domaines, ceci nous conduit à penser qu'il serait souhaitable que les systèmes auteurs de STI du futur (spécialement leur mode auteur) intègrent l'aspect coopératif à travers le world wide web. Ceci est l'un de nos objectifs dans cette thèse.

Dans ce chapitre nous allons commencer par décrire les notions de base concernant le domaine du TCAO (Travail Coopératif Assisté par Ordinateur) ainsi que les taxonomies adoptées dans la littérature. Nous nous intéressons plus spécialement à sa branche qui s'occupe des collecticiels d'édition coopérative, type de collecticiels qui nous inspire dans la conception de notre système puisque notre STI n'est en fin de compte qu'un ensemble de documents à introduire et à coordonner par les auteurs. Nous décrivons ensuite le mode auteur CamHits en présentant son architecture logicielle ainsi que son fonctionnement.

6.2 Travail Coopératif Assisté par Ordinateur

Littéralement, l'expression « travail coopératif » désigne un travail effectué en commun par plusieurs personnes aboutissant à une œuvre commune. Les personnes interagissent pour se donner ou accomplir un objectif commun, lequel implique une répartition des tâches (coordination) et une convergence des efforts (coopération) des membres du groupe. Nous précisons que l'expression « travail coopératif » est toujours orientée vers la réalisation d'un produit final tangible, alors que ce n'est pas toujours le cas pour « travail en équipe » qui peut être une finalité – d'apprentissage, par exemple – sans nécessairement donner lieu à un résultat tangible.

Les dix dernières années ont vu la taille des réseaux informatiques s'accroître dans une proportion considérable. Parallèlement à cela une pléthore d'applications réparties mettant en jeu ces nouvelles ressources ont vu le jour. Parmi celles-ci se trouve toute une catégorie d'applications regroupées sous le vocable TCAO (Travail Coopératif Assisté par Ordinateur : en anglais CSCW pour Computer Supported Cooperative Work).

Le but de ces applications (appelées Collecticiels ou Groupwares) est de fournir la logistique nécessaire à un groupe de personnes qui travaillent à un but commun et qui se trouvent généralement géographiquement dispersés. Si ce mode de travail coopératif existe depuis longtemps, il prend un tout autre intérêt lorsque les systèmes informatiques peuvent apporter un gain de temps précieux, en libérant les personnes des déplacements physiques, réunions de travail, et des tâches fastidieuses liées à la diffusion de l'information (téléphone, fax, courrier), en

assurant la cohérence des informations manipulées ou encore en coordonnant les actions de chacune des parties en présence.

Parmi ces applications, le travail d'édition coopérative de documents constitue une classe qui mobilise une bonne partie de la communauté scientifique travaillant sur le TCAO. En effet, ce type d'outils représente un réel intérêt puisque selon les études, 65% à 85% des articles, papiers, rapports et documentations techniques sont écrits conjointement par plusieurs auteurs [Ede & Lunsford 90], [Bair 85]. De nombreuses réalisations ont vu le jour ces dernières années tant au niveau de projets universitaires que de grands groupes industriels.

6.2.1 Notion de Collecticiel

Le « collecticiel » (ou groupware) est un néologisme qui désigne tout à la fois les processus de travail en équipe et les outils logiciels qui supportent ces processus. En d'autres termes, le Collecticiel peut être compris comme un ensemble de méthodes et de techniques de travail en équipe, ces méthodes et techniques étant « instrumentées » par des outils logiciels conçus pour améliorer les mécanismes de communication, de coopération et de coordination

Sur les plans organisationnels et humains, le collecticiel désigne de nouvelles façons de travailler privilégiant la coopération entre les individus et une meilleure coordination des actions menées. Elles reposent généralement sur un ensemble d'outils génériques : de messagerie, de bibliothèque (stockage et partage de documents, édition conjointe, etc.), de réunion, de calendrier (agendas partagés) , de gestion de tâches, etc.

L'idée d'utiliser les ordinateurs pour aider les humains à collaborer remonte au moins à la fin des années 50, avec l'invention de la notion de document hypertexte et hypermédia, par Ted Nelson et Douglas Englebart et la description de son utilisation pour créer une véritable intelligence collective dans les organisations. Depuis, l'adoption d'outils tels que le courrier électronique et les forums de discussion ont modifié nos manières de travailler.

Cependant, ces systèmes ne représentent qu'une partie des outils de communication médiatisée. Hormis pour le dialogue en temps réel, ils reposent sur des échanges de messages relativement lents, et principalement textuels. Les relations qu'ils créent entre individus, même si elles permettent de constituer de véritables communautés virtuelles, ne sont pas assez fortes pour remplacer le contact direct entre personnes situées dans un même lieu.

D'autres types de collecticiels existent pour les situations nécessitant un plus fort couplage entre participants. C'est le cas des négociations commerciales où l'influence directe entre

personnes est essentielle. De même le contrôle aérien demande un haut niveau de sécurité et des opérateurs bien coordonnés [Chatty 96]. Dans ce cas les recherches sur le TCAO proposent des outils basés sur des protocoles de communication informatiques permettant de synchroniser des applications pouvant être situées à des milliers de kilomètres, et de transmettre de la voix ou de la vidéo sur les mêmes réseaux. Par exemple les éditeurs partagés permettent à plusieurs personnes de travailler simultanément sur un même document, texte ou dessin, De même les agendas partagés permettent à un groupe de personnes de mettre en commun leurs emplois du temps.

6.2.2 Coopération vs Collaboration

Il peut s'avérer nécessaire de distinguer les deux termes au demeurant proches, souvent employés indifféremment.

- Un processus sera collaboratif s'il est issu d'acteurs qui voient différents aspects d'un problème, exploitent de façon constructive leurs différences et cherchent à mettre en œuvre des solutions communes.
- Un processus sera coopératif lorsque ses acteurs ont des intérêts similaires qu'ils planifient ensemble, où ils négocient leurs rôles mutuels et partagent des ressources pour atteindre un objectif commun tout en maintenant leur identité séparée

La coopération entraîne également un certain nombre de contraintes pour les individus. Travailler à plusieurs pour atteindre un objectif commun implique de renoncer à un certain degré de liberté, à contribuer à une tactique commune, à accepter une coordination et une discipline propre à l'équipe de travail, il faut agir en commun.

Dans la littérature, les définitions des termes "coopération" et "collaboration" changent d'un domaine à l'autre et d'un auteur à l'autre au sein d'un même domaine. Il ressort néanmoins une différenciation autour du partage des objectifs : objectif global partagé ou objectifs individuels, et autour du partage des tâches ou leur regroupement. Un consensus semble se former autour de la distinction entre les activités en groupe organisées et coordonnées, auquel cas il s'agit de « coopération », et les activités de groupe spontanées et autogérées par des règles implicites, auquel cas, il s'agit de « collaboration ».

Dans le domaine de l'enseignement coopératif assisté par ordinateur (CSCL), [Roschelle & Teasley 95] considèrent que le travail coopératif est accompli par le partage de l'effort entre les participants et qu'il s'agit d'une activité où chaque personne est responsable d'une partie de la résolution du problème. D'un autre côté, il considère que le travail collaboratif implique

l'engagement mutuel des participants dans un effort coordonné pour résoudre le problème ensemble.

Dans le même domaine, d'après [Sinia 02], la coopération et la collaboration ne diffèrent pas en termes de distribution ou de non-distribution de la tâche, mais en vertu de la façon dont elles sont décomposées. En coopération, la tâche est décomposée (hiérarchiquement) en sous tâches indépendantes ; en collaboration, les processus cognitifs peuvent être (hiérarchiquement) décomposés en couches entrelacées. En coopération, la coordination est nécessaire seulement lors de l'assemblage des résultats partiels alors que la collaboration est une activité synchrone coordonnée qui est le résultat d'une tentative permanente pour construire et pour maintenir une conception partagée du problème.

6.2.3 Coopération synchrone vs Coopération asynchrone

La présence simultanée ou différée des participants à une activité de coopération permet de qualifier la coopération, respectivement de synchrone ou d'asynchrone. Cette distinction est répercutée sur les logiciels de coopération utilisés. Ellis, Gibs et Rein [Ellis & al 91] proposent deux taxonomies pour les différentes catégories de logiciels de coopération. La première taxonomie, la plus connue, est basée sur les notions de « temps et d'espace ». La deuxième taxonomie est fonctionnelle et se base sur le type de partage : « orienté tâche » ou « orienté environnement de travail ». Les logiciels de coopération peuvent appartenir à l'une des quatre catégories de la taxonomie spatio-temporelle [Ellis & al 91] [Grudin 94] (**Figure 6.1**), à savoir :

- Activité de type “même place/même moment” soutenue par l'interaction de type face-à-face.
- Activité de type “même place/différents moments” soutenue par l'interaction asynchrone.
- Activité de type “différentes places/même moment” soutenue par l'interaction synchrone distribuée.
- Activité de type “différentes places/différents moments” soutenue par l'interaction asynchrone distribuée.

Pour la classification fonctionnelle, [Ellis & al 91] distinguent :

- Les activités à faible besoin de partage de tâche, comme celles soutenues par les systèmes temps partagé.
- Les activités à fort besoin de partage de tâche, comme celles soutenues par les systèmes de revue de logiciel.

- Les activités à faible besoin de partage d’environnement, comme celles soutenues par la messagerie électronique (e-mail).
- Les activités à fort besoin de partage d’environnement, comme celles soutenues par les salles de réunion virtuelle.

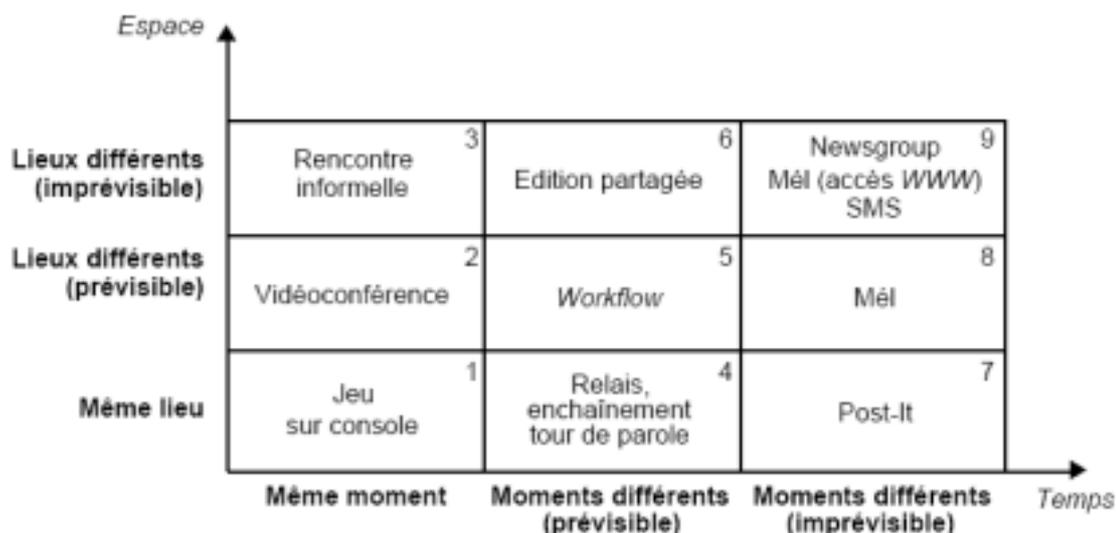


Figure 6.1. Classification espace-temps

6.3 L'édition coopérative de documents

Le domaine de l'édition coopérative fait partie évidemment du TCAO. Les éditeurs partagés sont des systèmes dédiés à l'édition collaborative de documents. Ces outils sont complexes à réaliser, en particulier pour la gestion des tâches concurrentes comme le "défaire" et "refaire" (undo et redo) ou la fusion de différentes versions. Ce domaine fait appel à un vaste spectre de compétences puisqu'en effet, doivent être considérés non seulement les aspects techniques liés au matériel, mais également les aspects humains, les interfaces homme-machine et l'intégration d'un individu avec ses spécificités dans un groupe d'individus [Larcy 00].

La coopération peut être définie comme le processus par lequel un groupe de participants géographiquement dispersés essaie de réaliser une certaine tâche en commun. Dans ce contexte, cette tâche commune est la rédaction d'un document au sens large du terme. Ce document peut être par exemple un article conjointement rédigé par une équipe de chercheurs, une documentation (par exemple celle d'un système d'exploitation faisant intervenir des programmeurs, des techniciens et des commerciaux), un cours rédigé par plusieurs enseignants, un contrat international réunissant plusieurs cabinets d'avocats et juristes, etc..

Cette section a pour but de présenter l'édition coopérative de documents dans sa généralité et de dégager un certain nombre de points qui sont importants pour notre travail.

6.3.1 Le processus d'édition coopérative

L'étude de Posner [Posner & Baeker 92] montre que le processus d'édition coopérative fait apparaître d'une part plusieurs types d'activités, et d'autre part une ou plusieurs stratégies de coordination. L'édition, qu'elle soit coopérative ou non, fait appel à plusieurs types d'activités puisqu'elle est constituée de phases distinctes : la définition d'un plan plus ou moins précis, la rédaction proprement dite, la modification éventuelle du plan, les révisions internes et ou externes, les corrections et raffinements, puis finalement une phase de polissage afin de dégager la version finale, les phases intermédiaires constituent un processus itératif.

L'édition coopérative nécessite également une ou plusieurs stratégies de coordination puisqu'il faut prendre en compte les actions des différentes personnes impliquées dans la coopération. Cette coordination est réalisée par un dialogue implicite ou explicite entre les différents protagonistes et peut, suivant la phase considérée, revêtir des formes diverses mettant en oeuvre des communications en « synchrones » où émetteur et récepteur sont actifs au même instant (le chat par exemple) ou « asynchrones » où la réception peut se faire en différé (l'email par exemple).

Les types d'activités et leurs stratégies de coordination associées sont :

- **Définition du plan du document** Associée à la définition du plan, se trouve une phase de brainstorming pendant laquelle les idées des différents participants sont évaluées, modifiées et raffinées, jusqu'à l'émergence d'un point de vue commun ou à défaut celui d'une majorité. Différents types d'actions ont été identifiés par Ellis [Ellis & al 91) au sujet de ce travail de groupe :
 - **Apport indépendant** où un participant accorde peu d'attention à ceux que font les autres membres du groupe.
 - **Apport incrémental** où un participant enrichit une proposition déjà connue du groupe. Celle-ci peut être issue de n'importe quel participant.
 - **Apport consensuel** où c'est le groupe qui décide de l'apport à effectuer.
 - **Apport partitionné** où certains membres du groupe sont désignés comme étant responsables d'un point particulier du plan.
 - **Apport procès-verbal** où un participant paraphrase ce qui est discuté par le groupe.

- **Rédaction** : La phase de rédaction peut faire appel à plusieurs stratégies [Posner & Baeker 92]. Mis à part la première (Rédacteur séparé), elles font référence à un groupe de participants travaillant de concert à la phase de rédaction.
 - **Rédacteur séparé** où chaque participant se voit attribuer une tâche distincte. Cette approche fait appel à un travail plus personnel puisque les interactions avec les autres participants n'interviennent que dans un second temps, lorsque le travail de chacun est finalement évalué, commenté, corrigé puis intégré dans le document final.
 - **Rédacteur scribe** où l'un des participants est responsable de la transcription en direct des idées émanant du groupe (ce mode n'est pas sans rappeler l'apport procès-verbal dans la définition du plan).
 - **Rédacteur unique** où l'idée est quelque peu différente puisque la transcription n'est pas effectuée en temps réel, un participant est responsable après la discussion de synthétiser les idées échangées par le groupe.
 - **Rédaction conjointe** où le groupe peut conjointement rédiger la partie concernée. La communication entre auteurs est alors très présente et nécessite un éditeur acceptant en temps réel des modifications provenant de plusieurs origines ou un moyen de communication en temps réel annexe (vidéo-conférence ou téléphone).
- **Révision/modification** : Les phases de révision et modification font appel à une succession d'interactions entre les participants responsables de chacune des deux phases. La communication est alors soit directe (un réviseur peut par exemple demander à l'auteur des précisions sur un point resté obscur à ses yeux), soit indirecte (des annotations sont ajoutées au texte lui-même ; dans ce cas la connaissance des auteurs n'est pas nécessaire). Ceci entraîne des communications en mode synchrone ou asynchrone suivant la disponibilité de chacun.
- **Polissage** : La phase de polissage, suivant le nombre de personnes qu'elle implique et leurs façons de travailler, peut revêtir deux formes. Dans la première, chacun procède à une passe et modifie le texte en conséquence. Dans la seconde, les différents protagonistes mettent en oeuvre une phase de brainstorming dans laquelle ils décident d'un commun accord des modifications à apporter au texte. Ces deux formes peuvent être, bien entendu, combinées à loisir.

En conclusion, l'édition coopérative se compose de phases d'activités fortement synchrones où les participants travaillent en même temps et de phases asynchrones où les participants

travaillent à des instants différents [Miles & al 93]. La nature même du travail est variée puisque l'on peut identifier des activités diverses : planification, rédaction, révision, communication, etc. Finalement, la manière dont les participants partagent l'information et coordonnent leurs activités dépend de beaucoup de facteurs. En particulier des relations entre les participants et de leurs rôles sociaux aussi bien que de l'état d'avancement du projet commun [Lewis & Hodges 88].

6.3.2 Fonctionnalités de base dans un collecticiel d'édition coopérative

L'éditeur est le composant autour duquel s'articule la coopération. Il doit inclure en plus des fonctionnalités classiques d'un éditeur mono-usager, les mécanismes permettant la communication entre les usagers et le partage du document. Ce partage implique en particulier le contrôle des accès aux données partagées.

- **Granularité des données partagées**

Un document peut être considéré comme un bloc monolithique ou comme un ensemble de parties plus petites. Dans ce dernier cas deux unités sont à considérer. D'une part l'unité élémentaire ou atomique qui correspond à la plus petite partie accessible par l'utilisateur (par exemple, si la ligne de texte est définie comme unité élémentaire alors il n'est pas possible d'accéder séparément aux caractères qui la composent). D'autre part, l'unité utilisable, composée d'une ou de plusieurs unités élémentaires. Une opération invoquée sur une unité utilisable sera exécutée sur toutes les unités élémentaires qui la composent. Informellement, l'unité élémentaire définit le grain minimal pour une opération, et l'unité utilisable définit le grain effectif que peut choisir l'utilisateur.

La littérature fait référence à deux approches en ce qui concerne la modélisation d'un document. La première considère qu'un document n'est qu'une suite de lignes, voire de caractères, la seconde se base sur la structure récursive d'un document décomposé en parties logiques, c'est à dire en sections, contenant des sous-sections et ainsi de suite jusqu'aux paragraphes. Informellement, à une partie logique correspond une entrée dans la table des matières du document. Le modèle de décomposition du document influence la définition de la granularité.

Par exemple, dans la première catégorie, l'éditeur Mace [Newman-Wolfe & al 91] et l'éditeur Mule [Pendergast & Vogel 90] considèrent la ligne comme l'unité élémentaire et un bloc contigu de lignes comme l'unité utilisable. Dans la deuxième catégorie, l'éditeur Quilt [Fish & al 88] [Leland & al 88] définit la partie logique la plus grande (section) comme étant à la fois l'unité

élémentaire et l'unité utilisable. Ainsi, deux utilisateurs peuvent accéder à deux sections concurremment, mais ils ne peuvent pas en revanche accéder de manière concurrente à deux sous-sections de la même section.

La première méthode permet une granularité beaucoup plus fine mais occulte en revanche la structure du document. Par exemple, il est possible de faire un bloc avec les deux dernières lignes du chapitre 1 et les six premières lignes du chapitre 2. Quelle est la sémantique qui peut être associée à un tel bloc. De plus, pouvoir sélectionner un tel bloc signifie que l'on doit pouvoir effectivement y apporter des modifications. Effacer ce bloc entraînera toutefois de profonds bouleversements à la structure du document. En outre, l'inconvénient de cette première méthode est que plus la granularité est faible plus la concurrence peut être grande et donc plus le réseau sera sollicité.

- **Gestion des données**

La gestion des données peut être caractérisée par deux aspects : (1) données dupliquées ou non dupliquées, (2) données centralisées ou réparties. En première approche, si l'on considère la manière dont les données sont utilisées nous pouvons formuler les deux règles suivantes :

- Si les opérations de consultation sont les plus fréquentes alors il est préférable que les données soient dupliquées et présentes sur chacun des sites des utilisateurs.

- Si les opérations de modification sont les plus fréquentes alors il est préférable d'adopter une solution centralisée avec une seule copie.

Malheureusement, il n'est pas simple de définir dans le cadre de l'édition coopérative laquelle de ces deux approches opposées est la bonne. De plus, les précédentes règles ne tiennent pas compte des événements extérieurs tels que les pannes. La centralisation implique que le stockage se fait sur un seul site serveur alors que la répartition suppose que chacun des sites utilisateurs est responsable du stockage d'une partie des données. L'approche centralisée va à l'encontre de l'autonomie puisque tous les sites utilisateurs dépendent du serveur central. Il constitue un goulot d'étranglement et un site critique en cas de pannes.

L'approche répartie nécessite que les sites utilisateurs offrent la fonctionnalité de serveur en plus de celle de client. De plus, certains sites peu performants peuvent dégrader les performances globales de tout le système en rendant leurs données difficilement accessibles. La solution répartie favorise le travail des utilisateurs possédant leurs données locales au détriment des autres.

Dupliquer les données permet de pallier les pannes des sites de stockage. En effet, si plusieurs copies sont disponibles, la probabilité qu'une copie soit accessible croît proportionnellement. Toutefois, la duplication nécessite des mécanismes de gestion pour assurer la cohérence mutuelle des copies. Sans duplication, dans le meilleur des cas, une donnée peut rester inaccessible pendant toute la durée de la panne du site où elle est stockée (cas favorable où une copie en mémoire permanente existe). En l'absence de pannes, les performances sont toutefois optimales. Ne pas dupliquer les données laisse les utilisateurs à la merci d'une panne de site ou de lien de communication.

- **Communication**

La communication concerne l'échange d'informations entre les participants géographiquement réparties. Deux types de fonctionnements sont à considérer : le « mode synchrone » où chaque utilisateur voit les interactions des autres participants en temps réel, et le « mode asynchrone » où les modifications ne sont pas immédiatement répercutées globalement.

Le premier mode est désigné par l'acronyme WYSIWIS (What You See Is What I See) qui exprime bien la philosophie sous-jacente. Plusieurs éditeurs dont GROVE [Ellis & al 91] et Mule [Pendergast & Vogel 90] fonctionnent selon ce modèle (**Figure 6.2.**).

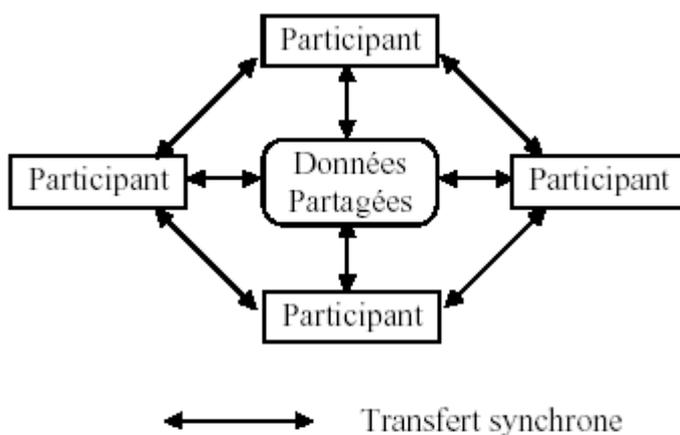


Figure 6.2. Coopération synchrone

Ils sont caractérisés par une granularité fine (caractère ou ligne) qui permet des mises à jour fréquentes (la fréquence des mises à jour étant liée à la taille des données à modifier). Poussé à l'extrême, ce mode implique que non seulement l'espace de travail (les fichiers manipulés) est le même pour tous les utilisateurs mais encore que l'espace d'affichage (généralement la fenêtre de visualisation) est identique. Par exemple, le défilement occasionné par un des utilisateurs se

répercute sur tous les écrans et la position des curseurs de chacun des participants est visible par tous. La **(Figure 6.3)** montre l'interface de l'éditeur partagé Calliope [Mitchell 96] lorsque deux rédacteurs éditent simultanément une même zone de texte. Le premier a pris le contrôle du mot 'take', entouré par un rectangle gris représentant un verrou, tandis que le deuxième a pris le contrôle de tout le second paragraphe (entouré également d'un cadre gris). Ces opérations sont accessibles par un menu déroulant. Les zones de texte entourées de rectangles gris ne sont modifiables que par la personne qui en a demandé le contrôle.

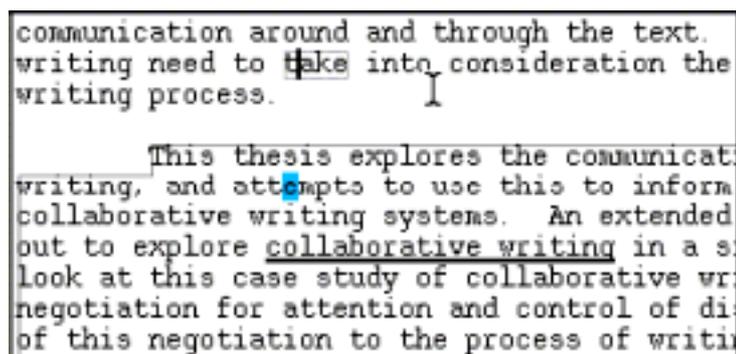


Figure 6.3. Interface de l'éditeur partagé Calliope [Mitchell 1996]

Finalement, un mode purement « asynchrone » **(Figure 6.4)** considère que les utilisateurs travaillent dans un espace de travail local et qu'ils diffusent leurs modifications quand bon leur semble (par exemple : l'éditeur Prep [Neuwirth & al 90], l'éditeur Quilt [Fish & al 88]). Il permet en particulier de travailler avec des versions brouillons qui sont améliorées avant d'être livrées aux autres participants. Le mode asynchrone favorise généralement le caractère privé du travail de chacun. Certains systèmes considèrent plusieurs modes comme par exemple CES [Greif & al 93] qui propose un mode purement asynchrone ainsi qu'un mode qui autorise un seul participant à modifier le document tout en permettant aux autres participants de voir les modifications en temps réel.

- **Cohérence des données**

Que les données soient dupliquées ou non, garantir la cohérence des données partagées est important. En effet, des accès concurrents aux données partagées sont possibles et par ce biais peuvent entraîner des incohérences. Par exemple, deux opérations modifiant une même portion de texte peuvent être entrelacées, produisant un résultat non souhaité. Lorsque la duplication est mise en œuvre à ce problème s'ajoute celui de la cohérence mutuelle des copies. Plusieurs copies peuvent effectuer des opérations dans des ordres différents ce qui entraîne une incohérence de

l'ensemble des copies. Différents critères de cohérence sont définis dans la littérature. Ces critères offrent des cohérences plus ou moins fortes au sens des contraintes de sérialisation des opérations.

Garantir la cohérence est important. Toutefois, maintenir une cohérence forte sur les données, obtenir la dernière mise à jour ou suivre les modifications en temps réel n'est pas toujours indispensable. Par exemple, une version obsolète ou incohérente peut s'avérer suffisante lorsqu'il s'agit uniquement d'avoir un aperçu de son contenu, voire seulement son plan. Il est donc utile d'offrir aux utilisateurs plusieurs qualités de service afin de leur laisser le choix entre une réponse rapide avec un résultat éventuellement obsolète ou une réponse plus lente avec une certitude absolue d'obtenir la dernière version. L'utilisation de caches locaux permet par exemple de diminuer les temps de latence pour certaines opérations, au détriment de la cohérence (par exemple, SharedBook [Lewis & Hodges 88], Mace [Newman-Wolfe & al 91]).

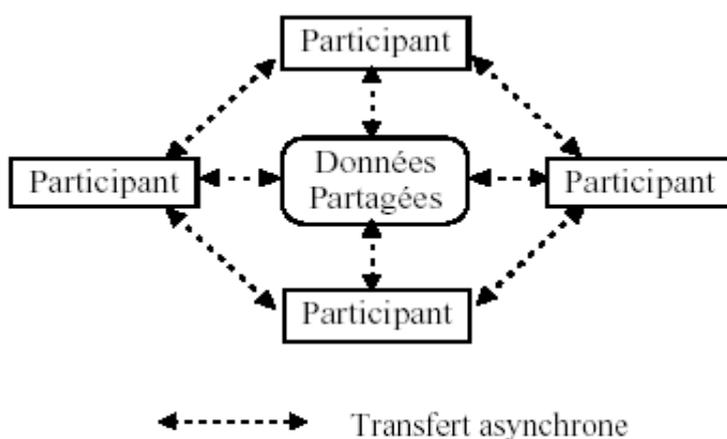


Figure 6.4. Coopération asynchrone

6.4 Mode auteur coopératif CamHits

Le mode auteur coopératif CamHits [Talhi & al 05] [Talhi & al 06] consiste en un éditeur de connaissances qui présente aux auteurs tous les outils nécessaires à l'élaboration coopérative d'un STI Hypermédia. Du point de vue d'un auteur, construire un STI avec CamHits consiste à introduire, via cet éditeur, un ensemble d'objets qui seront manipulés par le mode apprenant. Ces objets sont constitués de la matière à enseigner sous formes d'uah, du réseau de prérequis sous forme de graphes orientés, des paramètres du STI sous forme de prédicats, et des connaissances tutorielles sous forme de règles de production.

6.4.1 Modélisation de CamHits

La coopération dans CamHits est introduite au niveau de l'édition de la matière à enseigner et au niveau de l'édition du réseau de prérequis. Ces deux composants sont bien structurés : la matière étant hiérarchisée en parties, chapitres et uah, et le réseau de prérequis en sous réseaux (prérequis-parties, prérequis-chapitres et prérequis-uah). Cette structure s'adapte bien pour la fragmentation de ces deux composants et constitue de ce fait la base de notre approche d'édition coopérative.

Le principe d'édition coopérative que nous avons exploité repose en fait sur les deux concepts clés utilisés dans la plupart des éditeurs coopératifs : la *fragmentation* et l'attribution de *rôles d'édition* sur les différents fragments. Nous avons défini trois *rôles* pour les auteurs dans CamHits : auteur principal, constructeur et lecteur/commentateur.

Avant tout projet de construction d'un STI, une phase de négociation via des outils de communication s'impose. Les *rôles* sont alors attribués par l'auteur principal aux différents auteurs autour des différents fragments du cours conformément à leurs compétences et à leurs disponibilités. De l'analyse précédente, nous pouvons extraire trois univers conceptuels bien distincts [Lancy 00] :

- Le monde des données ;
- Le monde des utilisateurs qui accède et manipule les éléments de l'ensemble des données ;
- Le monde de l'organisation qui structure les deux mondes précédents.

La (**Figure 6.5**) représente simplement les relations entre ces trois espaces ainsi que les outils permettant de les manipuler

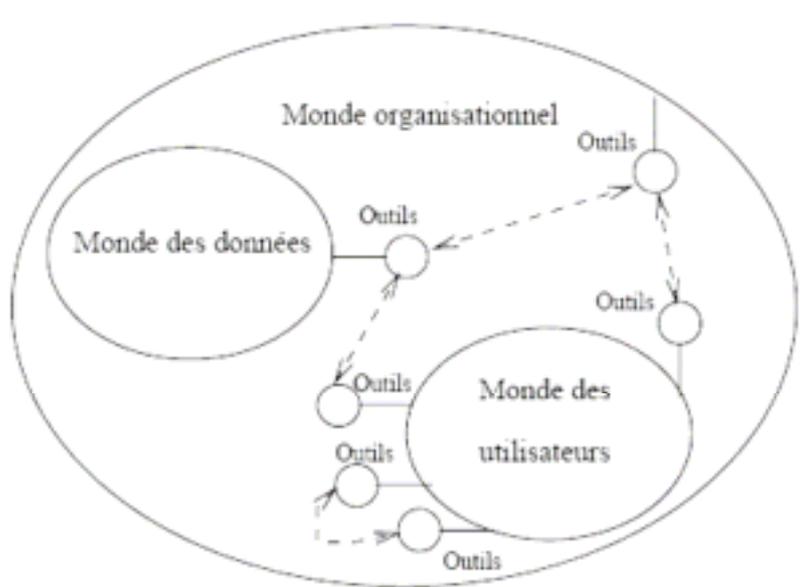


Figure 6.5. Les trois univers conceptuels de CamHits

6.4.1.1 Le monde des données

Par données, nous entendons ici les objets à produire en tant que résultat de la coopération. Il s'agit donc ici essentiellement de documents électroniques. Les outils permettant d'obtenir et de manipuler de tels documents sont des éditeurs qu'ils soient coopératifs ou non. Ici les termes de documents et d'éditeurs sont à prendre au sens large et ils englobent par exemple outre le couple classique (texte, traitement de texte) les couples (image, éditeur d'image), (son, éditeur de son), etc. Par essence même une application coopérative nécessite que les données soient partagées entre ses différents utilisateurs. Le partage peut être plus ou moins fin.

Un document partagé peut être considéré comme un bloc monolithique indissociable ou comme un ensemble d'objets plus petits, lesquels peuvent être ou non structurés entre eux (un document peut être vu par le système comme une simple suite de caractères ou tout au contraire comme un ensemble structuré récursivement en parties logiques). Les possibilités de partage dans le premier cas sont réduites aux documents entiers ce qui empêche la concurrence des actions et implique de travailler d'une manière séquentielle. La seconde approche que nous adoptons offre plus de latitude et de souplesse en matière de coopération simultanée.

Deux unités caractérisant les éléments d'un document sont considérées dans la littérature. Il s'agit de l'unité élémentaire correspondant à l'entité la plus fine accessible à l'utilisateur (dans notre cas c'est l'uah), c'est à dire sur laquelle il peut invoquer une action et de l'unité utilisable

composée d'une ou plusieurs unités élémentaires (dans notre cas un chapitre par exemple ou une partie du module). Si l'unité utilisable peut être choisie par l'utilisateur, ce qui lui permet de faire varier la granularité du partage, l'unité élémentaire représente une contrainte système.

6.4.1.2 Le monde des utilisateurs

Il regroupe les éléments de perception du système par l'utilisateur :

- Perception de lui-même à savoir ce qu'il est, ce qu'il fait (respectivement a fait) et pourquoi il le fait (respectivement l'a fait) ;
- Perception des autres à savoir ce qu'ils sont ce qu'ils font (respectivement ont fait) et pourquoi ils le font (respectivement l'ont fait) ;
- Perception des données à savoir ce qu'elles sont (leur sémantique), quel est leur état (valeur courante) et quel est leur histoire ?

Les éléments du monde des utilisateurs doivent donc capter l'awareness (la conscience du groupe). Les outils appartenant au monde des utilisateurs sont donc essentiellement des interfaces vers les autres univers conceptuels du système y compris lui-même. Un outil d'édition regroupe donc en fait deux outils distincts appartenant à deux mondes différents : l'outil qui manipule les données proprement dites (monde des données) et son interface utilisateur (monde des utilisateurs). Le monde des utilisateurs doit posséder des outils spécifiques pour capter les interactions et faciliter la conscience des autres :

Outil de messagerie : Il permet une interaction asynchrone entre deux utilisateurs ou groupe d'utilisateurs. C'est le plus utilisé des collecticiels. Remarquons que les messages peuvent être libres et à la charge totale de l'utilisateur ou qu'ils peuvent être structurés et disposer de champs automatiquement remplis par le système identifiant ainsi de manière précise le contexte de l'utilisateur au moment de la création du message. Le ou les récepteurs du message peuvent ainsi déterminer les conditions qui ont motivé cet envoi ou cette diffusion et réagir en conséquence.

Forum électronique : Il reprend le principe de la messagerie électronique avec deux différences majeures :

- ce ne sont pas les messages qui viennent aux utilisateurs mais les utilisateurs qui viennent les chercher ;
- un forum possède une histoire accessible à tous les membres du forum. Cette histoire s'enrichit au fur et à mesure des participations.

6.4.1.3 Le monde organisationnel

Cet univers conceptuel regroupe les éléments qui relèvent de la structuration du travail en groupe. Si les données évoquées précédemment occupent l'espace de production d'un collectif ces différents éléments occupent l'espace de coordination. Un environnement de TCAO ne peut ignorer cette manière de procéder. C'est elle qui structure la coopération. Les problèmes informatiques spécifiques au monde de l'organisation sont essentiellement les suivants :

- La création et destruction des projets et des activités ;
- L'ajout ou suppression de membres (individus ou documents) à un projet ou activité ;
- Le contrôle et la synchronisation des activités afin de respecter les délais spécifiés ;
- La modification des rôles ;
- Le contrôle de la sécurité : en effet, l'appartenance d'un individu à un projet, activité ou tâche et le rôle qu'il y joue détermine les droits qu'il possède sur ces ensembles ainsi que les droits qu'il possède sur les documents membres de ces ensembles ;
- La persistance des données ;

La définition d'un modèle d'interaction coopératif efficace repose sur le choix de modes d'interaction naturels et avec le minimum de contraintes possibles. Le processus de construction coopérative d'un STI est caractérisé par un enchaînement de phases au cours desquelles les auteurs peuvent travailler individuellement ou collectivement. Ainsi, les auteurs accèdent aux différents fragments du STI et la coopération prend place à travers des annotations et des commentaires liés aux fragments développés par les autres

6.4.2 Architecture logicielle de CamHits

Le mode auteur coopératif est organisé selon une architecture client/serveur centralisée [Orfali & al 97] (**Figure 6.6**). Par conséquent, toutes les communications transitent automatiquement par le site central (ou serveur). A chaque site auteur on associe un processus client (PCL) qui accomplit toutes les tâches traitables localement (les tâches d'édition par exemple). Quant au niveau central, nous définissons un processus serveur (PSR) qui gère toutes les communications entre les différents PCL et tient à jour le contenu de la copie centrale et la structure logique du STI. Le PSR détient à son niveau les versions les plus récentes des objets du STI ainsi que sa structure logique, tandis que les stations des différents auteurs peuvent contenir des versions qui ne sont pas forcément à jour.

Par conséquent, il appartiendra à ces auteurs de les récupérer du site central. Au lancement de l'application par un auteur, le PSR prépare aux PCL toutes les conditions pour leur permettre d'évoluer de façon totalement autonome, il n'interviendra que dans certains cas tels que l'accès aux données partagées, l'initiation d'une communication, etc.

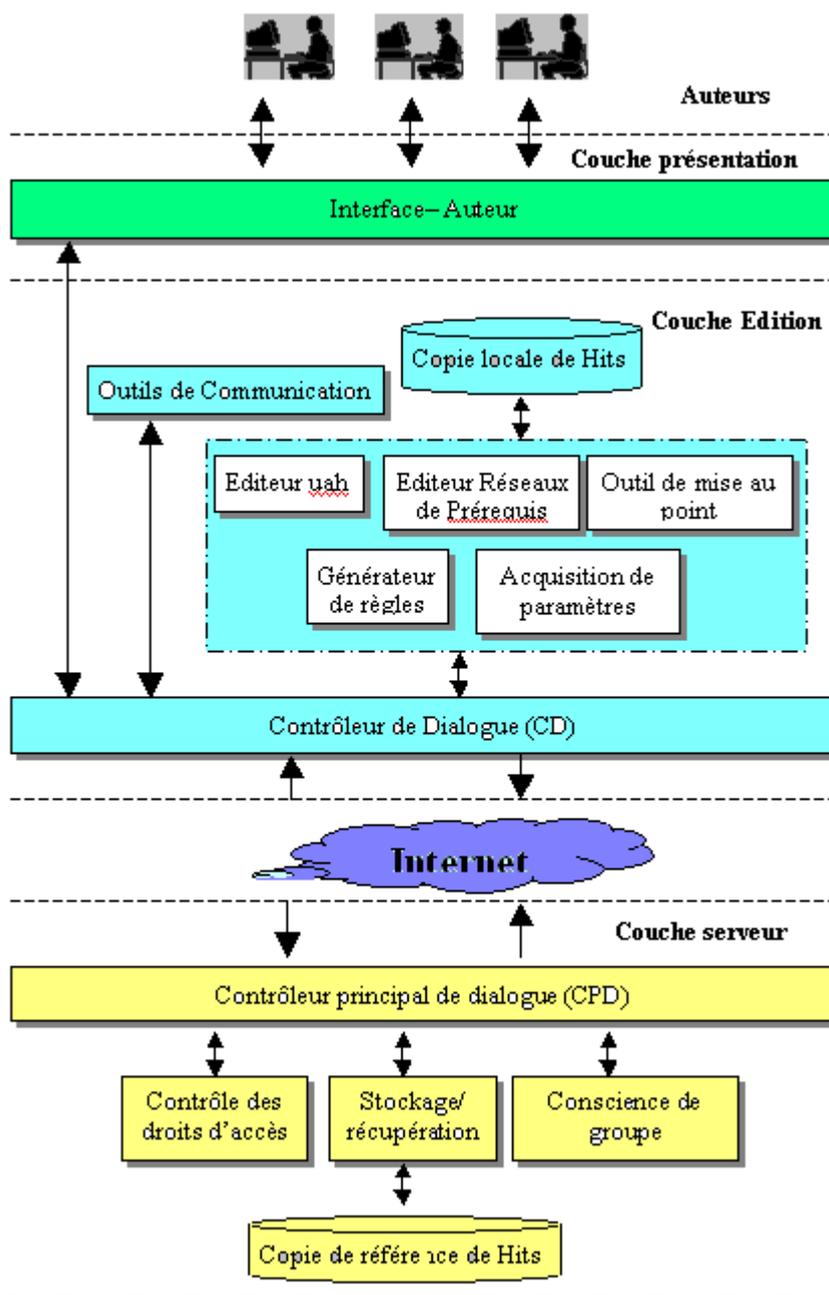


Figure 6.6. Architecture logicielle en couches du mode auteur

Comme le montre la (Figure 6.6), l'architecture logicielle offre plusieurs types de traitements que nous pouvons décomposer en trois couches : la couche serveur, la couche éditeur et la

couche présentation. Chaque couche est structurée comme une collection de modules regroupant chacun plusieurs objets capables de réaliser le type de traitement approprié.

La double nécessité d'assurer à la fois les échanges d'informations entre les deux couches du poste client et entre le client et le serveur, se traduit par la présence de « contrôleurs de dialogue ». Nous interposons donc entre la couche de présentation et la couche d'édition un contrôleur de dialogue (CD), et entre la couche serveur et la couche d'édition le contrôleur principal de dialogue (CPD). L'arrivée d'un événement implique automatiquement la transition du message associé par les contrôleurs de dialogue qui sont les seuls à pouvoir décider des traitements exacts à déclencher parmi ceux qui sont définis au sein d'une couche.

6.4.2.1 Couche Serveur

Cette couche regroupe plusieurs types de traitements, parmi lesquels nous distinguons ceux qui sont liés à la gestion de la structure logique du STI, ainsi que les contenus des composants du STI. Ils permettent ainsi aux auteurs de stocker et de récupérer les objets du STI dont la structure logique est déclarée aussi bien au niveau central qu'au niveau local. Cette couche est aussi responsable des opérations de contrôle des droits d'accès, du traitement des événements et de la notification de leurs conséquences aux auteurs. Dans le cas de notification des événements, par exemple, le module qui en est chargé gère un ensemble de files d'attentes telles que la file des engagements, la file de blocage, etc [Zidani 00]. Chaque fois qu'il est invoqué suite à un événement, ce processus identifie les auteurs destinataires et procède à la structuration des notifications sous forme de messages transmissibles. Ces messages seront alors mis à la disposition d'un autre module émetteur qui se chargera de l'émission.

6.4.2.2 Couche Editeur

Cette couche regroupe à son tour plusieurs types de traitements permettant à chaque auteur de manipuler les objets composant le STI. Ces traitements incluent aussi bien le support des actions individuelles, que l'aspect de partage et de gestion de la transparence. Par exemple, l'accès à un fichier dans un système mono-usager délivre directement son contenu. Par contre dans notre cas, ce processus déclenchera une suite de traitements tels que la vérification des droits d'accès de l'auteur, de l'état de blocage du composant et enfin l'avertissement des auteurs travaillant sur ce même composant. Au niveau de chaque site, les traitements associés permettent à l'auteur d'enregistrer localement les contenus des composants qui lui sont accessibles. Il sollicitera

régulièrement le serveur pour maintenir à jour les versions de ces composants. Les différents composants de la couche éditeur sont :

1. **Editeur d'uah /réseau de prérequis** : Ce composant est constitué de deux modules permettant la création/mise à jour des différents objets du STI (uah, réseaux de prérequis). Le premier module permet l'édition des uah. Le deuxième module permet à un auteur d'éditer les réseaux de prérequis sous forme graphique. Le réseau est formé de nœuds et d'arcs liant ces nœuds et indique les différents cheminements possibles entre les constituants de la matière à enseigner. Trois niveaux sont utilisés dans le réseau d'OPs. Un premier niveau montre les prérequis entre parties, un second décrit les prérequis entre les chapitres d'une partie, et le troisième montre les prérequis entre les uah d'un chapitre. Des outils sont fournis également pour l'édition du réseau de concepts.
2. **Module d'acquisition des paramètres du STI** : Via ce module, l'auteur principal doit spécifier au système la manière avec laquelle est décomposée la matière d'enseignement (nombre de parties, nombre de chapitres par partie, nombre d'uah par chapitre, etc.). Ces paramètres sont mémorisés sous forme de prédicats puis utilisés pour instancier les règles maîtresses. Par exemple le prédicat nbuah (1,2,4) indique que le chapitre 2 de la partie 1 contient 4 uah.
3. **Générateur des règles instances** : Ce module permet à l'auteur principal de générer les cinq paquets de règles instances qui représentent les différents plans de tutorat. Cette génération est effectuée par instanciation de cinq paquets de règles maîtresses et est réalisée sur la base des paramètres du STI introduits via le module précédent.
4. **Outil de mise au point** : Comme la plupart des modes auteurs, CamHits offre un outil permettant d'assister l'auteur dans la détection des erreurs et des incohérences qui peuvent se produire durant la construction du STI. Il s'agit essentiellement d'une mise au point technique qui consiste à rendre le STI exécutable sans erreurs, c'est à dire conforme au modèle imposé par CamHits. Le contrôle de compatibilité matière/paramètres par exemple permet de vérifier si les paramètres indiqués par l'auteur correspondent réellement aux contenus des différentes entités de matière stockées.

6.4.2.3 Couche Présentation

Cette couche regroupe un ensemble organisé d'objets interactifs définissant la partie perceptible du système (boutons, curseur, barre de défilement, thermomètres de progression de tâches, menus déroulants, etc.). Ainsi à tout objet modélisant une partie du domaine de notre

application, nous associons une technique de présentation matérialisée par un objet interactif qui réagit aux actions de l'auteur.

L'objectif de la couche présentation est de rendre le système facile à manipuler en présentant à l'auteur une vue explicite qui ne comporte pas d'ambiguïtés. A part les menus déroulant traduisant les différentes fonctions, nous trouvons spécialement une boîte à outils contenant des icônes référençant les fonctions les plus utilisées, et des palettes graphiques permettant la construction graphique du réseau de prérequis.

6.4.2.4 Les contrôleurs de dialogue CD et CPD

Ces contrôleurs (**Figure 6.7**) se composent chacun de trois modules indépendants réalisant respectivement les tâches de contrôle, d'émission et de réception de messages. Le module de contrôle regroupe toutes les fonctions qui permettent de coordonner et synchroniser l'exécution des différents modules au sein des trois couches, conformément aux actions des différents auteurs. Il dispose à tout instant de toutes les informations nécessaires pour déterminer exactement quels sont les objets à invoquer au sein des couches dont il est responsable.

A chaque fois qu'un événement se manifeste, le message matérialisant cet événement est délivré au module de contrôle par le récepteur associé. Le module de contrôle réagit alors en suivant trois étapes : analyser l'événement, dresser un plan d'actions puis exécuter le plan établi.

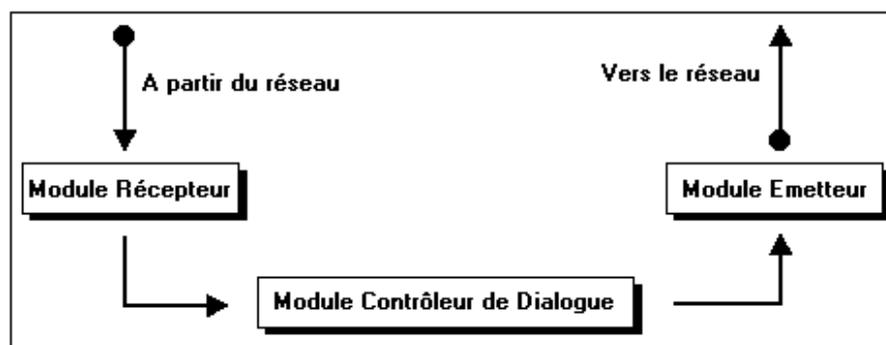


Figure 6.7. Structure des contrôleurs de dialogue CD et CPD

Un événement généré par un auteur à partir de l'interface, aboutit à son CD qui l'analyse et détermine s'il est traitable localement ou s'il nécessite l'intervention du serveur. S'il s'agit d'un traitement local, le CD commence à invoquer les modules de la couche éditeur et si l'événement exige une réponse alors le CD élabore le message approprié et le délivre à l'auteur via son interface.

Par contre si l'événement nécessite l'intervention du serveur, le CD déclenche les traitements pour charger toutes les informations constituant les messages à émettre et déclenche le module émetteur associé. Le module récepteur du serveur reçoit les messages émis et les délivre au module de contrôle du CPD qui les analyse à son tour et élabore un plan d'actions. Il procède ensuite à l'exécution de ce plan en invoquant les modules de la couche serveur concernés par les traitements de cet événement. Une fois les traitements achevés, la réponse reprend le chemin inverse sous forme d'un ou plusieurs messages selon la taille des informations renvoyées.

6.4.2.5 Conscience de groupe

Lorsque des personnes travaillent en « co-présence », c'est-à-dire en même temps et dans le même lieu, elles échangent un certain nombre d'informations, certaines implicites et d'autres explicites [Muhammad & al 05]. Ces informations créent chez chaque participant une conscience de groupe, qui lui permet :

- De comprendre et mesurer l'activité et la dynamique du groupe ;
- De situer sa propre action au sein du groupe, et ainsi de coordonner ses propres activités avec celles des autres.

Cette notion de conscience de groupe (awareness) est originellement définie par Dourish et Belloti [Dourish & Belloti 92] comme étant « la compréhension des activités des autres, qui permet de donner un contexte à sa propre activité ». Dans une équipe virtuelle, la disparition de la co-présence entraîne la disparition de certaines informations, particulièrement les informations implicites, qui entraîne à son tour la disparition de la conscience de groupe. Le but, dans le contexte des applications de coopération, est donc de reconstruire la conscience de groupe grâce à des modèles et par le biais de mécanismes adaptés. Modéliser cette conscience de groupe nécessite :

- L'identification des informations utilisées en co-présence pour construire la conscience de groupe ;
- Une synthèse claire et utilisable des nombreuses informations que l'on peut obtenir des systèmes informatiques ;
- Une représentation à l'utilisateur d'une information pertinente mais non obstructive.

Pour développer le support informatique approprié, nous avons dégagé les principaux facteurs concernés par la transparence au sein de l'espace de travail. La (**Table 6.1**) résume les éléments considérés ainsi que les questions que peuvent se poser les auteurs participants.

Facteurs	Interrogations des participants
<i>Identification</i>	<i>Qui participe à l'activité ?</i>
<i>Lieu</i>	<i>Où est-il ?</i>
<i>Présence</i>	<i>Est-il présent en session ?</i>
<i>Rôle</i>	<i>Avec quel rôle intervient-il ?</i>
<i>Modifications</i>	<i>Qu'est ce qu'il a modifié ?</i>
<i>Fragments</i>	<i>Sur quels fragments travaille t-il ?</i>

Table 6.1. Facteurs concernés par la conscience du groupe

CamHits exploite un mécanisme de notification qui lui permet de diffuser aux auteurs les différents événements qui se produisent au sein de l'espace de travail partagé. Au cours du processus de coopération, les participants prennent alors connaissance de “*qui fait quoi, où et quand?*”. C’est ainsi que CamHits peut fournir à chaque participant des informations concernant ses collègues (rôle, identité, etc.), les interactions au sein de l'espace de travail (activités courantes, parties manipulées, modifications réalisées, etc.), ainsi que les états de chaque fragment du STI (bloqué, libre, auteurs l’ayant manipulé, instant de blocage, etc.).

Le processus de gestion des événements est à la charge du processus central (**Figure 6.6**). Comme le serveur encapsule la copie de référence, la notification d'événements liés aux différents fragments du STI se trouve ainsi facilitée. Chaque auteur peut demander au serveur d'être informé à chaque fois qu'un événement se produit au sein de l'espace partagé. Par exemple, il peut établir un engagement afin de recevoir les modifications apportées à un fragment spécifique du STI, qui lui seront alors automatiquement délivrées. Les engagements sont établis manuellement par les participants.

Par exemple, en ce qui concerne la partie «*Normalisation des bases de données*», l'auteur Said peut configurer les événements dont il souhaite être notifié, il spécifie au système qu'il souhaite être notifié à chaque fois qu'elle est bloquée ou libérée. Il sollicite également le droit de blocage sur cette partie ainsi que les mises à jour réalisées par les autres auteurs. Said peut également spécifier la manière selon laquelle il désire être informé par des marques sur les objets, des messages à l'écran ou des messages sonores. Enfin, il peut être concentré sur un travail et donc, pour ne pas

être perturbé, décider de fermer son espace de travail. De cette manière, nous permettons aux participants de contrôler eux-mêmes le degré d'ouverture de leur espace de travail.

6.4.3 Le module d'acquisition des paramètres

Ce module a pour but d'instancier les prédicats constituant les classes de paramètres du STI. L'activation de ce module conduit à l'affichage d'un dialogue avec l'auteur principal (**Figure 6.8**) en vue de saisir les paramètres décrivant le STI créé.

CamHits : saisie des paramètres du STI
<p>Quel est le nombre d'épreuves de prétest ? _ Pour l'épreuve 1, quel est le nombre de réponses ? _ Pour l'épreuve 2, quel est le nombre de réponses ? _ ...</p> <p>Quel est le nombre de parties du cours ? _ Pour la partie 1, quel est le nombre de chapitres ? _ Pour le chapitre 1, quel est le nombre d'uah ? _ Pour l'uah1, quel est le nombre de questions ? _ Pour la question 1, quel est le nombre de réponses ? _ Pour la question 2, quel est le nombre de réponses ? _ ... Pour l'uah1, quel est le nombre d'exercices ? _ Pour l'exercice 1, quel est le nombre de solutions ? _ Pour l'exercice 2, quel est le nombre de solutions ? _ ... etc.</p>

Figure 6.8. Dialogue de saisie des paramètres du STI

Une fois que la donnée est tapée et validée par l'auteur, elle est contrôlée par rapport aux limites imposées par CamHits (par exemple : le nombre de sections doit être ≥ 1 , le nombre de questions doit être ≥ 0 , le nombre de réponses doit être ≥ 2 , etc.) puis elle est utilisée pour instancier le prédicat correspondant. Une portion de l'algorithme général d'instanciation des différentes classes de paramètres est donnée ci-dessous :

Algorithme acquisition-paramètres

Début

```

entrer Nbepre;                               /* nombre d'épreuves de pretest */
instancier nbepre(Nbepre);
    pour i=1, Nbepre faire
        entrer Nbrepre ;                       /* nombre de réponses de l'épreuve */
    
```

```
instancier nbreppre( i, Nbreppre );  
fpour  
entrer Nbpar ; /*nombre de parties*/  
instancier nbpar(Nbpar) ;  
pour j=1, Nbpar faire  
    entrer Nbchap ; /*nombre de chapitres*/  
    instancier nbchap(j,Nbchap) ;  
    pour k=1, Nbchap faire  
        entrer Nbuah ; /*nombre d'uah*/  
        instancier nbuah(j,k,Nbuah) ;  
        pour l=1, Nbuah faire  
            entrer Nbques ; /*nombre de questions*/  
            instancier nbques (j, k, l, Nbques) ;  
        etc.
```

Fin.

6.4.4 Le Générateur des règles instances

Ce module a pour but de générer la base de connaissances relative à un STI donné, c'est à dire les paquets de règles instances. Cette génération est effectuée par instanciation des paquets de règles maîtresses. L'instanciation est réalisée sur la base des paramètres du STI saisis par l'auteur comme indiqué ci dessus. L'activation de ce module conduit à l'affichage d'un dialogue (**Figure 6.9**) puis à l'exécution séquentielle d'un certain nombre de sous-modules qui sont spécialisés chacun dans la génération d'un paquet de règles maîtresses.

Afin de montrer un exemple de règles maîtresses et comment s'effectue l'opération d'instanciation, nous prenons le cas des « règles de planification » dans le cas du parcours basé sur les composants.

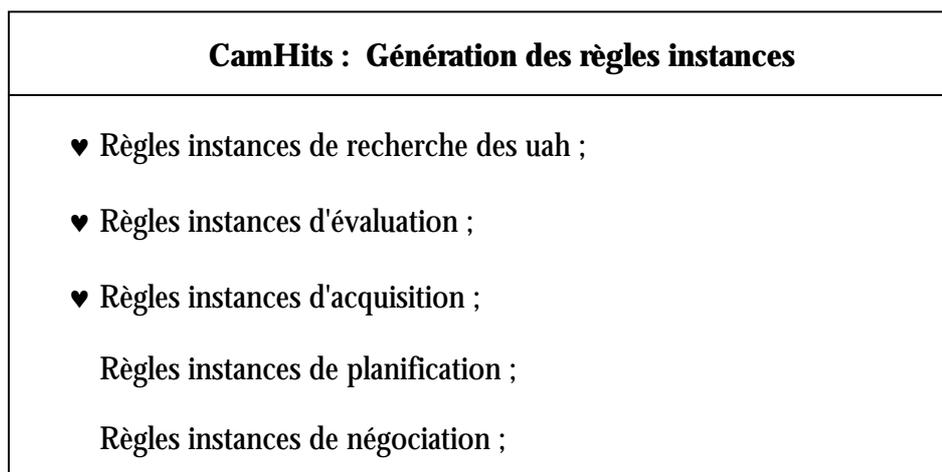


Figure 6.9. Dialogue affiché à l'activation du générateur des règles instances

a) Un exemple de règles maîtresses : les règles de planification

Comme nous l'avons déjà mentionné chapitre 5, les règles maîtresses de planification sont conçues pour assurer un parcours en profondeur d'abord des quatre premiers niveaux de l'univers des uah. Les règles mères de planification sont alors comme suit:

- RMP1 Si démarrage-normal ou acquis-rien Alors présenter-sommaire*
- RMP2 Si présenté-sommaire Alors effectuer-pretest*
- RMP3 Si réussi-pretest Alors présenter-résumé-par(1)*
- RMP4 Si non(réussi-pretest) Alors arrêt-avec-regret*
- RMP5 Si acquis-par(Npar) Alors présenter-résumé-par(Npar+1)*
- RMP6 Si acquis-par(NparMax) Alors effectuer-posttest-module*
- RMP7 Si réussi-posttest-module Alors acquis-module*
- RMP8 Si non(réussi-posttest-module) Alors remédier-module*
- RMP9 Si acquis-module Alors arrêt-avec-félicitation*
- RMP10 Si présenté-résumé-par(Npar) Alors présenter-résumé-chap(Npar,1)*
- RMP11 Si acquis-chap(Npar,Nchap) Alors présenter-résumé-chap(Npar, Nchap+1)*
- RMP12 Si acquis-chap(Npar, NchapMax) Alors effectuer-posttest-par(Npar)*
- RMP13 Si présenté-résumé-chap(Npar,Nchap) Alors présenter-uah(Npar,Nchap,1)*
- RMP14 Si acquis-uah(Npar,Nchap,Nuah) Alors présenter-uah(Npar,Nchap,Nuah+1)*
- RMP15 Si non(acquis-uah(Npar,Nchap,Nuah)) Alors présenter-uah(Npar,Nchap,Nuah)*
- RMP16 Si acquis-uah(Npar,Nchap,NuahMax) Alors effectuer-posttest-chap(Npar,Nchap)*
- RMP17 Si présenté-uah(Npar,Nchap,Nuah) Alors tester-uah(Npar,Nchap,Nuah)*

RMP18 Si présenté-uah(Npar,Nchap,Nuah) Alors acquis-uah(Npar,Nchap,Nuah)

RMP19 Si présenté-sommaire Alors présenter-résumé-par(1)

RMP20 Si acquis-par(NsecMax) Alors acquis-module

RMP21 Si réussi-posttest-par(Npar) Alors acquis-par(Npar)

RMP22 Si non(réussi-posttest-par(Npar) Alors remédier-par(Npar)

RMP23 Si réussi-posttest-chap(Npar,Nchap) Alors acquis-chap(Npar,Nchap)

RMP24 Si non(réussi-posttest-chap(Npar,Nchap) Alors remédier-chap(Npar,Nchap)

Remarque 1. La règle RMP18 est utilisée dans le cas d'une uah non évaluable.

Remarque 2. Les règles RMP19 et RMP20 sont utilisées dans le cas où le STI ne contient pas de pretest et/ou de posttest du module

Remarque 3. Les arguments NparMax, NchapMax et NuahMax désignent respectivement les plus grands numéros de partie, de chapitre et d'uah.

b) Instanciation des règles maîtresses

L'instanciation des règles maîtresses de planification est effectuée sur la base des paramètres du STI pour pouvoir choisir les règles à instancier. Par exemple, dans le cas où ne figurerait pas de pretest dans le STI, c'est à dire quand le paramètre « nombre d'épreuves de pretest » est égal à zéro, les deux règles RMP2 et RMP3 ne seront pas utilisées. C'est plutôt la règle RMP19 qui sera utilisée. Cela signifie que, juste après la présentation du sommaire, le système passe à la présentation du résumé de la partie 1. Nous décrivons ci-dessous une portion de l'algorithme permettant d'instancier les règles RMP17 et RMP18.

Algorithme instanciation ;

début

*ouvrir en lecture RMPLAN ; /*fichier des règles maîtresses de planification*/*

*ouvrir en lecture PARTUT ; /*fichier des paramètres du STI*/*

*ouvrir en écriture RIPLAN ; /*fichier des règles instances de planification*/*

...

prendre nbsec(Nbsec) dans PARTUT;

pour i=1, Nbpar faire

prendre nbchap(i, Nbchap) dans PARTUT;

pour j=1, Nbchap faire

prendre nbuah(i, j, Nbuah) dans PARTUT;

pour k=1, Nbuah faire

```
prendre nbques(i, j, k, Nbques);
prendre nbexo(i, j, k, Nbexo);
si Nbques = 0 et Nbexo = 0 alors /* uah non évaluable*/
    prendre RMP18 dans RMPLAN;
sinon
    prendre RMP17 dans RMPLAN;
fsi
    /*instanciation*/
    Npar := i; Nchap := j; Nuah := k;
    écrire règle instanciée dans RIPLAN;
    fpour
    fpour
    fpour
...
fermer tous les fichiers ouverts;
fin
```

6.4.5 L'outil de mise au point

Comme la plupart des modes auteurs, CamHits offre un outil permettant d'assister l'auteur dans la détection des erreurs et des incohérences qui ont pu se produire dans le STI créé. La mise au point que nous entendons ici n'est pas la mise au point pédagogique qui consiste en l'amélioration du contenu du STI, mais une mise au point technique qui consiste à rendre le STI exécutable sans erreurs, c'est à dire conforme au modèle imposé par CamHits.

La mise au point pédagogique incombe totalement aux auteurs. Il n'y a aucune assistance de la part du système. Un auteur peut, après avoir saisi toutes les connaissances constituant le STI et mis au point (techniquement) ce dernier, passer en mode apprenant pour jouer le rôle de l'apprenant et ainsi évaluer pédagogiquement son STI. S'il juge nécessaire d'apporter quelques changements à ce dernier, il reviendra alors au mode auteur puis il effectue, grâce à l'éditeur, n'importe quelle opération sur les uah. Il peut, en effet, modifier, ajouter ou supprimer des uah tant qu'il le voudra.

La mise au point technique, quant à elle, est assistée par un outil que l'auteur activera une fois le STI créé. Cet outil aidera l'auteur à détecter les anomalies qui ont pu se glisser lors de la création du STI. Trois types de contrôle sont effectués sur le STI créé :

1. Un contrôle de présence des entités permettant de vérifier la présence des entités obligatoires tels que les OP, l'univers des uah, les paramètres du STI, les paquets de règles instances, les différents réseaux de prérequis ;

2. Un contrôle de compte permettant de vérifier si l'auteur a respecté les contraintes imposées par le système quant au nombre de modèles de réponses et modèles de solutions associés respectivement à une question ou à un exercice. Nous rappelons que le système exige au moins deux modèles de réponses pour chaque question posée (respectivement deux modèles de solutions pour chaque exercice proposé);

3. Un contrôle de compatibilité entre la base de documents et les paramètres du STI permettant de vérifier si les paramètres indiqués par l'auteur correspondent vraiment aux contenus des différentes entités saisis.

6.5 Conclusion

Dans ce chapitre nous avons décrit la conception du mode auteur coopératif CamHits. Ce permet à des auteurs géographiquement distants de coopérer pour produire un STI selon un canevas prédéfini dans le système. Le STI produit, réside sur un serveur et peut donc être utilisé par différents apprenants à distance.

Le mode auteur CamHits est conçu selon une architecture logicielle centralisée basée sur le concept client-serveur. Il permet à plusieurs auteurs de se connecter à une session de travail caractérisée par un espace de coopération et une politique de contrôle. L'espace de coopération est représenté par un ensemble de composants structurés (uah, réseaux de prérequis, paramètres du STI, règles tutorielles) et des outils d'édition et de communication. La politique de contrôle gère la participation des utilisateurs à la session de travail et la négociation du droit d'intervention sur un composant du STI. Nous avons choisi une architecture modulaire et donc évolutive, dans le sens où elle facilite la conception de nouveaux modules et leur intégration de manière incrémentale.

Le prochain chapitre est consacré à l'implémentation d'un prototype de notre système et à une présentation de quelques fenêtres lors de l'expérimentation de nos outils implémentés.



Chapitre 7

Prototypage et utilisation

7.1 Introduction

Dans la phase de conception de notre système nous avons tenu à ce que notre architecture soit totalement indépendante de toute technologie d'implémentation. Le marché des langages de programmation sur le web et les stratégies à adopter sont tellement nombreuses qu'il est parfois difficile de faire un choix, voici quelques exemples d'architectures envisageables [Chaleat & al 05]:

- Applet Java : Le système réside entièrement dans une applet Java. Cette applet pourra être téléchargée par un apprenant en visitant l'URL spécifique, puis exécutée dans le poste client (poste de l'apprenant) ;
- Architecture CGI : Toutes les fonctionnalités du système résident sur le serveur (dans un script CGI écrit par exemple en C, Perl, C++) et l'interaction de l'apprenant avec le programme CGI se fait via un navigateur web standard (comme Internet Explorer ou Mozilla FireFox) qui envoie la requête de l'apprenant au serveur web pour lancer le programme CGI. Ce dernier exécute ce qui est demandé puis génère une page HTML qu'il retourne au serveur qui l'achemine à son tour au navigateur ;
- Architecture PHP/ASP/JSP : Le module PHP (ou ASP ou JSP) est intégré dans le serveur web lui-même. Lors d'une requête provenant du navigateur, le serveur exécute le code PHP (ou ASP ou JSP) faisant partie d'une page HTML et les résultats sont intégrés au flot HTML qu'il renvoie au navigateur.

C'est cette dernière architecture que nous avons adoptée pour le prototypage de TalHits. Nous avons en effet trouvé dans les cinq langages PHP, Html/JavaScript et Java/JSP une grande souplesse pour exprimer nos besoins de générer des pages web dynamiques adaptées à chaque apprenant.

Dans ce chapitre nous présentons l'architecture client-serveur de notre système auteur TalHits, en tenant compte des langages et des technologies que nous avons choisies.

7.2 Architecture classique client/serveur Web

L'architecture classique client/serveur Web peut être décomposée en trois parties distinctes [Delestre 00] :

- L'utilisateur active des liens hypertextes. Ces actions sont alors interprétées Les fichiers HTML contenant les informations qui vont être présentées à l'utilisateur.
- Le serveur HTTP, qui réceptionne les requêtes des clients, sélectionne le bon fichier (HTML, image, etc.) et le renvoie au client en le préfixant d'une information appelée "type mime".
- Le client qui est en fait un programme spécifique, appelé navigateur. A l'aide de ce navigateur comme des requêtes et par conséquent transmises au serveur HTTP destinataire.

Bien entendu ceci n'est qu'une vision simpliste du fonctionnement du World Wide Web, cette architecture peut être résumée par la (**Figure 7.1**).

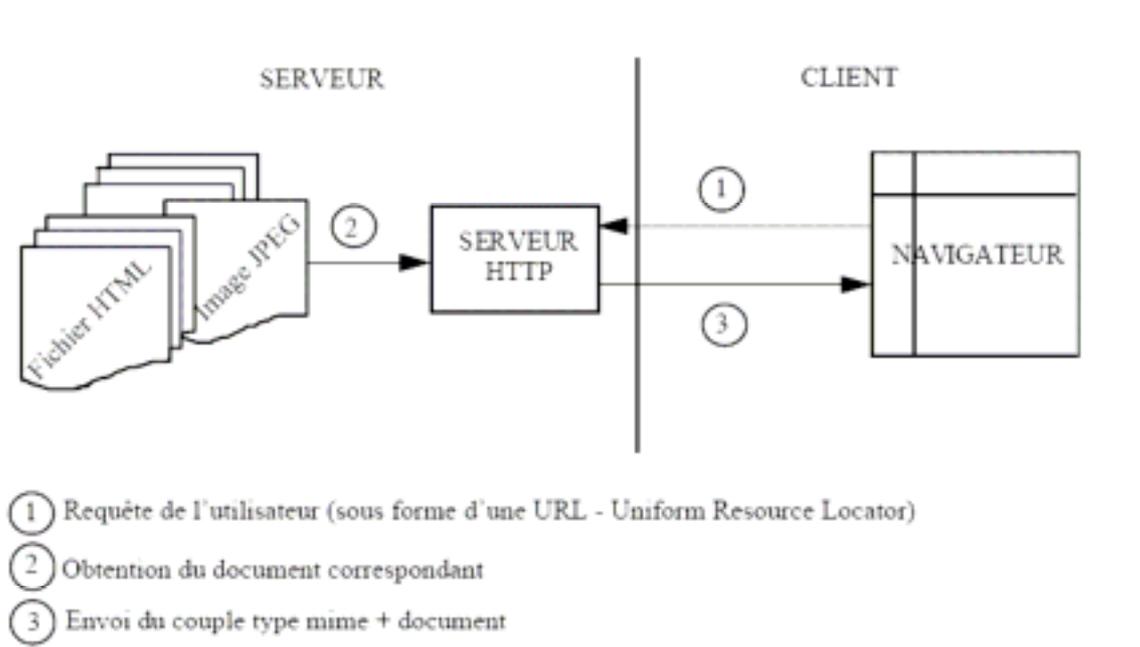


Figure 7.1. Architecture classique client-serveur web

7.3 Architecture basée sur les langages de script PHP/ASP/JSP

Cette architecture peut être vue comme une remise au goût du jour de l'architecture basé sur les programmes CGI. En effet, les services rendus par ces scripts sont équivalents à ceux rendus par les programmes CGI, Ils ont entre autres les trois avantages suivants sur les programmes CGI [Delestre 00]:

- Tout d'abord, alors qu'un programme CGI est un processus qui est lancé à chaque requête reçue, un script php/asp/jsp est un processus qui est persistant au niveau du serveur (il peut être lancé lors du lancement du serveur HTTP ou lors de la réception de la première requête).
- Ensuite les scripts lancent un processus fils par requête, alors que les programmes CGI lancent des processus indépendants. Cela permet par exemple de pouvoir créer une mémoire commune à toutes les requêtes ou encore permettre aux processus de communiquer entre eux.

L'architecture d'un serveur Web utilisant les scripts peut être résumée par le schéma de la (Figure 7.2).

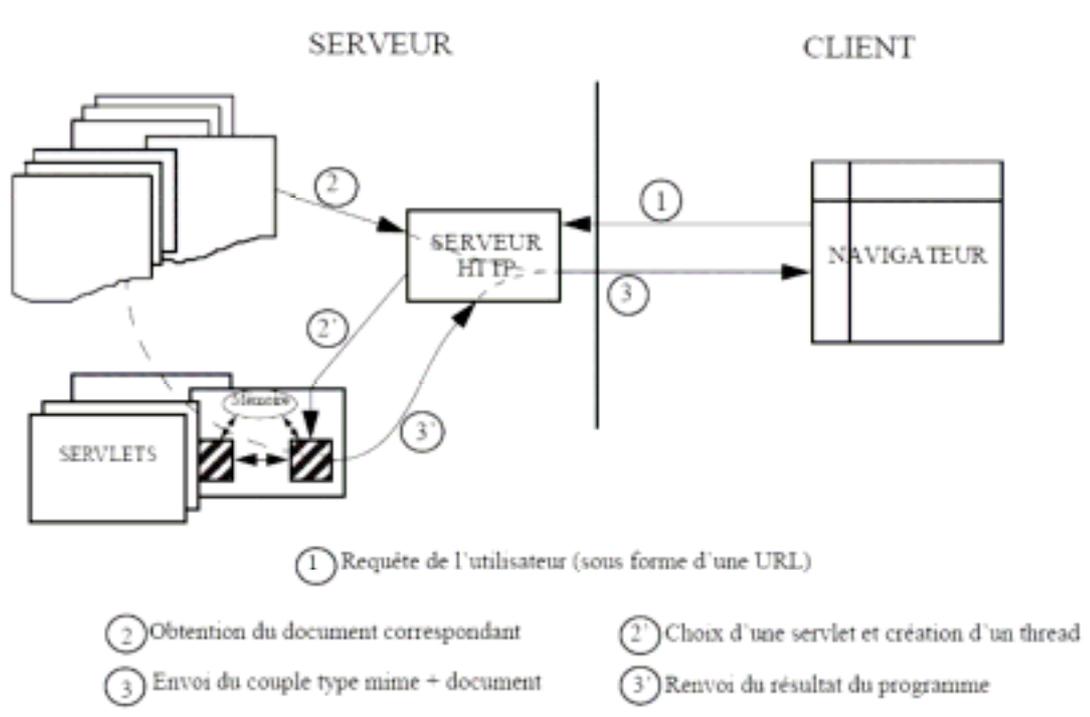


Figure 7.2. Architecture client serveur utilisant des langages de script

7.4 Choix du serveur de bases de données

Dans l'élaboration du prototype de TalHits, nous avons besoin de sauvegarder, de rendre persistant les objets créés par les différents usagers du système. On a en effet besoin de sauvegarder les différents modèles du domaine (cours, réseaux de prérequis, paramètres du STI, etc.) et les résultats des apprenants au cours d'une session d'apprentissage. L'utilisation d'une base de données semble alors nécessaire, puisque c'est sur cet outil que l'on va déporter tous les problèmes de concurrence d'accès, de réplication de l'information et de recherche efficace d'informations précédemment sauvegardées.

Les bases de données les plus répandues sont les bases de données relationnelles. Apparues dans les années 80, elles reposent sur le modèle relationnel qui lui-même repose sur la théorie des ensembles et fournit un langage ensembliste pour manipuler les relations ou les tables de la base de données : SQL est le plus connu de ces langages. Nous avons donc décidé d'utiliser une base de données relationnelle, à notre avis parmi les plus utilisées dans le contexte du web : MySQL.

MySQL représente un véritable bond en avant pour le système de gestion de bases de données open source le plus populaire au monde [Chaleat & al 05]. On sait que MySQL est depuis des années la base de données de prédilection lorsqu'il s'agit de gérer des sites web à fort volume et des applications de bases de données embarquées. La version 5.0, qui introduit de nouvelles fonctionnalités tout à fait exceptionnelles, va encore accélérer son adoption par les entreprises. Les progrès qu'elle apporte dans le domaine du développement applicatif, du traitement des transactions, de l'intégrité des données et de la facilité d'administration placent le serveur de base de données MySQL sur un pied d'égalité avec des bases de données propriétaires coûtant de nombreuses fois son prix.

MySQL est un serveur de bases de données relationnelles SQL très rapide, multi-thread, robuste et multi-utilisateurs. MySQL est un logiciel libre développé sous double licence GPL et licence commerciale. Il a été conçu et est maintenu par une société suédoise, MySQL AB, fondée par deux Suédois et un Finlandais : David Axmark, Allan Larsson et Michael « Monty » Widenius.

MySQL fonctionne sur beaucoup de plates-formes différentes, incluant Linux, Mac OS, OS/2, Solaris, SunOS, Tru64 Unix, Windows 95, 98, NT, 2000 et XP.

Les bases de données MySQL sont accessibles en utilisant les langages de programmation C, C++, Delphi /Eiffel, Java, Perl, PHP, Python, et Tcl ; une API spécifique est disponible pour chacun d'entre eux. Une interface ODBC appelée MyODBC est aussi disponible.

Parmi les outils associés à MySQL, nous trouvons :

- PhpMyAdmin, un outil d'administration de bases de données MySQL écrit en php.
- MySQL administrator, logiciel d'administration de bases de données développé par MySQL AB.
- MySQL Query Browser; interface graphique pour maintenir la base et y faire des requêtes développé par MySQL AB.
- MySQL Workbench, logiciel de conception et de modélisation de base de donnée développé par MySQL AB.
- EasyPhp, une solution complète web (PHP, MySQL, APACHE) gratuite aussi bien pour Linux que pour Windows

7.5 Choix du serveur Web

Le terme serveur Web désigne (voir http://fr.wikipedia.org/wiki/Serveur_Web) :

- Un ordinateur tenant le rôle de serveur informatique sur lequel fonctionne un logiciel serveur HTTP ;
- Le logiciel serveur HTTP lui-même.

La plupart des ordinateurs utilisés comme serveur Web sont reliés à Internet et hébergent des sites Web du World Wide Web. Les autres serveurs se trouvent sur des intranets et hébergent des documents internes d'une entreprise, d'une administration, etc.

Un serveur HTTP ou démon HTTP ou HTTPd (HTTP daemon) ou (moins précisément) « serveur Web », est un logiciel servant des requêtes respectant le protocole de communication client-serveur (HTTP). Le protocole HTTP a été développé pour le Web et les pages Web sont en pratique toujours servies avec ce protocole. D'autres ressources du Web comme les fichiers à télécharger ou les flux audio ou vidéo sont en revanche fréquemment servies avec d'autres protocoles.

Les serveurs HTTP les plus utilisés sont :

- Apache HTTP Server de la Apache Software Foundation (voir www.apache.org) ;
- Internet Information Services de Microsoft (www.microsoft.com/WindowsServer2003);
- Sun Java System Web Server de Sun Microsystems (www.sun.com/software/products/) ;

Le plus populaire est Apache HTTP Server qui sert environ 64% des sites Web en mai 2006 selon Netcraft (**Figure 7.3**) (voir http://news.netcraft.com/archives/web_server_survey.html). Notre choix a donc porté sur ce serveur http.

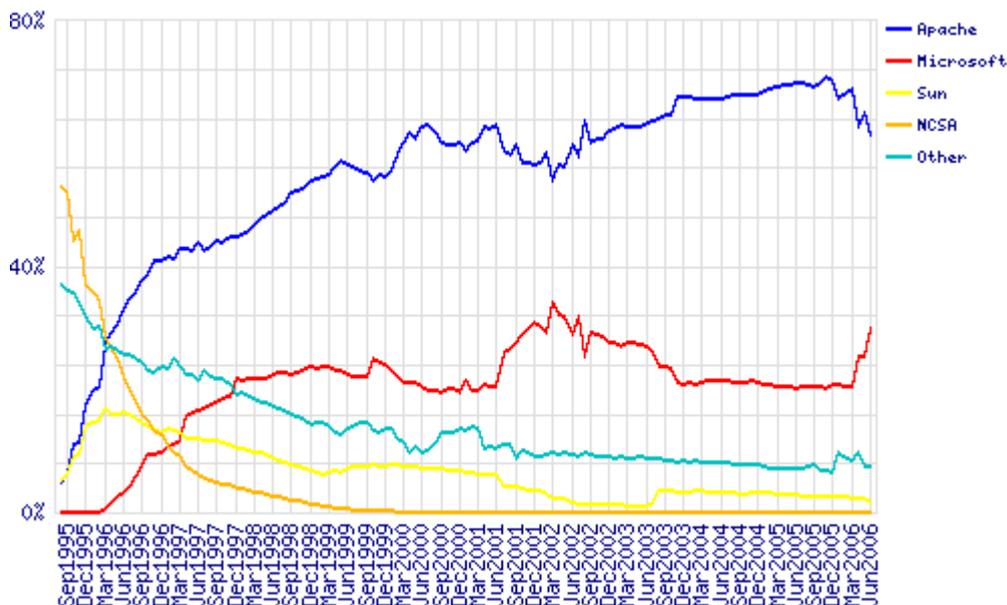


Figure 7.3. Graphes d'importance d'utilisation des serveurs web les plus utilisés

Le serveur Apache est apparu en avril 1995. Au début, il s'agissait d'une collection de correctifs et d'additions au serveur NCSA HTTPd 1.3, qui était dans le domaine public et le serveur HTTP alors le plus répandu. De cette origine, de nombreuses personnes affirment que le nom Apache vient de a patchy server, soit « un serveur rafistolé ». Par la suite, Apache a été complètement réécrit, de sorte que, dans la version 2, il ne reste pas de trace de NCSA HTTPd.

Depuis avril 1996, selon l'étude permanente de Netcraft, Apache est devenu le serveur HTTP le plus répandu sur Internet. En mai 1999, il faisait tourner 57 % des serveurs Web, et depuis début 2004, il se maintient à environ 65 % de part de marché.

La fondation Apache (Apache Software Foundation ou ASF) a été créée en 1999 à partir du groupe Apache (Apache Group) à l'origine du serveur en 1995. Depuis, de nombreux autres logiciels utiles au World Wide Web sont développés à côté du serveur HTTP. Depuis, Apache s'est illustré pour devenir le meilleur des serveurs HTTP sur le marché du net.

Apache fonctionne principalement sur les systèmes d'exploitation Unix (GNU/Linux, BSD et UNIX) et Windows. La version Windows n'est considérée comme stable que depuis la version 2 d'Apache.

Apache est conçu pour supporter de nombreux modules lui donnant des fonctionnalités supplémentaires : interprétation du langage Perl, PHP et JSP, serveur proxy, Common Gateway Interface, Server Side Includes, protocoles de communication additionnels, etc.

Les possibilités de configuration d'Apache sont une fonctionnalité phare. Le principe repose sur une hiérarchie de fichiers de configuration, qui peuvent être gérés indépendamment. C'est notamment utile aux hébergeurs Web qui peuvent ainsi servir les sites de plusieurs clients à l'aide d'un seul serveur HTTP. Pour les clients, cette fonctionnalité est rendue visible par le fichier « htaccess ». Parmi les logiciels aidant à la maintenance d'Apache, les fichiers de log peuvent s'analyser à l'aide de nombreux scripts et logiciels libres tels que Awstats ou Webalizer. Plusieurs interfaces graphiques facilitent la configuration du serveur.

7.6 Le mode auteur

7.6.1 Outils utilisés

Pour implémenter un prototype du mode auteur CamHits, nous avons utilisé le langage Java 2 SDK (version 1.4.2) et le langage de script JSP (Java Server Pages) [Tasso & Ermacore 04] sous le contrôle du serveur web Apache Tomcat et le SGBD MySQL. Le mode auteur est structuré de la manière suivante [Kraba & Ouazene 04] :

- Au niveau du serveur, nous trouvons trois composants majeurs :
 1. Le système de gestion de base de données MySQL qui permet de stocker toutes les informations que nous voulons persistantes.
 2. Le serveur HTTP Apache Tomcat (<http://java.sun.com/products/servlet/index.jsp>). Ce serveur est en charge de construire les cours qui sont présentés à l'apprenant. Pour se faire, il peut communiquer avec la base de données (via le SGBD) et avec l'applet éditeur fonctionnant sur le poste client
 3. Les scripts JSP qui constituent le mode auteur proprement dit.
- Au niveau du client, nous trouvons le navigateur MS Internet Explorer, le Java Runtime Environment (JRE version 1.4.2) et une applet permettant à un enseignant l'édition de texte et la communication avec le serveur HTTP.

Le langage de script JSP permet de générer des pages web dynamiques. Une page JSP est une page HTML contenant du code Java, elle porte l'avantage de séparation entre le code HTML et le

code Java. Concrètement avec les JSP, les parties statiques de la page HTML sont écrites en HTML et les parties dynamiques de la page sont écrites en Java.

Pour exécuter des JSP il faut un moteur JSP sur le serveur web, ce moteur construit une Servlet Java à partir de la page JSP (cette phase est parfois appelée la traduction de la JSP) puis compile et exécute la Servlet (**Figure 7.4**).

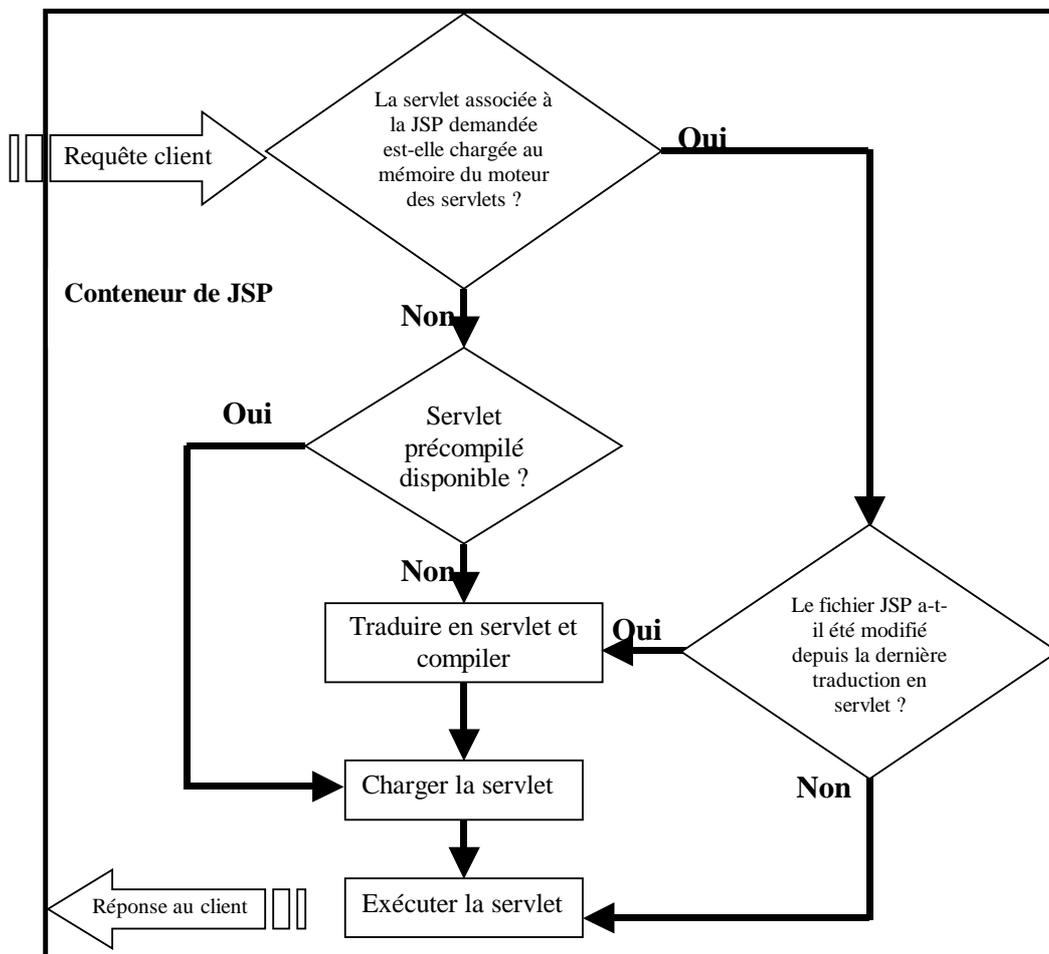


Figure 7.4. Algorithme d'exécution de la JSP

L'importance de Java [Puybaret 03] dans la programmation en réseau vient du fait qu'il constitue un langage interprété et non compilé. Supposons que sur l'un des ordinateurs du réseau, il existe un programme que l'on souhaite exécuter aussi sur trois autres. Il faudrait donc dans ce cas, non pas télécharger le fichier exécutable, mais le fichier source et le et le recompiler sur chaque machine à part avant de pouvoir l'exécuter. Mais puisque le programme est écrit en java, il suffirait que chaque machine télécharge le même fichier compilé (Byte Code) et qu'elle le passe à travers son interpréteur Java, qui se charge alors du bon fonctionnement du programme.

En effet, Java repose sur une machine virtuelle, qu'on appelle Java Virtual Machine (JVM). Le fichier source Java (.java) est compilé en un fichier binaire (.class) exécutable uniquement sur cette machine virtuelle, la machine virtuelle se charge alors de transformer les instructions JVM en des instructions natives de la plate-forme d'accueil.

Le code intermédiaire produit est indépendant des plates-formes : il pourra être exécuté sur tous types de machines et systèmes pour peu qu'ils possèdent l'interpréteur de code Java. Même la programmation graphique réseau et système est totalement indépendante des machines et systèmes. Les problèmes de "portage" qui occupent une partie non négligeable du temps de développement des logiciels disparaissent ainsi.

Conçu pour développer des applications en réseaux, les manipulations des objets distants ou locaux en Java se font de la même manière. Par exemple, l'ouverture d'un fichier local ou distant se programme de manière identique. La classe Socket permet la programmation d'applications Client/Serveur de manière aisée.

7.6.2 Fonctionnalités de base du mode auteur CamHits

Le mode auteur se présente sous la forme d'un site web classique dont la première page demande aux enseignants de se déclarer au niveau du système (grâce à la saisie d'un identifiant et d'un mot de passe) puis de préciser le mode de connexion (**Figure 7.5**).

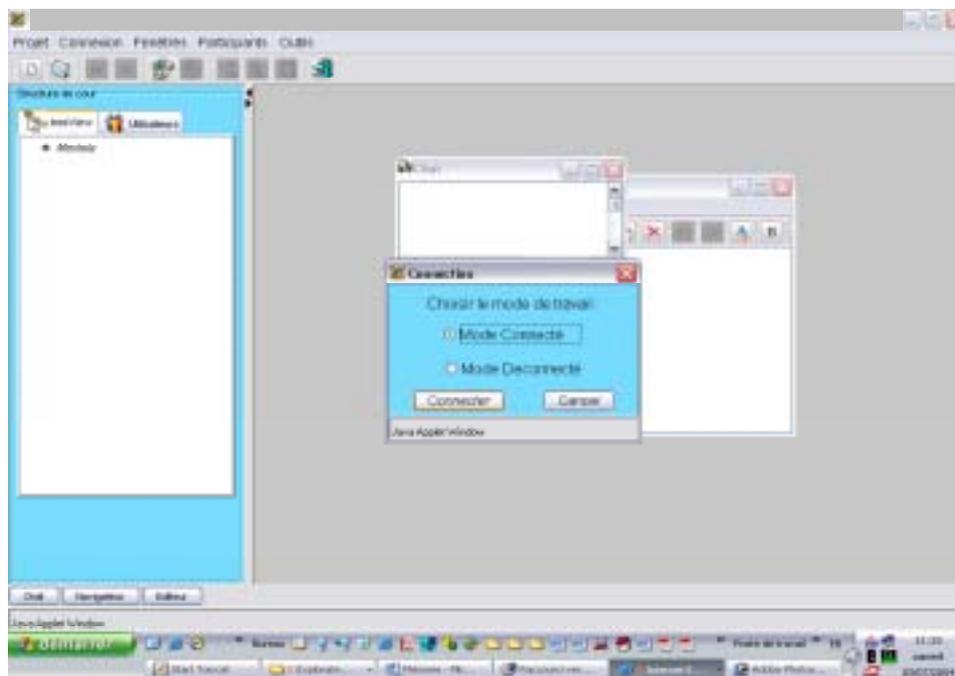


Figure 7.5. Boîte de dialogue du choix du mode de connexion

Au départ, le chef de projet va créer un nouveau projet (document), et le décompose en parties puis saisit le résumé du module (**Figure 7.6**).

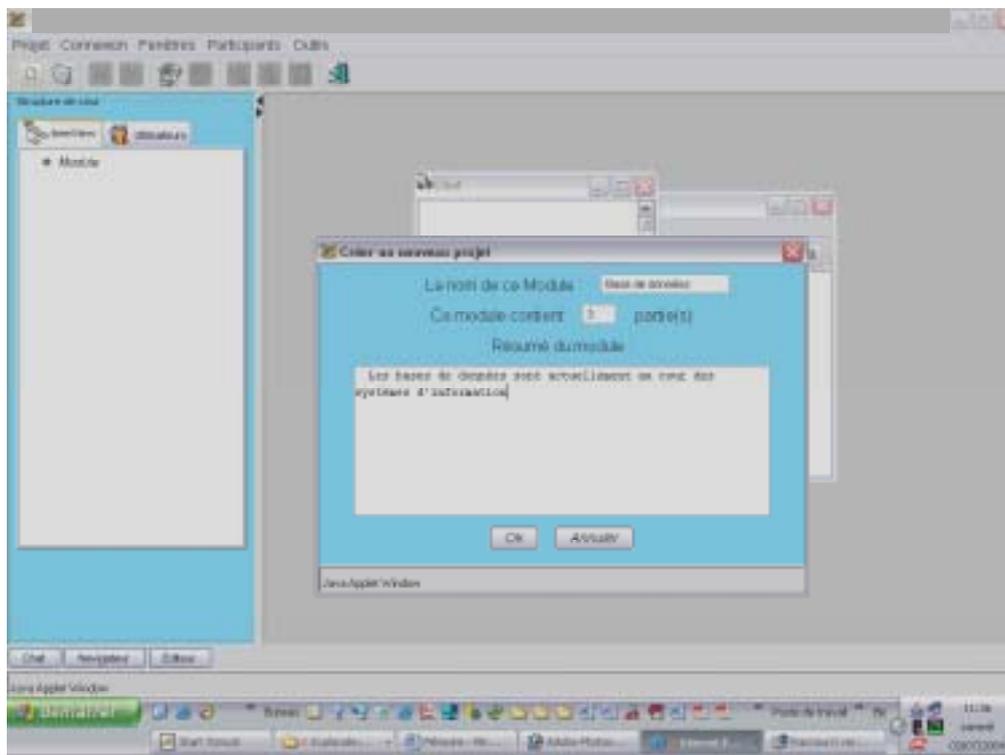


Figure 7.6. Créer un nouveau projet

Lorsqu'un nouveau participant vient de se connecter, le document sera chargé au niveau de son site, la structure logique de ce document sera alors affichée, et l'état de chaque nœud du document apparaît au niveau de l'interface (**Figure 7.7**).

Le mode de notification d'un nœud est reflété par l'aspect graphique de la marque qui lui est associé (une icône) dans la structure logique et qui indique également le rôle que joue le participant sur ce nœud. L'icône associée à chaque nœud reflète son état d'allocation, par exemple l'icône  qui apparaît sur l'interface d'un auteur représente un nœud déjà alloué par un autre auteur et l'icône  apparaît sur l'interface de l'auteur qui l'a alloué (**Figure 7.8**). Un auteur ne peut travailler évidemment que sur les nœuds qui appartiennent au fragment auquel il est affecté (partie chapitre, etc.) et que sur un nœud alloué. Il établit un engagement et le serveur répond selon l'état de ce nœud.

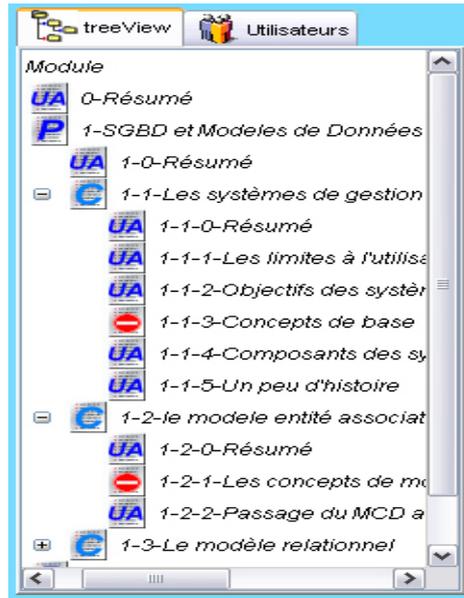


Figure 7.7. La structure logique du document

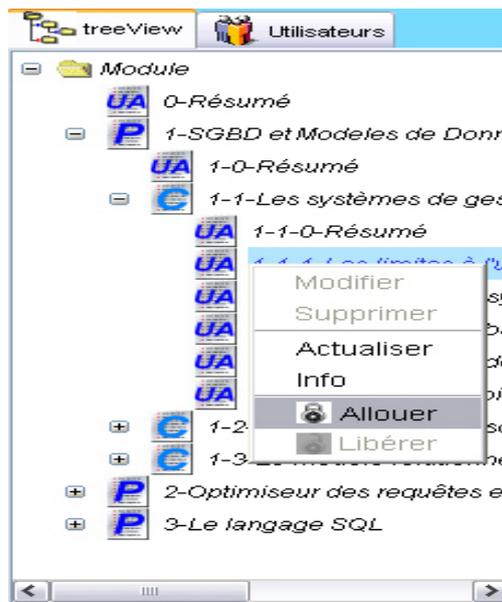


Figure 7.8. Allocation d'un nœud

Pour consulter le contenu d'un nœud il suffit de cliquer deux fois sur ce nœud. Le contenu s'affiche alors dans une fenêtre d'exploration (Figure 7.9).

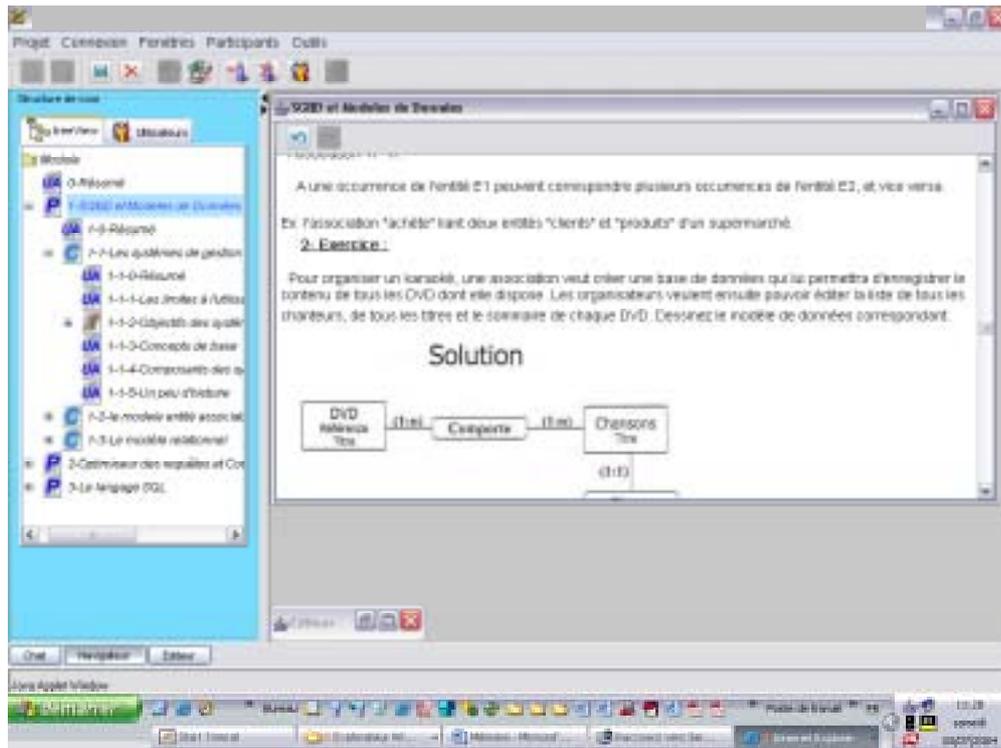


Figure 7.9. L'explorateur de l'éditeur CamHits

L'entrée et le départ d'un membre du groupe à une session de travail sont explicitement signalés aux autres membres. Par conséquent, les auteurs connaissent à tout instant la liste des personnes présentes en session. Chaque auteur peut consulter la liste des co-auteurs à chaque moment, la liste apparaît comme indiqué en (Figure 7.10) où chaque identificateur d'un auteur est accompagné d'une icône qui désigne son état actuel de connexion.

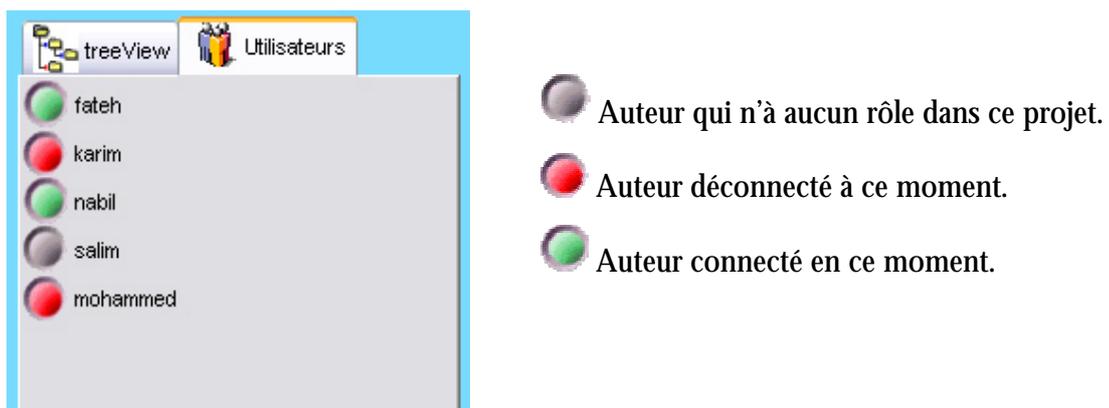


Figure 7.10. L'état de connexion des auteurs à un moment donné

Au cours d'une session de travail, l'auteur principal (chef de projet) est le seul auteur qui a le rôle de spécifier les rôles des autres co-auteurs (**Figure 7.11**). Au cours d'une session de travail, les auteurs peuvent échanger des messages pour discuter de l'état d'avancement du travail ou négocier le blocage des différents fragments et ce, grâce aux outils de messagerie et de forum.



Figure 7.11. Affectation des rôles aux co-auteurs

Lorsqu'un auteur termine son travail avec un nœud, il peut libérer ce dernier (**Figure 7.12**).

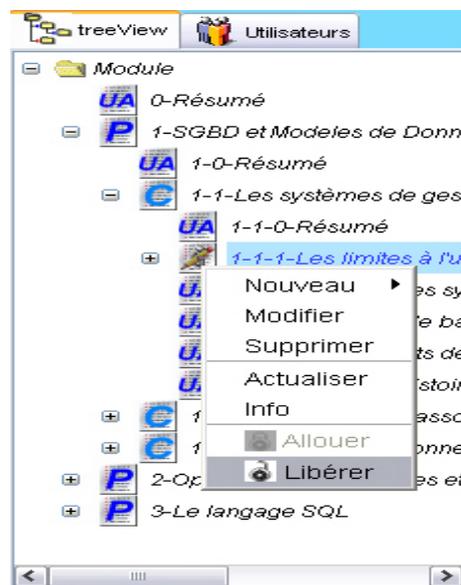


Figure 7.12. Libération d'un nœud

Si un auteur désire modifier le contenu d'un nœud (un média de type « texte »), il suffit de cliquer deux fois sur ce nœud, le contenu sera affiché dans la fenêtre éditeur (**Figure 7.13**).

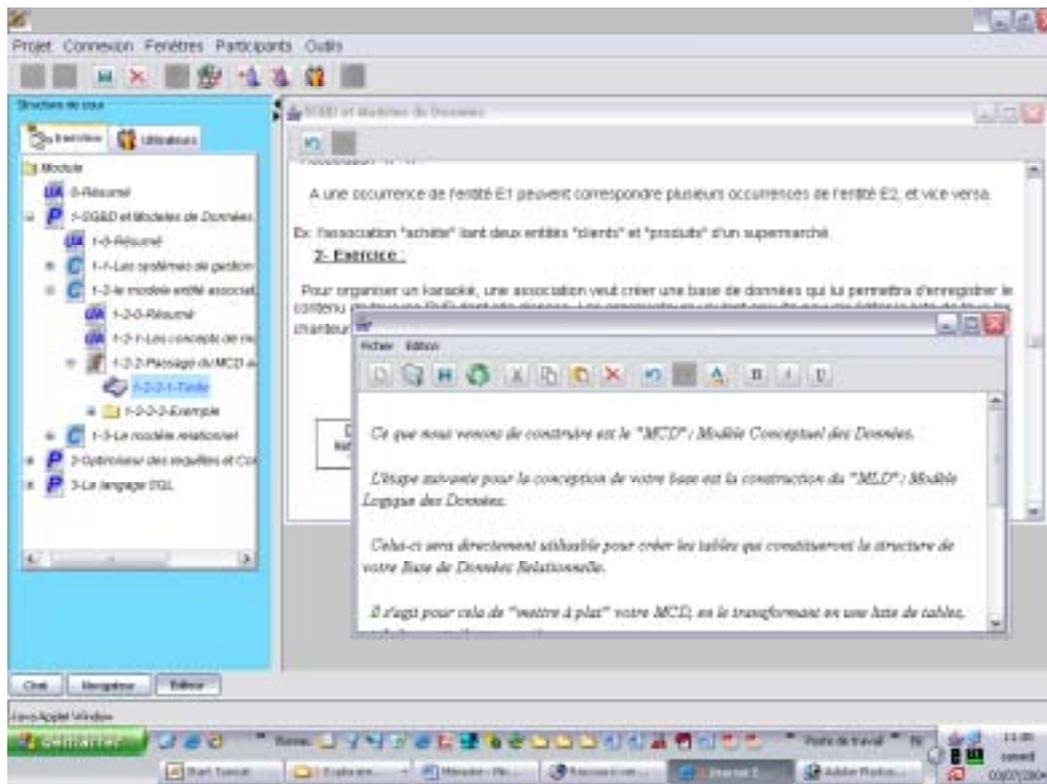


Figure 7.13. Fenêtre d'édition de texte

Lors de la validation d'un rédacteur de son travail, le contenu et la structure sont transmis au serveur qui prend alors la charge d'informer les autres participants. L'auteur peut évidemment ajouter un autre média de type (image, son, etc.) (Figure 7.14).

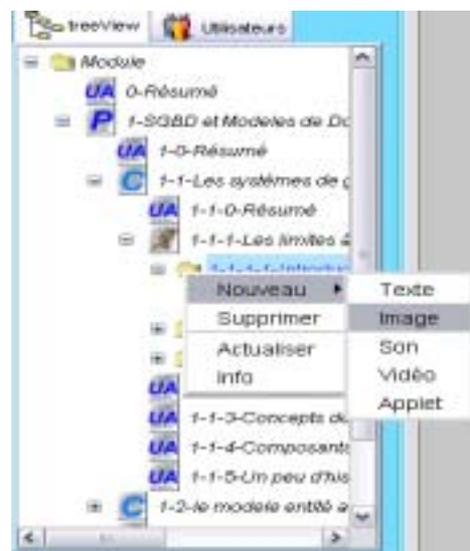


Figure 7.14. Ajout d'un média (image, son, etc.)

7.7 Le mode apprenant

7.7.1 Outils utilisés

Pour implémenter un prototype du mode apprenant Hits, nous avons utilisé le langage Php (version 4) sous le contrôle du serveur web Apache et le SGBD MySQL. Le mode apprenant est structuré de la manière suivante [Djaghrouri & al 04] :

- Au niveau du serveur, nous trouvons deux composants majeurs :
 1. Le système de gestion de base de données MySQL qui permet de stocker toutes les informations que nous voulons persistantes.
 2. Le serveur HTTP Apache qui est en charge de construire les cours qui sont présentés à l'apprenant. Pour se faire, il peut communiquer avec la base de données (via le SGBD MySQL).
 3. Les scripts Php qui constituent le mode apprenant proprement dit.
- Au niveau du client, nous trouvons le navigateur MS Internet Explorer.

7.7.1.1 Présentation du langage de scripts PHP

Les pages web qui circulent entre les différents sites connectés à Internet, sont produites en général à l'aide du code HTML. Ce code sera ensuite interprété au niveau des clients par le navigateur.. Le code HTML à son tour peut être produit de deux façons :

1. La production statique du code HTML : Dans ce cas le code est soit directement saisi par le programmeur, soit il est généré automatiquement par un éditeur spécialisé tel que : Microsoft Front Page, Netscape Composer, etc.
2. La production dynamique du code HTML : Dans ce cas, le code HTML est le résultat d'exécution d'un programme au niveau du serveur. L'exécution de tels programmes du côté serveur, intègre généralement des fonctions de traitement de formulaires ou des données d'une base de données externes à la page en lançant des requêtes destinées au SGBD implanté au niveau du serveur. Les programmes qui permettent la génération dynamique des pages Web peuvent être écrits en plusieurs et différents langages de programmation dont le langage Php (créé par Rasmus Lerdorf en 1994) qui est le plus utilisé actuellement sur le Net.

Le langage PHP (Personal Home Page au début et Hypertext Preprocessor actuellement) est un langage de script coté serveur, et dont le code est directement immergé au milieu du code HTML de la page Web à générer d'une manière dynamique [Chaleat & al 05]. En utilisant PHP sur un serveur web, les pages ou les sites hébergés sur ce serveur deviennent une véritable application web interactive au lieu d'un ensemble de pages statiques plus au moins actualisées.

Pourquoi avons-nous choisi Php pour prototyper notre mode apprenant ? La réponse à cette question est que nous avons trouvé en ce langage un outil très puissant pour exprimer nos besoins, en voici quelques caractéristiques de ce langage :

- PHP est un produit Open Source tout en étant indépendant des plates formes ;
- PHP a été conçu pour fonctionner sur le web. La connexion et l'interrogation d'une base de données est une tâche simple pouvant être accomplie en deux ou trois lignes de code, car le moteur de scripts PHP est parfaitement optimisé pour les temps de réponse nécessaires à des applications web ;
- La syntaxe de PHP s'inspire largement du langage C, PHP est un langage multi plate-forme. Il a été porté sur de nombreuses stations UNIX, telles que Linux, où il fonctionne aussi bien que sur des machines Windows ;
- PHP offre au programmeurs plus de 1200 fonctions utilisables dans des applications très variées, il couvre pratiquement tous les domaines en rapport avec le web (fonctions HTTP, accès au systèmes de fichiers, fonctions de retouche et de génération dynamique des images avec la bibliothèque GD, utilisation des sockets Internet, support de services utilisant les protocoles tel que SNMP, POP3, etc.) ;
- PHP s'interface à de nombreuses bases de données SQL, en offrant des fonctions dédiées pour la prise en charge directe des principaux systèmes de gestion de bases de données relationnelles. PHP prend notamment en charge l'accès au serveur de bases de données MySQL largement utilisé pour les applications web. Et en possédant des fonctions ODBC, PHP permet de s'interfacer d'une manière conviviale à toute base de données ODBC ;
- PHP est Orienté Objet, il offre une couche objet qui s'est intégrée au style procédural du langage. PHP offre un ensemble de constructions permettant aux scripts de définir les principaux concepts de la programmation orientée objet (les classes, les objets, les méthodes, l'héritage, etc.).

7.7.1.2 Mode de fonctionnement de PHP

Contrairement aux scripts CGI, le code PHP fait partie intégrante de la page HTML. Pour concevoir un script php, il suffit de se dire que l'on conçoit une simple page HTML qui constituera le résultat de retour du script. Donc, tout ce qui est de l'HTML "pur" sera retourné tel quel par le serveur. Ce script consiste simplement en la succession de :

- la balise de début `<? ;`
- des instructions écrites dans le langage php ;
- la balise de fin `?> ;`

PHP est intégré comme sous programme du serveur Web Apache. PHP fait ainsi partie intégrante d'Apache, il est ainsi toujours prêt à interpréter et à exécuter des scripts sur demande, sans réinitialisation. Cette variante fait l'économie de la procédure d'initialisation, ce qui se traduit par une rapidité de traitement sensiblement plus élevée que les programmes CGI.

7.7.1.3 Manipulations d'une base de données MySQL avec des scripts PHP

La manipulation de bases de données MySQL est effectuée en envoyant des requêtes à l'aide de scripts PHP. Voici les principales fonctions PHP qui nous permettent d'administrer une BDD MySQL [Chaleat & al 05] :

a) Connexion au serveur MySQL : Pour pouvoir envoyer des requêtes à MySQL, il faut en premier lieu établir une connexion avec le serveur MySQL en utilisant la fonction `mysql_connect` (toutes les fonctions php/mysql commencent par `mysql_`). Cette fonction appelle en arguments le nom de la machine hôte où se trouve le serveur MySQL, le nom de l'utilisateur autorisé et le mot de passe de cet utilisateur.

```
$connexion = mysql_connect("infres","mastmul","xcxcxc");
```

Cela tentera d'établir une connexion avec la base MySQL dont le serveur tourne sur l'hôte infres, sous le nom d'utilisateur mastmul. Une fois que la connexion est établie, on peut utiliser les fonctions de manipulation des bases de données. Les connexions peuvent être persistantes aux bases de données, c'est à dire des connexions qui ne se referment pas à la fin du script.

b) Déconnexion de la base : La déconnexion se fait grâce à la requête `mysql_close($connexion)` ;

c) Opérations sur les bases de données : PHP dispose de toute une série de fonctions permettant d'utiliser MySQL comme la création d'une BDD, la création d'une table ou d'un enregistrement, la suppression d'une table ou d'un enregistrement, etc. Exemples : `(Mysql_create_db(), Mysql_drop_db(), Mysql_fetch_array(), Mysql_select_db(), etc.`

7.7.2 Fonctionnalités de base du mode apprenant Hits

Le mode apprenant Hits se présente sous la forme d'un site web classique dont la première page demande aux enseignants de se déclarer au niveau du système (grâce à la saisie d'un

identifiant et d'un mot de passe) sinon de s'inscrire si c'est la première fois qu'on utilise le système.

Hits est doté de trois fonctionnalités principales : apprentissage libre (mode information), apprentissage guidé (mode formation) et outils de communication (messagerie, forum et usagers en ligne). D'autres fonctionnalités sont mises à la disposition de l'apprenant afin de satisfaire ses besoins en aide comme des outils du genre (calendrier, liens, calculatrice, recherche, impression, etc.). Dans ce que suit nous présenterons quelques fenêtres de Hits classées selon le scénario de déroulement d'une session d'apprentissage.

7.7.2.1 La page d'accueil

La page d'accueil du mode apprenant Hits est illustrée dans la (Figure 7.15).

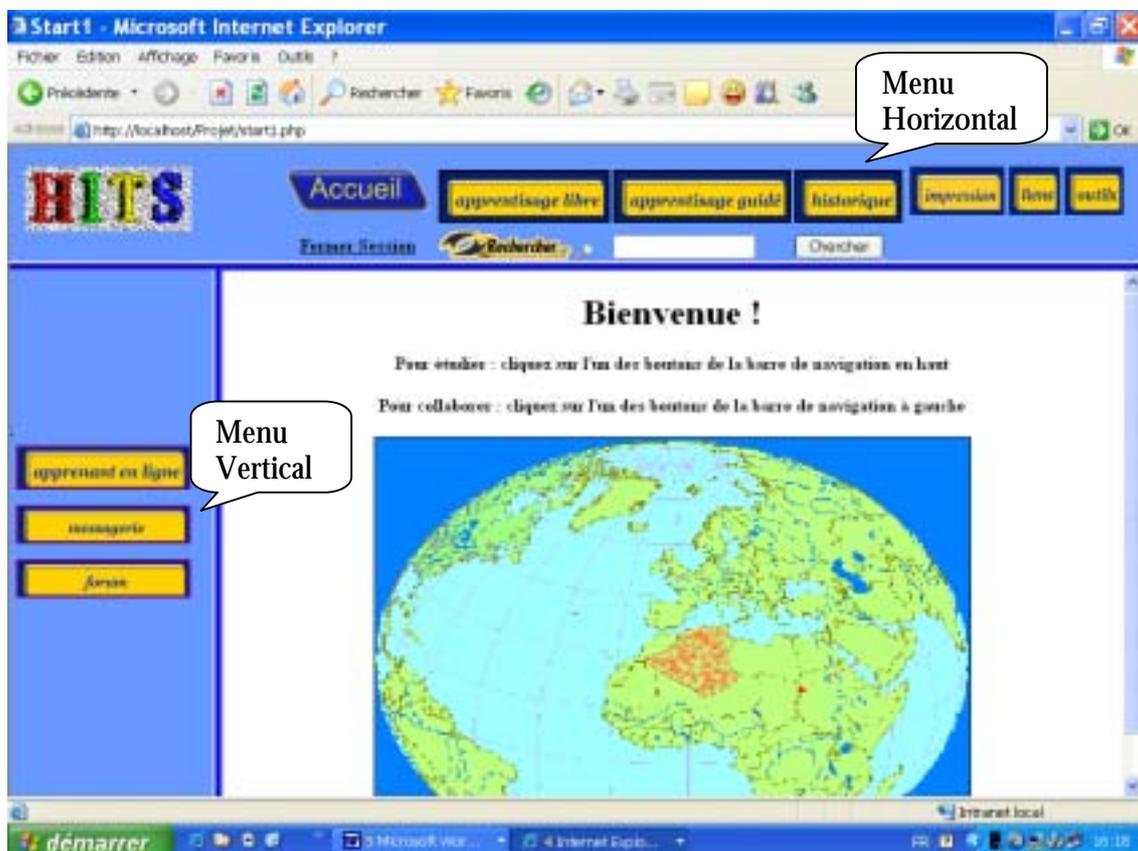


Figure 7.15. Page d'accueil du mode apprenant

7.7.2.2 Apprentissage libre :

Ce mode permet à l'apprenant de consulter librement le module d'enseignement sans aucun guidage du système et sans aucune évaluation des connaissances de l'apprenant (Figure 7.16).



Figure 7.16. Apprentissage libre (mode information)

7.7.2.3 Apprentissage guidé

Ce mode donne la possibilité à l'apprenant d'être guidé et évalué durant sa session d'apprentissage. Dans ce mode, deux cas sont envisageables (Figure 7.17), soit que l'apprenant est déjà inscrit soit qu'il n'est pas encore inscrit.



Figure 7.17. Dialogue d'ouverture d'une session d'apprentissage

a) Cas d'un apprenant déjà inscrit

L'apprenant qui est déjà inscrit et désirant continuer sa formation doit remplir le formulaire d'entrée à la session présenté dans la (Figure 7.17). Après la validation, le système charge le modèle de l'apprenant concernant le module choisi et détermine la dernière uah acquise dans la session précédente puis il lui affiche une fenêtre dans laquelle il lui indique le dernier point d'arrêt dans la dernière session et l'invite à continuer à partir de ce point.

b) Cas d'un apprenant qui n'est pas encore inscrit

L'apprenant qui n'est pas inscrit doit d'abord s'inscrire en cliquant sur le lien er sur le lien indiqué à la fin de la figure V.5 (Cliquez Ici), ce qui invoque l'affichage du formulaire d'inscription présenté dans la (Figure 7.18). Lorsque la validation ne reflète aucune erreur, un nouvel enregistrement est ajouté à la table **Apprenant**.

The image shows a web form titled "Inscription" with a blue header. Below the title, there is a blue bar with the text "Inscription". Underneath, there are two lines of text: "Remplissez minutieusement le formulaire suivant." and "Attention ! tous les champs sont obligatoires." The form itself is a light blue box containing several input fields: "Login" (s_talhi), "Mot de passe" (masked with dots), "Confirmer mot de passe" (masked with dots), "Nom" (TALHI), "Prénom" (Said), "Email" (s_talhi@yahoo.fr), "Photo" (c:/mephoto.gif) with a "Parcourir..." button, and "Module" (systeme expert) with a dropdown arrow. An "Envoyer" button is at the bottom right. A blue arrow points to the bottom center of the form.

Figure 7.18. Formulaire d'inscription

Une fois que l'apprenant est inscrit, le système négocie avec lui, le point d'entrée dans le cours. Pour le choix du point d'entée dans le cours, le système présente à l'apprenant trois étapes de choix dont chacune correspond à un niveau d'objectif du module d'enseignement {partie (OPG), chapitre (OPS), uah (OPO)}. A chaque étape, le système offre à l'apprenant la possibilité de commencer à partir de la partie ou le chapitre courant en cliquant sur le bouton "**Commencer**" (Figure 7.19) sinon il l'invite à préciser plus le niveau hiérarchique en cliquant sur le bouton "**Préciser plus le niveau**" en lui permettant la consultation de la table des matières à tout moment en cliquant sur le bouton "**Table des matières**".

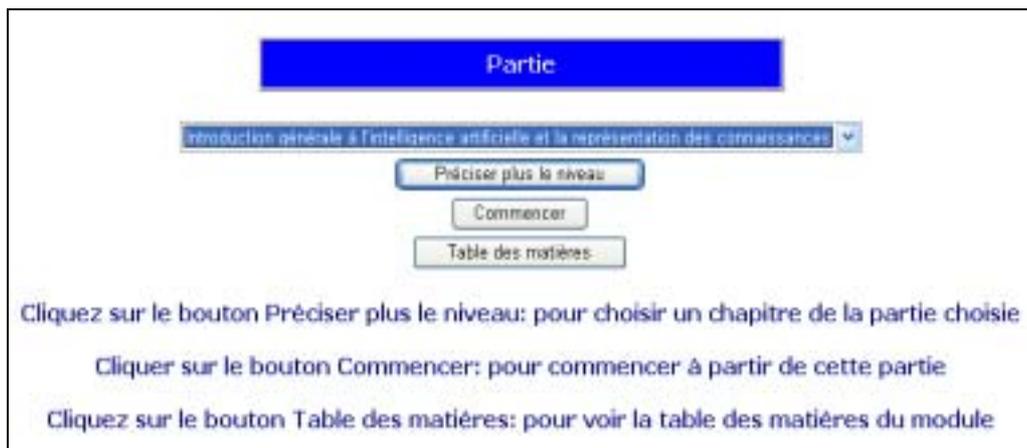


Figure 7.19. Choix du point d'entrée dans le cours

Après la confirmation du point d'entrée le système élabore un plan de la session d'apprentissage comme décrit dans le chapitre 5. Généralement le système cherche à connaître le niveau de l'apprenant par un ensemble de prétests (Figure 7.20) et une fois ce niveau déterminé, il passe à la planification du cours en se basant sur les réseaux de prérequis.

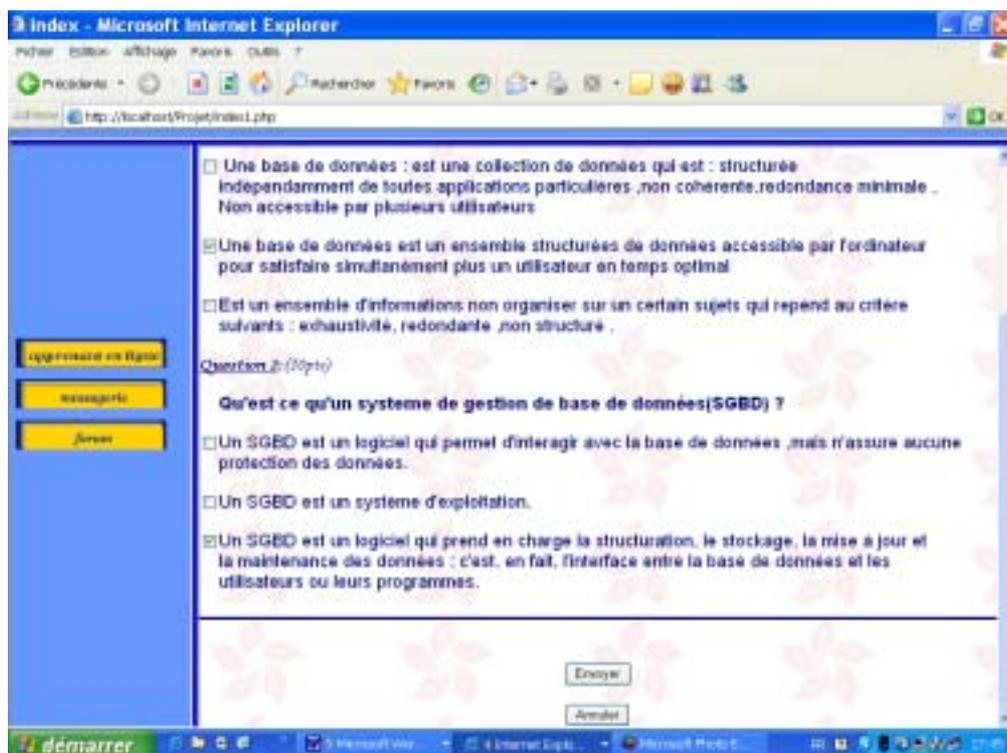


Figure 7.20. Prétest sur la partie choisie par l'apprenant

Quant l'évaluateur juge que le prétest est réussi, le résultat est retourné au planificateur qui continue la session d'apprentissage avec l'apprenant (présenter par exemple la première uah de la partie choisie, c'est à dire le résumé de la partie choisie comme illustrée dans la **Figure 7.21**).



Figure 7.21. Présentation de la première uah de la partie choisie

7.7.2.4 Communication entre usagers

Afin de permettre aux apprenants de collaborer entre eux ou bien de contacter un enseignant, deux outils de communication asynchrone sont fournis : la messagerie électronique et le forum de discussion. Aussi, à tout moment, l'apprenant peut consulter la liste des membres du groupe et ceux qui sont connectés au même moment et ce en cliquant sur le bouton "apprenants en ligne" (**Figure 7.22**). En cliquant sur l'adresse email de cet usager, il peut lui envoyer un message. La même fonctionnalité peut être obtenue en cliquant sur le bouton "Messagerie" (**Figure 7.23**).

Enfin, le bouton "Forum" permet à l'apprenant de consulter le forum de discussion en envoyant des questions, répondre à des questions d'autres usagers ou tout simplement lire les discussions des autres sur les différents sujets abordés (**Figure 7.24**).

	Photo	Nom	Prénom	Email	Module
		DJAGHROURI	Ola	ola@hits.com	base de données
		DJEBBAR	Nadia	dida@hits.com	base de données
		HACHACHENA	Samia	samia@hits.com	base de données
		TALHI	Said	TALHITS@hits.com	base de données

Figure 7.22. Affichage des apprenants en ligne

Envoyer un message

A : ola @hits.com

Objet : Aide

Message :

Bonsoir Ola,
 Peut tu m'envoyer les cours
 complémentaires au module bases
 de données que tu a téléchargé
 hier sur internet.
 Merci d'avance.
 &+

Envoyer

Retour Fermer session

Figure 7.23. Utilisation de la messagerie électronique



Figure 7.24. Accès au forum de discussion

7.8 Conclusion

Dans ce chapitre nous avons présenté l'implémentation d'un prototype du système auteur TalHits à travers modes d'exploitation : le mode auteur et le mode apprenant. Les fonctionnalités essentielles de chaque mode sont alors exposées. Pour chaque mode, nous avons exposé en premier lieu les différents outils employés pour l'implémentation ainsi que les motivations de nos choix. Nous avons alors opté pour des outils de grande audience à l'heure actuel dans le développement des applications web : {Java, JSP, Serveur Apache Tomcat, SGBD MySQL} pour le mode auteur et {Php/MySQL, Serveur Apache, SGBD MySQL} pour le mode apprenant. Nous avons exposé en deuxième lieu, les fonctionnalités de base offertes dans chaque mode à travers des exemples concrets de fenêtres et de boîte de dialogue.

Le prototype étant expérimenté avec deux modules d'enseignement enseignés en quatrième année du cycle ingénieur en informatique à l'université de Batna qui sont : le module « bases de données » et le module « systèmes experts ». Evidemment, nous ne pouvons pas prétendre avoir implémenté toutes les fonctionnalités décrites dans les chapitres 4, 5 et 6. Ceci nécessitera beaucoup plus d'efforts et de temps. Nous espérons les compléter très prochainement.



Conclusions et perspectives

Conclusions

Au terme de cette thèse, nous tentons de tirer les conclusions du travail présenté et de discuter les perspectives d'amélioration de TalHits. Pour ce faire, nous établissons tout d'abord un bref bilan des thèmes abordés, nous faisons ensuite le point sur notre contribution, puis nous indiquons quelques pistes d'extension du système.

Tout au début, nous avons souligné que cette thèse s'insère dans le cadre des systèmes auteurs pour l'enseignement à distance. Dans ce domaine, nous avons classé les systèmes auteurs existants en deux catégories : les systèmes auteurs mono-usagers (utilisés par un utilisateur individuellement) et les systèmes auteurs coopératifs (utilisés par plusieurs utilisateurs géographiquement distants, à travers un réseau informatique). Nous avons alors situé notre travail dans la deuxième catégorie de systèmes, c'est à dire les systèmes auteurs coopératifs, un sujet d'actualité née avec l'expansion du réseau des réseaux internet.

Un système d'EAD est généralement réalisé pour satisfaire des besoins bien précis et il est destiné à être exploité dans un contexte bien précis également. La majorité des systèmes offrent des systèmes hypermédias qui consistent à tourner des pages html en suivant des liens préconçus. En ce qui nous concerne, nous avons mentionné que le système auteur TalHits est destiné aux enseignants et étudiants universitaires pour compléter les enseignements présentiels suivies à l'université. Afin qu'il soit facilement utilisable par la majorité des enseignants et des étudiants, nous avons proposé que le système soit basé sur la pédagogie par objectifs, une approche qui est d'une part largement indépendante du domaine à enseigner et qui est, d'autre part, connue et utilisée par la majorité des enseignants et des étudiants universitaires. Deux caractéristiques sont intégrés dans le système : l'intelligence dans le processus d'enchaînement des uah et la coopération dans le processus de production des objets pédagogiques.

Nos objectifs de travail étant fixés, nous avons ensuite établi un bref bilan de l'état de l'art dans le domaine de l'enseignement médiatisé par ordinateur en vue de fixer le vocabulaire et les concepts utilisés dans ce domaine et de dégager les architectures des systèmes réalisés aussi bien

dans le cadre de l'EIAO que dans celui de l'EAD. Dans le deuxième chapitre, nous nous sommes intéressés à l'intégration des systèmes hypermédias et des tuteurs intelligents pour assurer le guidage des apprenants à distance et dans le troisième chapitre nous nous sommes concentrés sur les systèmes auteurs traditionnels et leur évolution sur le web.

En résumé, nous avons constaté que dans le contexte de l'EAD, les concepts jusqu'alors établis en EAO, EIAO classiques n'ont pas changé. Un environnement d'enseignement à distance possède les mêmes objectifs qu'un didacticiel classique ou un STI, à savoir, assister l'apprenant en phase d'apprentissage. Ce qui a changé c'est le critère spatial et temporel. L'apprenant est sensé suivre ses cours à travers un réseau informatique et donc non contraint pour se déplacer à l'université pour le faire. Les données et programmes ne sont plus centralisées sur la machine de l'apprenant comme dans l'ère EAO/EIAO, mais se trouvent répartis un peu partout dans le monde et l'apprenant ne dispose que d'un navigateur pour accéder aux différentes ressources. Une autre dimension est donc ajoutée au concepteur de ces environnements, celle de la programmation client-serveur.

L'analyse de l'état de l'art étant établie, nous avons ensuite présenté notre modèle de système auteur. Etant conscient que la réalisation de systèmes auteurs est une tâche difficile à entreprendre, nous avons commencé par baliser notre chemin par dresser un ensemble de besoins que notre système essaiera de satisfaire. Nous avons alors privilégié des STI fondés essentiellement sur la transmission de pages hypermédias en utilisant la pédagogie par objectifs.

Nous avons par la suite montré l'architecture logicielle de notre système qui consiste en un « mode auteur » et un « mode apprenant ». Ce dernier en particulier, consiste en un STI sur le web qui intègre les caractéristiques d'hypermédia et d'intelligence.

Les fonctionnalités générales de TalHits étant décrites, nous avons abordé par la suite sa conception détaillée en décortiquant à un niveau d'abstraction plus bas tous ses composants. Nous avons décrit en premier lieu le mode apprenant, ensuite le mode auteur et enfin le prototypage des deux modes relativement à des contraintes techniques.

Ce travail nous a révélé que la réalisation de systèmes auteurs pour l'EAD est une tâche extrêmement délicate. Il faut concevoir des modèles mettant en oeuvre le plus possible des connaissances généralisables (connaissances du domaine, connaissances pédagogiques et connaissances de diagnostic de l'apprenant) et d'assurer le couplage entre ces connaissances.

De plus, il faut offrir à l'auteur un langage de modélisation qui soit, d'une part, suffisamment complexe pour lui permettre de représenter son raisonnement aussi bien que possible, et d'autre part, suffisamment simple d'utilisation, car le problème est déjà suffisamment difficile pour ne pas ajouter de complexité liée à la formulation.

Notre travail se distingue des autres travaux de la communauté scientifique sur les systèmes auteurs par le fait que notre mode auteur est coopératif et que notre modèle de STI est basé sur la pédagogie par objectifs. Nous n'avons pas pris connaissance jusqu'à ce jour de travaux similaires. Tout d'abord, TalHits permet aux enseignants à suivre une certaine méthodologie pour construire leurs cours.

Laissant les préoccupations informatiques à côté, ils ne se préoccupent que de la modélisation pédagogique de leur cours en le découpant sur la base d'objectifs pédagogiques en trois niveaux hiérarchiques (parties, chapitres, uah) puis de concevoir les réseaux de prérequis d'objectifs et de concepts entre les différents éléments qui constituent le cours.

Cette fragmentation va les aider à coopérer entre eux pour introduire les différents objets pédagogiques via le mode auteur. Les apprenants, eux, ont la possibilité de négocier leur formation en choisissant les objectifs qu'ils désirent atteindre et peuvent également communiquer entre eux en vue d'une collaboration fructueuse. Le système est donc capable d'adapter le cours à chacun des apprenants en se basant sur les objectifs à atteindre et les connaissances antérieures de l'apprenant.

Perspectives d'amélioration du système

Toutefois, nous sommes conscients que notre système n'est pas un produit achevé au point de le "verrouiller" et le distribuer aux différents acteurs. Aussi, nous ne prétendons pas que notre travail est exempt de tout défaut ou d'être complet. Nous croyons que TalHits est passible de plusieurs améliorations. Nous indiquons ci-dessous une liste non exhaustive de ces améliorations que nous souhaitons les faire, d'abord au niveau de la réalisation actuelle (donc à très court terme), puis sur le plan conceptuel lui-même (donc à moyen et à long terme).

Comme nous l'avons déjà signalé auparavant, le prototype réalisé ne reflète pas entièrement les fonctionnalités décrites dans les chapitres 4, 5 et 6. Il faut donc compléter la réalisation en implantant les fonctionnalités manquantes. Ensuite, il faut effectuer une validation expérimentale plus poussée avec des enseignants et des apprenants de différentes spécialités. Ceci permettra de dégager d'éventuels problèmes relatifs à l'utilisation du système et nous conduit à mettre en

évidence d'autres améliorations souhaitables : corriger des incohérences et/ou optimiser le code en vue d'obtenir de meilleures performances (temps de réponse, place mémoire). Une validation du système en vue de supporter la langue arabe est aussi envisageable à court terme.

Sur le plan conceptuel, nous envisageons d'utiliser le langage XML. Ce langage devient de plus en plus le langage de référence du web remplaçant alors le langage Html. Cette transition permettra d'ajouter une définition sémantique aux documents hypermédias qui est pour le moment absente. Le Html possède des balises destinées essentiellement à la présentation des données que le développeur lui fournit et en aucun cas il ne tente de les interpréter.

Par contre, les balises du XML définissent plutôt la sémantique de données, c'est à dire que le balisage créé par le développeur donnera une signification précise des données fournies. Associé avec le langage XSL (eXtended Stylesheet Language), des cours adaptés à un niveau de granularité plus fin peuvent être obtenus. Notre uah pourra donc se décomposer en éléments plus fins du genre : définition, théorème, exemple, schéma, etc. Les remédiations aux anomalies des apprenants pourront alors être traités avec une grande finesse.

La prise en compte de ces améliorations nécessiterait alors l'enrichissement de la base de connaissances de notre système par d'autres nouvelles connaissances et peut induire au développement d'autres composants dans l'architecture logicielle. Nous pensons que le développement de ces perspectives permet d'ouvrir d'autres portes de recherche sur le système TalHits.



Bibliographie

- [Ackerman 95] Ackerman E., « Environnements interactifs : culture de zappeurs ou culture d'auteurs », Environnements Interactifs d'Apprentissage avec Ordinateur, 4ème Journées EIAO-ENS de Cachan, Tome 2, Editions Eyrolles, 1995.
- [Aiken & al 80] Aiken R.M., Hughes C.E., Moshell J.M., "An Introductory Computer Science Curriculum Based on an Affordable Computer System", Micro-Computers in Secondary Education, éd. E.D. Tagg, Amsterdam, North Holland, 1980.
- [Alpert & al 99] Alpert S-R, Singley M-K, Fair Weather P-G, "Deploying intelligent tutors on the web: an Architecture and an example", International Journal Of Artificial Intelligence in Education, Volume 10, n°2, pages 183-197, 1999. 1999.
- [Ariadne 04] <http://ariadne-eu.org>, site visité en 2004.
- [Arruarte & al 97] ARRUARTE A., FERNANDEZ-CASTRO I., FERRERO B., GREER J., « The IRIS shell: how to build ITSs from pedagogical and design requisites », International Journal of Artificial Intelligence in Education, volume 8, n° 3-4, pages 341-381, 1997.
- [Bair 85] Bair J.H., "The need for collaboration tools in offices", In proceedings of the AFIPS'85, Office Automation Conference, 1985.
- [Balacheff & al 98] Balacheff N., Bonneville F., Masseur N., Pavel P., Soury-Lavergne S., Vadcard L., « TéléCabri : principes d'une machine partenaire du formateur dans un contexte d'apprentissage distant », rapport d'activité scientifique 1995-1997, 1998, pages 174-175, 1998.
- [Balasubramanian 94] Balasubramanian V., "State of the Art Review on Hypermedia Issues And Applications", Graduate School of Management, Rutgers University, Newark, NJ, 1994.
- [Balpe 90] Balpe J-P, "Hyperdocuments, Hypertextes, Hypermédias", Edition Eyrolles, 1990.
- [Bannon & al 91] Bannon L., Robinson M., Schmidt K., Proceedings of the second European conference on computer-supported cooperative work (ECSCW'91), Kluwer Academic, Dordrecht, the Netherlands, 1991.
- [Baron 01] BARON M., « Intelligence Artificielle et EIAH », Ecole et Sciences Cognitives, février 2001.
- [Barth 93] Barth B., « Le savoir en construction », Edition Retz, Paris, 1993.
- [Behaz & al 03] BEHAZ A., DJOUDI M., ZIDANI A., « Approche de modélisation et d'adaptation des documents pédagogiques hypermédias en enseignement à distance », Actes du 6ième colloque CIDE'6, Caen, France, 2003.
- [Behaz & Djoudi 04] Behaz A., Djoudi M., « Modélisation et adaptation des documents pédagogiques hypermédias en enseignement à distance » 7ème Colloque Africain sur la Recherche en Informatique (CARI'04), Hammamet, Tunisie, 22-27 novembre, 2004.
- [Behaz & Djoudi 05] Behaz A., Djoudi M., « Génération dynamique de documents hypermédias adaptatifs dans un environnement numérique de travail », Revue ARIMA, Numéro spécial CARI'04, pages 25 à 53, 2005.

Bibliographie

- [Bestougeff & Fargette 82] Bestougeff H., Fargette J.P., « Enseignement et ordinateur », Edition Cedic, Paris, 1982.
- [Bloom 56] Bloom B. S., « Taxonomy of educational objectives : The classification of educational goals », Handbook I, cognitive domain. Longman, New York, 1956.
- [Bontemps & Martin 84] Bontemps G., Martin M., « Le rôle de l'enseignement assisté par ordinateur », dans Les Cahiers de Clio, 1984.
- [Bordeleau 99] Bordeleau P., « L'histoire des technologies informatiques et quelques-unes de leurs applications en éducation », 1999.
- [Boufaïda 95] Boufaïda M., « Méthodes et Outils pour la génération d'interfaces, la conduite du dialogue pédagogique et la mise en oeuvre du diagnostic », Thèse de Doctorat d'Etat de l'Université de Constantine, Constantine, 1995.
- [Bourda 01] Bourda Y., « Objets pédagogiques, vous avez dit objets pédagogiques », cahiers Gutenberg, n° 39-40, Mai, 2001.
- [Bourron & al, 95] Bourron Y., Chapuis J-P., Ruby J-L., « Pédagogie de l'audiovisuel et du multimédia », Editions Organisations, Paris, 1995.
- [Brown & al 82] Brown J-S., Burton R., Larkin K-M., "Diagnosing bugs in a simple procedural skill", in [Sleeman & Brown 82], pages 157-182, 1982.
- [Brown & Burton 82] Brown J-S., Burton R, "An investigation of computer coaching for informal learning activities", in [Sleeman & Brown82], pages 79-97, 1982.
- [[Brusilovsky 92] Brusilovsky P., "Intelligent Tutor, Environment and Manual for Introductory Programming", Educational and Training Technology International, 29(1), pages 26-34, 1992.
- [Brusilovsky & Pesin 94] Brusilovsky P., Pesin L., "An intelligent learning environment for CDS/ISIS users", In Levonen and Tukianinen (eds.), Proceedings. of The interdisciplinary workshop on complex learning in computer environments (CLCE'94), Joensuu, Finland, pages 29-33, 1994.
- [Brusilovsky 94a] Brusilovsky P., "Adaptative Hypermedia : An attempt to Analyze and Generalize", Multimedia, Hypermedia and Virtual Reality, First International Conference, MHVR'94, pages 288-304, 1994.
- [Brusilovsky & al 95] Brusilovsky P., Pesin L., "Visual annotation of links in adaptive hypermedia", CHI'95, Denver, USA, 1995.
- [Brusilovsky 96] Brusilovsky P., "Methods and techniques of adaptive hypermedia, User Modeling and User-Adapted Interaction", pages 87-129, 1996.
- [Brusilovsky & al 96] Brusilovsky, P., Schwarz, E., and Weber, G. "ELM-ART: An intelligent tutoring system on World Wide Web". In: C. Frasson, G. Gauthier and A. Lesgold (eds.) *Intelligent Tutoring Systems*. Lecture Notes in Computer Science, Vol. 1086, (Proceedings of Third International Conference on Intelligent Tutoring Systems, ITS-96, Montreal, June 12-14, 1996) Berlin: Springer Verlag, pp. 261-269, 1996.
- [Brusilovsky & al 96b] Brusilovsky P., Schwarz E., Weber G., "A tool for developing adaptive electronic textbooks on www", Proceedings of WebNet'96, AACE, 1996.
- [Brusilovsky 97] BRUSILOVSKY P., « Integrating hypermedia and intelligent tutoring technologies: from systems to authoring tools», in New media and telematic technologies for education, Twente University Press, Enschede, 1997.

Bibliographie

- [Brusilovsky & al 97a] Brusilovsky P., Nakabayashi K., Ritter S., "PAT-Online: A Model-Tracing tutor on the World-Wide Web", 1997. http://www.contrib.andrew.cmu.edu/~plb/AIED97_workshop/Ritter/
- [Brusilovsky & al 97b] Brusilovsky P., Ritter S., Schwarz E., "Distributed Intelligent tutoring on the web", Proceedings of the 8th World Conference of the AIED Society, Kobe, Japan, 18-22 August, 1997
- [Brusilovsky 98] Brusilovsky P., "Methods and techniques of Adaptive Hypermedia", Adaptive Hypertexte and Hypermedia, Kluwer Academic Publishers, pages 1-43, 1998.
- [Brusilovsky & al 98] Brusilovsky P., Eklund J., Schwarz E., "Web-based education for all: A tool for developing adaptive courseware", Computer Networks and ISDN Systems., pages 291-300, 1998.
- [Brusilovsky 99a] Brusilovsky P., "Adaptive Educational Systems on the World-Wide-Web: A Review of Available Technologies", 1999. <http://manic.cs.umass.edu/~stern/webits/itsworkshop/brusilovsky.html>
- [Brusilovsky 99b] Brusilovsky P., "Adaptive Hypermedia: From Intelligent Tutoring Systems to Web-Based Education", <http://manic.cs.umass.edu/~stern/webits/>, 1999.
- [Brusilovsky 01] Brusilovsky P., « Adaptive hypermedia », in user modeling and user adapted interaction, 2001.
- [Brusilovsky 03] Brusilovsky P., "Developing Adaptive Educational Hypermedia Systems: From Design Models to Authoring Tools", in Murray T., Blessing S. & Ainsworth S. (eds.), Authoring Tools for Advanced Technology Learning Environments, Kluwer Academic Publishers, 2003
- [Burton & Brown 78] Burton R., Brown J-S., "Diagnostic Models for procedural bugs in basic mathematical skills", Cognitive Science, Volume 2, pages 155-192, 1978.
- [Cadet 98] Cadet B., « Psychologie cognitive », IN PRESS Editions, Paris, 1998.
- [Carbonneaux & al 97] Carbonneaux Y., Laborde J-M., Madani J-M., « Une interface sur les graphes dans Cabri-Graphs », EIAO'97, Edition Hermes, pages 302-303, 1997.
- [Carbonnel 70] Carbonnel J., « AI in CAI: An artificial intelligent approach to Computer Assisted Instruction », IEEE Transactions on Man-Machine Systems, Volume 11, n°4, 1970.
- [Casse 90] CASSE P., « La formation performante », Edition Les Presses du Management, Paris, 1990.
- [Chaleat & al 05] Chaleat P., D. & Edwards Rouet J. R., « Php/MySQL et Javascript », Edition Eyrolles, 211 pages., 2005.
- [Charlot 98] Charlot J.M., « Formalisation et comparaison cognitives de modèles mentaux de novices et d'experts en situation de résolution de problèmes », Thèse de l'université de Sherbrooke, Quebec, Canada, 1998,
- [Charrier 01] Charrier J., « Développement de nouveaux modes d'accès au savoir : modes nouveaux de développement ? », Actes du 1^{er} colloque international sur l'université virtuelle, Université de la Formation Continue, pages 33-42, Alger, 28-30 mars, 2001.
- [Chatty 96] Chatty S., Girard P., Sire S., « Vers un support multimédia à la collaboration directe », Technique et Science Informatique, 15(9), 1996.
- [Clancey & Joerger 88] CLANCEY W., JOERGER K., « A practical authoring shell for apprenticeship learning », Proceedings of ITS-88, Montreal, pages 67-74, June 1988.

- [Clancey 82] CLANCEY W.J., "Tutoring rules for guiding a case method dialogue", in [Sleeman & Brown 82], pages 201-225, 1982.
- [Cockburn & Greenberg 93] COCKBURN A., GREENBERG S., « Making contact getting the group communicating with groupware », Proceedings of the ACM conference on organizational computing systems, pages 1-4, november 1993.
- [Collins 76] Collins A., "Processes in acquiring knowledge", Edition Lawrence Erlbaum Associates, pages 339-363, 1976.
- [De Bra & Calvi 98] De Bra P., Calvi L., « AHA : An open Adaptive Hypermedia Architecture », The New Review of Hypermedia and Multimedia 4, pages 115-139, 1998.
- [De La Passardière & al 01] De la Passardière B., GIROIRE H, « XML au service des applications pédagogiques », Revue Sciences et Techniques Educatives, Volume 8, n° 1, pages 99-112, Edition Hermes, 2001.
- [De la Passardière & Dufresne 92] De la Passardière B., Dufresne D., « Adaptive navigational tools for educational hypermedia ». In: Tomek (ed.), Computer Assisted Learning., Springer-Verlag, Berlin, pages 555-567, 1992.
- [De La Passardière 93] De La Passardière B., « Hypermédia et apprentissages », Actes des deuxièmes journées scientifiques, Institut National de Recherche Pédagogique, Paris, 1993.
- [De La Passardière, 85] B. De La Passardière, « De la conception à l'expérimentation d'outils logiciels pour l'initiation à la micro-informatique », Thèse Université Paris 6, 1985.
- [Decouchant & al 95] DECOUCHANT D., QUINT V., ROMERO M. , « Structured cooperative editing and group awareness », Proceedings of the Sixth International Conference on Human-Computer Interaction, volume. 20A, pages. 403-408, , Tokyo, July 9-14 1995.
- [Delestre 00] Delestre N., « Metadyne : Un hypermédia adaptatif dynamique pour l'enseignement », Thèse de Doctorat, Université de Rouen, France, 2000.
- [Denouet & Vignollet 00a] Denoué L., Vignollet L., « YAWAS un outil d'annotations pour les navigateurs du Web », Proceedings of RIAO, 2000.
- [Denouet & Vignollet 00b] Denoué L., Vignollet L., « An annotation tool for Web browsers and its applications to information retrieval », Proceedings of RIAO, 2000.
- [Deschenes 91] Deschênes A.-J., « Autonomie et enseignement à distance », Revue Canadienne pour l'éducation des adultes. V(1), pp 32-54, 1991.
- [Deschênes 93] Deschênes, A.-J., « La planification de l'apprentissage dans une activité de formation à distance ». Sainte-Foy, Québec : Télé-université, 1993.
- [Desmarais 00] DESMARAIS L., « La persévérance dans l'enseignement à distance : une étude de cas », Apprentissage des langues et systèmes d'information et de communication, vol 3, n° 1, 2000.
- [Després & George 01] DESPRES C., GEORGE S., « supporting learners activities in a distance learning environment », International Journal of Continuing Engineering Education and Lifelong Learning, Volume 11, n° 3, 2001.
- [Dessaint 95] Dessaint M.-P., « Au coeur de l'apprentissage: les objectifs et les activités », In Dessaint (Ed.), « La conception de cours: Guide de planification et de rédaction », (pp. 141-203), Sainte-Foy (Canada) : Presses de l'Université du Québec, 1995.
- [Djaghroui & al 04] Djaghroui O., Hachechna S., Djebbar N., « Conception et réalisation d'un tuteur intelligent sur le web », Mémoire d'ingénieur d'état encadré par Talhi Said, département d'informatique, université de Batna, 2004.

- [Dourish & Bellotti 92] Dourish P., Bellotti V., "Awareness and Coordination in Shared Workspaces", Proceedings of the ACM Conference on Computer-Supported Cooperative Work (CSCW-92), Toronto, Ontario, ACM Press, p. 107-114, 1992.
- [Ede & Lunsford 90] Ede L., Lunsford A., "Singular texts/plural authors: perspectives on collaborative writing", Southern Illinois University Press, Carbondale, 1990.
- [Ellis & al 91] Ellis C.A., Gibbs S.J., Rein G.L., "Groupware : Some Issues and Experiences", Communications of the ACM, Volume 34 n° 1, 1991.
- [Elsom-Cook & O'Malley 90] Elsom-Cook M.T., O'Malley C.E. "ECAL: Bridging the gap between CAL and Intelligent Tutoring Systems", Computers & Education, Volume 15 n° 1-3, 1990.
- [Englebart 62] Englebart D., "Augmenting human intellect: A conceptual framework, Summary Report Contract", SRI Project, Stanford Research Institute Menlo Park., 1962. <http://www.histech.rwth-aachen.de/www/quellen/engelbart/ahi62index.html>.
- [Faulhaber & Reinhardt 97] Faulhaber S., Reinhardt B., "D3-WWW-Trainer", Entwicklung einer Oberfläche für die Netzanwendung., 1997.
- [Fish & al 88] Fish R.S., Kraut R. E., Leland M.D., "Quilt: A Collaborative Tool for Cooperative Writing", In Proceedings of ACM International Conference on Office Information Systems, Volume 9(2-3), 1988.
- [Garzotto 94] Garzotto F., Mainetti L., Paolini P., "Adding Multimedia collections to the Dexter Model", Proceedings of ECHT94, ACM Press, Edinburgh, 1994.
- [George 01] George S., « SPLACH: système support de projet pour l'apprentissage collectif humain à distance », Thèse Université du Maine, 2001.
- [Gilbert & Han 99] Gilbert J-E., Han C-Y., "Arthur: Adapting Instruction to Accommodate Learning Style", In: Bra and Leggett (eds.) Proceedings. of WebNet'99, World Conference of the WWW and Internet, Honolulu, pages 433-438, 1999.
- [Giordan 98] Giordan A., « Apprendre ! », Belin, 1998.
- [Goffinet 02] Goffinet L., « Le multimédia et ses systèmes auteurs », Rapport interne, Institut d'Informatique Facultés Universitaires Notre-Dame de la Paix de Namur, 2002.
- [Gonschorek & Herzog 95] Gonschorek M., Herzog C., "Using hypertext for an adaptive helpsystem in an intelligent tutoring system". In J. Greer. (Ed.) Proceedings. of AIED'95, 7th World Conference on Artificial Intelligence in Education, Washington, pages 274-281, 1995.
- [Gouyet & al 95] Gouyet J. N., Najean T., Renard J., Enlart-Michel S., « Facteurs clés de succès des produits multimédia interactifs », rapport de l'Institut National de l'Audiovisuel, Étude Création Multimédia, 1995.
- [Greenberg & al 92] Greenberg S., Roseman K., Webster D., « Human and technical factors of distributed group drawing tools, interacting with computers », vol 4 n° 3, pages. 364-392, 1992.
- [Greif & al 93] Greif I., Seliger R., Weihl W., "A Case Study of CES: A Distributed Collaborative Editing System Implemented with Argus", IEEE Transactions on Software Engineering, Volume 18 N° 9, 1993.
- [Grudin 94] Grudin, J., "Computer-Supported Cooperative Work: Its History and Participation", IEEE Computer, 27, 5, 19-26, 1994.
- [Hameline 90] HAMELINE D., « Les objectifs pédagogiques en formation initiale et en formation continue », Edition ESF, 8ième édition, Paris, 1990.

Bibliographie

- [Herouz & al 02] HERROUZ A, DJOUDI M., « Conception d'un système d'assistance à la navigation et à l'apprentissage sur internet : SANA », Actes de la 7^{ème} MCSEAI, tome 1, Annaba, Algérie, 2002.
- [Holmes & Hawkes 94] Holmes D-J., Hawkes L-W., "Attribute Pattern-Matching Equations: a diagnostic technique for intelligent tutoring systems", Elsevier Science Publishers, New York, 1994.
- [Hoogeveen 95] Hoogeveen M., "Toward a New Multimedia Paradigm : is Multimedia Assisted Instruction Really Effective ?", ED-MEDIA'95 World Conference on Educational Multimedia and Hypermedia, 1995.
- [Hotte & Contamines 99] Hotte R., Contamines J., « From Leaders to intelligent learning collectives », workshop on educational Robotics, AI in Education, Le Mans, France, July 19-23, 1999.
- [Ims 04] « IMS Global Learning Consortium, Inc. », <http://www.imsproject.org/>, site visité en 2004
- [Janitza 85] Janitza J., « Enseignement assisté par ordinateur des langues étrangères », Paris, Hatier, 1985.
- [Jegade & al 95] Jegede O.J., Walkington J. et Naidu S., "An investigation into students' disposition to the use of objectives in distance learning materials", Educational Research, 37(3), pp 293-304, 1995.
- [Kay & Kummerfeld 94] Kay J. Kummerfeld R-J., "An individualised course for the C programming language", Proceedings of Second International WWW Conference, Chicago, IL, 17-20 October, 1994.
- [Khaldi 01] Khaldi E.H., « L'Université Virtuelle : défi du troisième millénaire », Actes du 1^{er} colloque international sur l'université virtuelle, Université de la Formation Continue, pages 21-26, Alger, 28-30 mars, 2001.
- [Kiyama & al 97] Kiyama M., Ishiuchi S., Ikede K., Tsujimoto M., Fukuhara Y., "Authoring methods for the web based Intelligent CAI System CALAT and its application to telecommunications service", Proceedings of AAI'97, Providence, RI, 1997.
- [Kraba & Ouazene 04] Kraba R., Ouazene N., « Conception et réalisation d'un éditeur coopératif pour l'enseignement à distance », Mémoire d'ingénieur d'état encadré par Talhi Said, département d'informatique, université de Batna, 2004.
- [Laborde 95] Laborde J-M., "Des connaissances abstraites aux réalités artificielles, le concept de micromonde: Cabri", Environnements Interactifs d'Apprentissage avec Ordinateur, Tome 2, D. Guin, J-F. Nicaud and Py (Eds), Eyrolles, Paris, 1995.
- [Lancy 00] Lancy S., « Infrastructure logicielle pour la gestion d'un environnement à objets répartis: application au télétravail coopératif », Thèse de Doctorat, Université de Rennes 1, 2000.
- [Landry 86] Landry F., « Revue de littérature sur le rôle des activités d'apprentissage dans l'enseignement à distance », Sainte-Foy, QC : Télé-Université, 1986.
- [Legendre 93] Legendre R., « Dictionnaire actuel de l'éducation », (2^{ème} édition), Montréal : Guérin Éditeur, 1993.
- [Lelouche 87] Lelouche R., « Apports de l'EIAO à l'EAO », Actes du Congrès Francophone sur l'Enseignement Assisté. par Ordinateur, pages 173-181, 1987.
- [Leman & al 96] Leman S., Marcenac P., Giroux S., « Un modèle multi-agents de l'apprenant », Sciences et techniques éducatives, Volume 3, n°4, pages 464-483, 1996.

- [Lesgold 88] Lesgold A., « Toward a theory of curriculum for use in designing intelligent instructional systems », MAND Ed, pp. 114-137, 1988.
- [Lewis & Hodges 88] Lewis B.T., Hodges J., "Shared books", Collaborative Publication Management for an Office Information System", In Proceedings of ACM International Conference on Office Information Systems, volume 9(2-3), 1988.
- [Lima 92] Lima J.C.M., « SINTONIA: vers un système intelligent pour la construction des diagnostics dans les tuteurs d'informatique », Thèse de doctorat, Université Paris VI, Paris, 1992.
- [LOM 01] «Draft for Learning Object Metadata (LOM 6.1) », 2001. <http://ltsc.ieee.org/wg12/>
- [Louanchi 87] Louanchi D., « Eléments de pédagogie », Edition OPU, Alger, 1987.
- [Ltsc 04] « Draft for Learning Object Metadata (LOM 6.1) », LTSC-IEEE LOM Working Group, <http://ltsc.ieee.org/wg12/>, site visité en 2004.
- [Luengo 97] Luengo V., « Un micro-monde de preuve int.grant la réfutation : Cabri-Euclide », EIAO'97, Edition Hermes, pages 85-97, 1997.
- [Madaule & al 87] Madaule F., Barril P., De La Passardiere B, Lecalvez F., Poc M., et. Urtasun M., « Systèmes d'enseignement assisté par ordinateur : étude comparative », Technique et Science Informatiques, Vol. 6 n°1, Eds. AFCET Bordas, France, 1987.
- [Mager 75] Mager R.F., « Preparing Instructional Objectives », (2nd edition.), Belmont, California : Pitman Learning inc, 1975.
- [Marcenac 92] Marcenac P., « EDDI : un prototype de système auteur pour des didacticiels intelligents », Génie éducatif, No 4&5, 1992.
- [Martens 93] Martens R.L. et Valcke M.M.A., « Validation of a theory about the functions and effects of embedded support devices in written study materials », In F. Lockwood et M. Valcke (Eds.), New developments in research in relation to distance education materials, pp. 65-84, 1993.
- [Martin 02] Martin B., « Introduction au world wide web », techniques de l'ingénieur, volume H 2 908, traité informatique, 2002.
- [Maurice 00] Maurice P., « Langages de script », techniques de l'ingénieur, H 3 118, traité informatique, 2000.
- [Mendelsohn 94] Mendelsohn P, « Peut-on vraiment opposer "savoirs et savoir-faire" quand on parle d'apprentissage ? », Entretiens Nathan, Sorbonne, manuscrit, 1994.
- [Michard 99] Michard A., « XML: Langage et Applications », Edition Eyrolles, 1999.
- [Miles & al 93] Miles V.C., McCarthy J.C., Dix A.J., Harrison M.D., Monk A.F., "Computer supported cooperative writing", chapter 8 Re-viewing Designs for a Synchronous-Asynchronous group editing environment, Springer-Verlag, 1993.
- [Miller 56] Miller G. A., "The magical number seven plus or minus two: some limits on our capacity for processing information", Psychological Review, pages 81-97, 1956
- [Misanchuk 94] Misanchuk E.R., "Print tools for Distance Education", In B. Willis (Ed.), "Distance education : strategies and tools", pp. 109-133, Englewoods Cliffs, New-Jersey : Educational Technology Publications inc, 1994.
- [Mitchell 96] Mitchell A, "Communication and shared understanding in collaborative writing". Master of science thesis, University of Toronto, 1996.

Bibliographie

- [Muhammad & al 05] Muhammad A., Martínez A.M. et Decouchant D., "Awareness and Coordination for Web Cooperative Authoring". In Proc. of AWIC'2005, The 3rd International Atlantic Web Intelligence Conference, Lecture Notes in Artificial Intelligence, no 3528, Springer Verlag, Lodz, Poland., 327-333, 2005.
- [Muller 06] Muller F., <http://francois.muller.free.fr>, site visité en 2006.
- [Murray 98] MURRAY T., « Authoring knowledge-based tutors : tools for content, instructional strategy, student model and interface design », *Journal of the Learning Sciences*, volume 7, n° 1, 1998.
- [Murray 99] Murray T. « Authoring intelligent tutoring systems : an analysis of the state of the art », *International Journal of AI in Education*, vol. 10, p. 98-129, 1999.
- [Nakabayashi & al 97] Nakabayashi K., Maruyama, M., Kato, Y., Touhei, H., and Fukuhara, Y. (1997) "Architecture of an intelligent tutoring system on the WWW". In: B. d. Boulay and R. Mizoguchi (eds.) *Artificial Intelligence in Education: Knowledge and Media in Learning Systems*. (Proceedings of AI-ED'97, World Conference on Artificial Intelligence in Education, Kobe, Japan, 18-22 August 1997) Amsterdam: IOS, pp. 39-46.
- [Nanard & Nanard 99] Nanard M., Nanard J., « Documents hypermédias : définitions et concepts associés », *techniques de l'ingénieur*, H 7 208, traité Informatique, 1999.
- [Nanard 94] Nanard M., « Hypermédias, éducation et formation : contribution à la structuration d'un champ émergent », *Séminaire Hypermédias Education et formation*, Université Pierre et Mari Curie (Paris VI), GTIEF, 1994.
- [Nanard 95] Nanard M., « Les hypertextes : au delà. des liens, la connaissance », *Sciences et techniques éducatives*, Volume 2, n°1, Edition Hermes, 1995.
- [Nanard 96] Nanard M., « Le multimédia, de l'âge artisanal à l'âge industriel », LIRMM, MONTPELLIER, 1996.
- [Nardi 96] Nardi A., « Context and consciousness, activity theory and human computer interaction », MIT Press, Cambridge, Massachusetts, 400 pages, 1996.
- [Nelson 65] Nelson T., "A File Structure for the Complex, The Changing and The Indeterminate", ACM 20th National Conference, 1965.
- [Neuwirth & al 90] Neuwirth C.M., Kaufer D.S., Chandhok R., Morris J., "Issues in the design of computer support for co-authoring and commenting", *Proceedings of International Conference on CSCW (CSCW'90)*, 1990.
- [Newman-Wolf & al 91] Newman-Wolfe R.E., Ramirez C.L., Pelimuhandiran H., Montes M., Webb M., Wilson D.L., « A Brief Overview of the DCS Distributed Conferencing System », In *Proceedings of the Usenix conference*, 1991.
- [Nicaud & Vivet 88] Nicaud J.F., Vivet M., « Les tuteurs intelligents: réalisations et tendances de recherches », *Journal Techniques et Sciences Informatiques*, Volume.7 n°1, pages. 21-45, Edition Hermes, 1988.
- [Nkambou & al 03] Nkambou R., Frasson C., Gauthier G.. "CREAM-Tools : An Authoring Environment for Knowledge Engineering in Intelligent Tutoring Systems". In : Murray, T., Blessing, S. and Ainsworth, S. (Eds) : *Authoring Tools for Advanced Technology Learning Environments : Toward cost-effective adaptative, interactive, and intelligent educational software*. Pp. 93-138. Kluwer Publishers, 2003.
- [Okazaki & al 97] Okazaki Y., Watanabe K., Kondo H., "An Implementation of the WWW Based STI for Guiding Differential Calculations", 1997. http://www.contrib.andrew.cmu.edu/~plb/AIED97_workshop/Okazaki/
-

Bibliographie

- [Oravep 00] ORAVEP, « Etude comparative technique et pédagogique des plate-formes pour la formation à distance », www.oravep.asso.fr, 2000.
- [Orfali & al 97] ORFALI R., HARKEY D., EDWARDS J., « Essential client/server survival guide », John Willeys & Sons Inc Eds, 2nd edition , New York, 1997.
- [Ouchen & al 02] Ouchen N., Boumaraf K., TALHI S., DJOUDI M., « Navigateur web avec annotation et marquage pour l'apprentissage à distance », Actes de la conférence internationale sur la langue arabe et les NTIC, HCLA, Alger, Algérie, 2002.
- [Pacull & al 94] PACULL F., SANDOZ A., SCHIPER A. , « Duplex : a distributed collaborative editing environment in large scale », Proceedings of the ACM Conference CSCW'94, ACM Press, 1994.
- [Paiva & al 95] Paiva A., Self J., Hartley R., "Externalising Learner models", Proceedings of World Conference on artificial Intelligence in Education, pages 509-516, Washington, August 16-19, 1995.
- [Palies 84] PALIES O., « Le projet ELECTRE: un exemple d'E.I.A.O dont l'objectif tutoriel passe par la construction automatique d'un modèle du sujet», Publication du GR 22, France, 1984.
- [Papert 80] Papert S., "Mindstorms, Children, computers and powerful ideas", Flammarion, 1980.
- [Paquette & Tchounikine 02] Paquette G., Tchounikine P., « Contribution à l'ingénierie des systèmes conseillers : une approche méthodologique fondée sur l'analyse du modèle de la tâche », Sciences et Techniques Educatives, Volume 9, n°1, 2002.
- [Peachey & McCalla 86] Peachey D.R., G. McCalla, "Using planning techniques in intelligent tutoring systems", International Journal of Man-Machines, Volume.24, n° 1, pages 77-98, 1986.
- [Pendergast & Vogel 90] Pendergast M. O., Vogel D., "Design and Implementation of a PC/LAN Based Multi-User Text Editor, In Proceedings of IFIP WG Conference on Multi-User Interfaces and Applications, North Holland, 1990.
- [Perriault 96] PERRIAULT J., « La communication du savoir à distance», l'Harmattan, Paris, 1996.
- [Poitiers 06] www.univ-poitiers.fr/michinov/cscl.html, site visité en 2006.
- [Posner & Baeker 92] Posner I.R., Baeker R.M., "How people write together", In Proceedings of the 25th Hawaii International Conference on System Sciences, volume IV, 1992.
- [Preau 00] PREAU, « Etude sur la téléformation », www.preau.asso.fr, 2000.
- [Pujole 97] Pujole G., « Architecture TCP/IP », techniques de l'ingénieur, H 2 288, traité Informatique, 1997.
- [Puybaret 03] Puybaret E., « Java », Edition Eyrolles, 272 pages,, 2003.
- [Reynard & al 98] REYNARD G., BENFORD S., GREENHALGH C., HEATH C., « Awareness driven video quality of service in collaborative virtual environments », Proceedings of the ACM conference CHI'98, 1998.
- [Rhe.ume 93] Rhe.ume J., « Les hypertextes et les hyperm. dia », Revue EducaTechnologie, Volume 1, n° 2, décembre, 1993..
- [Rigault -Ricardi 82] Rigault-Riccardi C., « Perception du rôle de l'enseignant à distance », Formation à distance perspectives et prospectives, pages 255-260, 1982.

Bibliographie

- [Roschelle & Teasley 95] Roschelle J., Teasley S., « The construction of shared knowledge in collaborative problem solving », dans O'malley, c.e. edition, Computer Supported Collaborative Learning., Springer-Verlag, Heidelberg, 1995.
- [Rowntree 94] Rowntree D., "Preparing materials for open, distance and flexible learning", (2ième édition.). Londres : Kogan Page, 1994.
- [Saint-Onge 92] Saint-Onge M., « Les objectifs pédagogiques: pour ou contre ? », Pédagogie collégiale. Vol 6 (2), pp 23-28, 1992.
- [Schierman & al 92] Schierman E., Teare S., McLaren J., "Towards a course development model for graduate level distance education", Journal of Distance Education, Vol VII(2), pp 51-65, 1992.
- [Sinia 02] SINIA R., "an overview of CSCL (Computer Supported Collaborative Learning), which by many is regarded as an emerging paradigm of educational IT use (computers in education)", Rapport Technique, 2002.
- [Sleeman & Brown 82] Sleeman D., Brown J.S., "Intelligent Tutoring Systems", Academic Press, London, 1982.
- [Smythe 02] SMYTHE C. « IMSQuestion & Test Interoperability : an overview », Final specification version 1.2, IMS, 2002.
- [Sommerville 88] Sommerville I., (Traduction française de B. DION) « Le génie logiciel et ses applications », InterEditions, Paris, 1988.
- [Stern & al 97] Stern, M., Woolf, B. P., Kuroso, J., "Intelligence on the Web ?", In: B. d. Boulay and R. Mizoguchi (eds.) *Artificial Intelligence in Education: Knowledge and Media in Learning Systems*. (Proceedings of AI-ED'97, 8th World Conference on Artificial Intelligence in Education, Kobe, Japan, 18-22 August 1997) Amsterdam: IOS, pp. 490-497, 1997.
- [Stevens & Collins 82] Stevens A., Collins A., "Misconceptions in students' understanding", in [Sleeman & Brown 82], pages13-50, 1982.
- [Tagliante 91] TAGLIANTE C., « L'évaluation », Édition Clé international , France, 1991.
- [Talhi 96] Talhi S. « Moalim: un système auteur de l'EIAO », Actes du 18ième Symposium DECUS FRANCE, Paris, 2-4 avril, 1996.
- [Talhi & al 96] Talhi S., Boufaïda M., Boufaïda Z., « Moalim: vers un système auteur pour l'EIAO », Actes du séminaire national en informatique (SNITO'96), pages.29-51, Université de Tizi Ouzou, Algérie, 11-13 novembre 1996.
- [Talhi & al 01] Talhi S., Djoudi M., Zidani A., « Un système auteur de tuteurs intelligents : évolution du mono-usager vers la coopération », Revue Sciences et Techniques Educatives, Volume 8, n° 1, pages 127-138, Edition Hermes, 2001.
- [Talhi & al 02] TALHI S, DJOUDI M., ZIDANI A., « Camits : système auteur coopératif de tuteurs intelligents », Actes du 5ième colloque CIDE'5, Hammamet, Tunisie, 2002.
- [Talhi & al 05] Talhi S., Djoudi M., Zidat S., Batouche M., « Un système tuteur intelligent hypermédia pour l'apprentissage à distance asynchrone », congrès international en informatique appliquée, CIIA'05, Bordj Bou Arréridj, Algérie, 19-21 Novembre, 2005.
- [Talhi & al 06] Talhi S., Djoudi M., Batouche M., « A cooperative authoring for intelligent tutoring system », Asian Journal of Information Technology (AJIT), ISSN: 1682-3915, Volume.5, n° 5, pp 497-503, 2006.

Bibliographie

- [Tardif 92] Tardiff J., « Pour un enseignement stratégique, l'apport de la psychologie cognitive », Éditions Logiques, Montréal, 1992.
- [Tasso & Ermacore 04] Tasso A., Ermacore S., « Initiation à JSP », Edition Eyrolles, 353pages., 2004.
- [Tenachi 94] Tenachi. A, « Modélisation de connaissances pédagogiques - systèmes d'aide à l'élaboration de cursus personnalisés », Thèse de Doctorat de l'Université Henri Poincaré, Nancy, 1994.
- [Thibaut & Verrier 83] J Thibaut M., Verrier G., « Vous avez dit PIGE ? », Les langues modernes, 1983.
- [Torcq & Flaschner 83] Torcq M., Flaschner T., « Langages de programmation, langages auteurs », dans les langues modernes, 1983.
- [Uhr 69] Uhr L, "Teaching machine programs that generate problems as function of interaction with students", Proceedings of the 24th national conference on CAI, pp. 125-134, 1969.
- [Uquebec 06] www.puk.quebec.ca, site visité en 2006.
- [Url 0] Authorware:<http://www.macromedia.com/software/authorware/>
Toolbook:<http://www.asymetrix.com/products/toolbook2/>
HyperCard :<http://hypercard.apple.com/>
Quest : <http://www.allencomm.com/software/quest/>
IconAuthor : <http://www.aimtech.com:80/products/iconauthor/>
Director : <http://www.macromedia.com/software/director/>
- [Url 1] <http://www.macromedia.com/shockwave/>
- [Url 10] <http://www.imsproject.org>
- [Url 11] <http://www.imsproject.org/specifications.html>
- [Url 12] <http://ariadne.unil.ch/main.htm>
- [Url 2] <http://www.asymetrix.com/products/toolbook2/neuron/>
- [Url 3] <http://www.allegiant.com/roadster/default.html>
- [Url 4] CBT Report, publié par CBT Solutions Magazine. http://www.cbtsolutions.com/html/r_guest1.htm, 1997.
- [Url 5] <http://www.allencomm.com/software/designer/>
- [Url 6] <http://www.asymetrix.com/products/librarian/>
- [Url 7] <http://www.qmark.com/perception/>
- [Url 8] <http://www.qmark.com/qml/index.html>
- [Url 9] <http://www.educom.edu>
- [VanLehn & Martin 97] VanLehn K., Martin J., "Evaluation of an assesement system based on bayesian student modeling", International Journal of Artificial Intelligence in Education, Volume 8, n°1, 1997, pages 179-221, 1997.
- [Vanneste 94] Vanneste P., "The use of reverse engineering in novice program analysis", (PhD Thesis),Katholieke Universiteit Leuven, 1994.
- [Varela 96] Varela F. J., « Invitation aux sciences cognitives », Seuil, 1996.

Bibliographie

- [Vassileva 97] Vassileva J., "Dynamic Courseware Generation on the WWW", Proceedings of the workshop : Adaptive Systems and User Modeling on the World Wide Web, Sixth International Conference on User Modeling, 1997.
- [Vergnaud 91] Vergnaud G., « La théorie des champs conceptuels », Recherches en didactique, 1991.
- [Viré 86] Viré G.H., « L'acquisition des automatismes et l'enseignement assisté par ordinateur », dans L'enseignement des langues anciennes aux grands débutants (problèmes, méthodes, finalités), édition C.Aziza, M. Dubuisson et E. Famerie, Liège, Université de Liège et de la Sorbonne Nouvelle, 1986.
- [Vivet 87] Vivet M., « Systèmes experts pour enseigner métaconnaissances et explications », Congrès International MARI/COGNITIVA'87, Paris, 18-22 mai, 1987.
- [Vygotski 85] Vygotski L. S., « Pensée et langage. Messidor », Éditions sociales, 1985.
- [Warendorf & Tan 97] K. Warendorf, C. Tan, "ADIS - An animated data structure intelligent tutoring system or Putting an interactive tutor on the www", In Brusilovsky, Nakabayashi and Ritter (eds.) Proceedings of Workshop "Intelligent Educational Systems on the World Wide Web" at AI-ED'97, 8th World Conference on Artificial Intelligence in Education, Kobe, Japan, pages 54-60, 18 August 1997.
- [Weber & Mollenberg 95] Weber G., Möllenberg A., "ELM-Programming-Environment : A Tutoring System for LISP Beginners". In: Wender, Schmalhofer and Böcker (eds.): Cognition and Computer Programming. Norwood, NJ: Ablex, pages 373-408, 1995.
- [Wenger 87] Wenger, « Artificial Intelligence and Tutoring Systems », Addison-Wesley Editions., USA, 1987.
- [Wu Yao Kuang 00] Wu Yao Kuang R., « La robustesse des systèmes auteurs multimédias : contribution théorique et mise en œuvre », Thèse de doctorat de l'Université Paris 8, 2000.
- [Zeiliger & al 97] Zeiliger R., Reggers T., Baldewyns L., Jans V., « Facilitating Web Navigation : Integrated tools for Active and Cooperative Learners », Proceedings of the 5th International Conference on Computers in Education, ICCE'97, Kuching, Sarawok, Malaysia, 1997.
- [Zidani & al 00] ZIDANI A., BOUFAIDA M., DJOUDI M., « JamEdit : un outil interactif et coopératif pour l'édition de documents », Technique et Science Informatiques, Vol. 19, n°1, 2000, p. 1-23, Hermes, Paris, 2000.
- [Zidani & al 02] Zidani A, Djoudi M, Zidat S., Talhi S., « un environnement coopératif pour l'apprentissage sur internet », Actes du 6ième colloque CARI'02, Yaoundé, Cameroun, 2002.
- [Zubot 93] Zubot M., « Writing your course », Saskatoon : University Extension Press, Univ. of Saskatchewan, 1993.



Annexes

Annexe 1

Nos Publications

1. Publications dans une revue internationale (avec comité de lecture) : 3

1) **Said Talhi**, Mahieddine Djoudi, Mohamed Batouche, "Authoring Grouware for Intelligent tutoring systems", Information Technology Journal (ITJ), ISSN : 1812-5638, volume 5, n° 5, pp 860-867, 2006.

2) **Said Talhi**, Mahieddine Djoudi, Mohamed Batouche, "A Cooperative Authoring for Intelligent Tutoring Systems", Asian Journal of Information Technology (AJIT), ISSN: 1682-3915, volume 5, n° 5, pp 497-503, 2006.

3) **Said Talhi**, Mahieddine Djoudi et Abdelmadjid Zidani, « Un système auteur de tuteurs intelligents : évolution du mono-usager vers la coopération », Revue Techniques et Sciences Éducatives (TSE), ISSN : 1265-1338, volume 8, n° 1-2, pp 127-138, 2001.

2. Publications dans des congrès internationaux (avec comité de lecture) : 16

1) **Said Talhi**, Mahieddine Djoudi, Samir Zidat, Mohammed Batouche, « Un système tuteur intelligent hypermédia pour l'apprentissage à distance asynchrone », Congrès international en informatique appliquée, CIIA'05, Bordj Bou Arréridj, Algérie, 19-21 novembre, 2005.

2) Zidat Samir, Djoudi Mahieddine, Zidani Abdelmadjid, **Talhi Said**, « Appréciation de la participation des apprenants en situation de résolution collaborative d'exercices », Congrès international en informatique appliquée, CIIA'05, Bordj Bou Arréridj, Algérie, 19-21 novembre, 2005.

3) **Said Talhi**, Mahieddine Djoudi, Mohamed Batouche, « Réalisation coopérative de tuteurs intelligents hypermédiés », Colloque euro méditerranéen pour l'approfondissement de la formation à distance, CEMAFORAD, Béjaia, Algérie, 12-14 novembre 2005, <http://elearning.freezee.org/cemaforad2/>

4) Samir Zidat, Mahieddine Djoudi, **Said Talhi**, Abdelmadjid Zidani, « Démarche pour l'évaluation de plate-formes d'enseignement à distance », Colloque euoro-méditerranéen pour

l'approfondissement de la formation à distance, CEMAFORAD, Béjaia, Algérie, 12-14 novembre, 2005, <http://elearning.freezee.org/semaforad2/>

5) **Said Talhi**; Mahieddine Djoudi, "A Cooperative Design of Hymermedia Intelligent Tutoring Systems", IEEE International Computer Systems & Information Technology Conference, ICSIT05, July 19-21, Algiers, Algeria, 2005.

6) Zidat Samir, Djoudi Mahieddine, Zidani Abdelmadjid et **Talhi Said**, « Résolution collaborative d'exercices en situation d'apprentissage à distance », Conférence Internationale: Sciences Electroniques, Technologies de l'Information et des Télécommunications, IEEE SETIT, ISBN 9973-41-902-2, actes sur CDROM, Sousse, Tunisie 27-31 mars, 2005.

7) **Said Talhi**, Mahieddine Djoudi, Mohammed C. Batouche, Salima Ouadfel, Samir Zidat, Abdelmadjid Zidani, « Formation A Distance Asynchrone Sur Le Web : Un Système Tuteur Intelligent Basé Sur L'Hypermédia », Conférence Internationale: Sciences Electroniques, Technologies de l'Information et des Télécommunications, IEEE SETIT, ISBN 9973-41-902-2, actes sur CDROM, Sousse, Tunisie, 27-31 mars, 2005.

8) Samir Zidat, Mahieddine Djoudi, Abdelmadjid Zidani, **Said Talhi**, "Collaborative Resolution of Exercises in Situation of Distance Learning", 2004 International Arab Conference on Information Technology (ACIT'2004) Constantine, Algeria, December 12-15, 2004.

9) Nawel Ouchen, Karima Boumaraf, **Said Talhi**, Mahieddine Djoudi, « Navigateur Web pour l'enseignement à distance », Congrès International Annuel sur l'arabisation des sciences, Le Caire, Egypte, 12/13 mars, 2003.

10) Samia Bessouf, Belkout Louiza, **Said Talhi**, Mahieddine Djoudi, « Un système intelligent pour l'enseignement à distance », Congrès International Annuel sur l'arabisation des sciences, Le Caire, Egypte, 12/13 mars, 2003.

11) Nawel Ouchen, Karima Boumaraf, **Said Talhi**, Mahieddine Djoudi, « Wen-Vecole : navigateur web avec annotations et marquage pour l'apprentissage à distance », Conférence Internationale sur la langue arabe et les technologies de l'information, Alger, 28-29 décembre 2002, pp 68-75, 2002.

12) **Said Talhi**, Mahieddine. Djoudi et Abdelmadjid Zidani, « Camits : Système Auteur Coopératif de Tuteurs Intelligents», 5ième Colloque International sur le Document Électronique (CIDE 2002), ISBN : 2-7261-1223-4, pp 107-121, Hammamet, Tunisie, 20-23 octobre, 2002.

13) Abdelmadjid Zidani, Mahieddine Djoudi, Samir Zidat, **Said Talhi**, « CHELIA : Un environnement coopératif pour le l'apprentissage sur Internet », 6ième Colloque CARI'2002 de l'INRIA, Yaoundé, Cameroun, 14-17 octobre 2002, ISBN : 2-7261-1214-5, pp 339-346, 2002.

14) Samir Zidat, Abdelmadjid Zidani, Mahieddine Djoudi, **Said Talhi**, « CHELIA : Une Plate-Forme Coopérative pour l'Apprentissage sur Internet », 2ième Colloque international sur l'université virtuelle, Alger, 15-17 mai, 2002.

15) **Said Talhi**, Mahieddine. Djoudi et Abdelmadjid Zidani, « Un système auteur de tuteurs intelligents : évolution du mono-usager vers la coopération », Conférence EIAO'01, 24-27 avril 2001, ISBN : 2-7462-0254-9, pp 127-138, Paris, 2001.

16) **Said Talhi**, Mahieddine Djoudi et Abdelmadjid Zidani, « MOALIM-CO : Un système auteur coopératif pour le développement de systèmes tuteurs intelligents », 5ième Colloque CARI'2000 de l'INRIA, Antananarivo, Madagascar, 16-19 octobre 2000, ISBN : 2-7261-1167-X, pp 281-288, 2000.

3. Publications dans des congrès internationaux (sans comité de lecture) : 2

1) Mahieddine Djoudi, Abdelmadjid Zidani, Samir Zidat, **Said Talhi** « Edition coopérative de cours structurés dans l'environnement CHELIA », Séminaire international Méthodologie de conception du cours écrit en EAD, UFC, Alger, 09-11 janvier, 2002

2) Mahieddine Djoudi, Abdelmadjid Zidani, **Said Talhi**, "SIC'ED: A Collaborative Intelligent System for Distance Learning", CHI2001 Workshop "Universal design: Towards universal access in the Information Society", Seattle, USA, 31 mars -5 avril, 2001.

4. Communications nationales avec comité de lecture : 2

1) **Said Talhi**, Mahieddine Djoudi, Mohamed Batouche, "Assistance à l'apprenant à distance par tuteur intelligent hypermédia", *Journées d'étude sur les TIC, JeTIC2006, Bechar Algérie*, 15-16 avril, 2006.

2) Nawel Ouchen, Karima Boumaraf, **Said Talhi**, Mahieddine Djoudi, « Navigateur web avec annotation et marquage pour l'apprentissage à distance », Journées d'Etudes Nationales sur l'Internet et la Didactique des Sciences, ENSET d'Oran, 28-29 novembre, 2004.



Annexe 2

Exemple de structure de la matière pour le module « systèmes experts »

Fragment	Intitulé	Nature
Partie (1)	Introduction générale à l'intelligence artificielle et la représentation des connaissances	
Résumé-partie (1)		
Chapitre (1,1)	Introduction à l'intelligence artificielle	
Résumé-chapitre (1,1)		
UAH (1,1,1)	Définitions et exemples	Non évaluable
UAH (1,1,2)	Repères historiques	Non évaluable
UAH (1,1,3)	Le cerveau et l'ordinateur	Évaluable
UAH (1,1,4)	L'IA et l'informatique	Non évaluable
UAH (1,1,5)	Les domaines d'application	Non évaluable
Chapitre (1,2)	Généralités sur la représentation des connaissances	
Résumé-chapitre (1,2)		
UAH (1,2,1)	Les types et les catégories des connaissances	Évaluable
UAH (1,2,2)	Les éléments de base de la représentation	Non évaluable
UAH (1,2,3)	Les représentations les plus utilisées	Évaluable
UAH (1,2,4)	Comment choisir une bonne représentation ?	Non évaluable
Chapitre (1,3)	Les représentations logiques	
Résumé-chapitre (1,3)		
UAH (1,3,1)	Calcul des propositions	Évaluable
UAH (1,3,2)	Calcul des prédicats	Évaluable
UAH (1,3,3)	Principe de résolution	Évaluable
UAH (1,3,4)	Programmation logique	Évaluable
Partie (2)	Les systèmes experts et les systèmes à règles de production	
Résumé-partie (2)		
Chapitre (2,1)	Les systèmes experts	
Résumé-chapitre (2,1)		
UAH (2,1,1)	Définition et architecture d'un système expert	Évaluable
UAH (2,1,2)	Typologie des systèmes experts	Non évaluable
UAH (2,1,3)	Les domaines d'application	Non évaluable
UAH (2,1,4)	Construction d'un système expert	Évaluable
UAH (2,1,5)	Exemple de dialogue avec un système expert	Non évaluable
Chapitre (2,2)	Les systèmes à règles de production	
Résumé-chapitre (2,2)		
UAH (2,2,1)	Les règles de production et la logique	Non évaluable
UAH (2,2,2)	Cycle de base d un moteur d inférence	Évaluable
UAH (2,2,3)	Mode de raisonnement : chaînage avant	Évaluable

Annexes

UAH (2,2,4)	Mode de raisonnement : chaînage arrière	Evaluable
UAH (2,2,5)	Les métaconnaissances	Non évaluable
UAH (2,2,6)	Avantages et inconvénients des SRP	Evaluable
Partie (3)	Le raisonnement approximatif et l'acquisition des connaissances	
Résumé-partie (3)		
Chapitre (3,1)	Le raisonnement approximatif	
Résumé-chapitre (3,1)		
UAH (3,1,1)	Probabilité et raisonnement BAYESIEN	Evaluable
UAH (3,1,2)	Raisonnement plausible et crédible	Non évaluable
UAH (3,1,3)	Les mesures de nécessité et de possibilité	Non évaluable
UAH (3,1,4)	Les ensembles flous	Non évaluable
UAH (3,1,5)	Le raisonnement dans le système MYCIN	Non évaluable
Chapitre (3,2)	L'acquisition des connaissances	
Résumé-chapitre (3,2)		
UAH (3,2,1)	Difficulté d'acquisition	Non évaluable
UAH (3,2,2)	Méthodes cognitiennes	Non évaluable
UAH (3,2,3)	Méthodes informatisées	Evaluable
Partie (4)	Etude de cas	
Résumé-partie (4)		
Chapitre (4,1)	Réalisation d'un micro système expert	
Résumé-chapitre (4,1)		
UAH (4,1,1)	Edition de la base de règles	Non évaluable
UAH (4,1,1)	Chargement de la base de règles	Non évaluable
UAH (4,1,1)	Chaînage avant	Non évaluable



Annexe 3

Exemple de base de règles instances générée pour le module « systèmes experts »

Base de règles de recherche des uah		
NR	Condition	Conclusion
1	presenter_resume_par(1)	resume_par_presente(1)
2	presenter_resume_par(1)	afficher_resume_par(1)
3	presenter_resume_par(2)	resume_par_presente(2)
4	presenter_resume_par(2)	afficher_resume_par(2)
5	presenter_resume_par(3)	resume_par_presente(3)
6	presenter_resume_par(3)	afficher_resume_par(3)
7	presenter_resume_par(4)	resume_par_presente(4)
8	presenter_resume_par(4)	afficher_resume_par(4)
9	presenter_resume_chap(1,1)	resume_chap_presente(1,1)
10	presenter_resume_chap(1,1)	afficher_resume_chap(1,1)
11	presenter_resume_chap(1,2)	resume_chap_presente(1,2)
12	presenter_resume_chap(1,2)	afficher_resume_chap(1,2)
13	presenter_resume_chap(1,3)	resume_chap_presente(1,3)
14	presenter_resume_chap(1,3)	afficher_resume_chap(1,3)
15	presenter_resume_chap(2,1)	resume_chap_presente(2,1)
16	presenter_resume_chap(2,1)	afficher_resume_chap(2,1)
17	presenter_resume_chap(2,2)	resume_chap_presente(2,2)
18	presenter_resume_chap(2,2)	afficher_resume_chap(2,2)
19	presenter_resume_chap(3,1)	resume_chap_presente(3,1)
20	presenter_resume_chap(3,1)	afficher_resume_chap(3,1)
21	presenter_resume_chap(3,2)	resume_chap_presente(3,2)
22	resume_chap_presente(3,2)	afficher_resume_chap(3,2)
23	presenter_resume_chap(4,1)	resume_chap_presente(4,1)
24	presenter_resume_chap(4,1)	afficher_resume_chap(4,1)
25	presenter_uah(1,1,1)	uah_presente(1,1,1)
26	presenter_uah(1,1,1)	afficher_uah(1,1,1)
27	presenter_uah(1,1,2)	uah_presente(1,1,2)
28	presenter_uah(1,1,2)	afficher_uah(1,1,2)
29	presenter_uah(1,1,3)	uah_presente(1,1,3)
30	presenter_uah(1,1,3)	afficher_uah(1,1,3)

31	presenter_uah(1,1,4)	uah_presente(1,1,4)
32	presenter_uah(1,1,4)	afficher_uah(1,1,4)
33	presenter_uah(1,1,5)	uah_presente(1,1,5)
34	presenter_uah(1,1,5)	afficher_uah(1,1,5)
35	presenter_uah(1,2,1)	uah_presente(1,2,1)
36	presenter_uah(1,2,1)	afficher_uah(1,2,1)
37	presenter_uah(1,2,2)	uah_presente(1,2,2)
38	presenter_uah(1,2,2)	afficher_uah(1,2,2)
39	presenter_uah(1,2,3)	uah_presente(1,2,3)
40	presenter_uah(1,2,3)	afficher_uah(1,2,3)
41	presenter_uah(1,2,4)	uah_presente(1,2,4)
42	presenter_uah(1,2,4)	afficher_uah(1,2,4)
43	presenter_uah(1,3,1)	uah_presente(1,3,1)
44	presenter_uah(1,3,1)	afficher_uah(1,3,1)
45	presenter_uah(1,3,2)	uah_presente(1,3,2)
46	presenter_uah(1,3,2)	afficher_uah(1,3,2)
47	presenter_uah(1,3,3)	uah_presente(1,3,3)
48	presenter_uah(1,3,3)	afficher_uah(1,3,3)
49	presenter_uah(1,3,4)	uah_presente(1,3,4)
50	presenter_uah(1,3,4)	afficher_uah(1,3,4)
51	presenter_uah(2,1,1)	uah_presente(2,1,1)
52	presenter_uah(2,1,1)	afficher_uah(2,1,1)
53	presenter_uah(2,1,2)	uah_presente(2,1,2)
54	presenter_uah(2,1,2)	afficher_uah(2,1,2)
55	presenter_uah(2,1,3)	uah_presente(2,1,3)
56	presenter_uah(2,1,3)	afficher_uah(2,1,3)
57	presenter_uah(2,1,4)	uah_presente(2,1,4)
58	presenter_uah(2,1,4)	afficher_uah(2,1,4)
59	presenter_uah(2,1,5)	uah_presente(2,1,5)
60	presenter_uah(2,1,5)	afficher_uah(2,1,5)
61	presenter_uah(2,2,1)	uah_presente(2,2,1)
62	presenter_uah(2,2,1)	afficher_uah(2,2,1)
63	presenter_uah(2,2,2)	uah_presente(2,2,2)
64	presenter_uah(2,2,2)	afficher_uah(2,2,2)
65	presenter_uah(2,2,3)	uah_presente(2,2,3)
66	presenter_uah(2,2,3)	afficher_uah(2,2,3)
67	presenter_uah(2,2,4)	uah_presente(2,2,4)
68	presenter_uah(2,2,4)	afficher_uah(2,2,4)
69	presenter_uah(2,2,5)	uah_presente(2,2,5)
70	presenter_uah(2,2,5)	afficher_uah(2,2,5)
71	presenter_uah(2,2,6)	uah_presente(2,2,6)

72	presenter_uah(2,2,6)	afficher_uah(2,2,6)
73	presenter_uah(3,1,1)	uah_presente(3,1,1)
74	presenter_uah(3,1,1)	afficher_uah(3,1,1)
75	presenter_uah(3,1,2)	uah_presente(3,1,2)
76	presenter_uah(3,1,2)	afficher_uah(3,1,2)
77	presenter_uah(3,1,3)	uah_presente(3,1,3)
78	presenter_uah(3,1,3)	afficher_uah(3,1,3)
79	presenter_uah(3,1,4)	uah_presente(3,1,4)
80	uah_presente(3,1,4)	afficher_uah(3,1,4)
81	presenter_uah(3,1,5)	uah_presente(3,1,5)
82	uah_presente(3,1,5)	afficher_uah(3,1,5)
83	presenter_uah(3,2,1)	uah_presente(3,2,1)
84	presenter_uah(3,2,1)	afficher_uah(3,2,1)
85	presenter_uah(3,2,2)	uah_presente(3,2,2)
86	presenter_uah(3,2,2)	afficher_uah(3,2,2)
87	presenter_uah(3,2,3)	Uah_presente(3,2,3)
88	uah_presente(3,2,3)	afficher_uah(3,2,3)
89	presenter_uah(4,1,1)	uah_presente(4,1,1)
90	presenter_uah(4,1,1)	afficher_uah(4,1,1)
91	presenter_uah(4,1,2)	uah_presente(4,1,2)
92	presenter_uah(4,1,2)	afficher_uah(4,1,2)
93	presenter_uah(4,1,3)	uah_presente(4,1,3)
94	presenter_uah(4,1,3)	afficher_uah(4,1,3)
95	tester_par(0)	afficher_test_par(0)
96	pretest_module(1)	afficher_test_par(0)
97	tester_par(1)	afficher_test_par(1)
98	tester_uah(0,0,0)	afficher_test_uah(0,0,0)
99	tester_uah(1,1,3)	afficher_test_uah(1,1,3)
100	tester_uah(1,2,1)	afficher_test_uah(1,2,1)
101	tester_uah(1,2,3)	afficher_test_uah(1,2,3)
102	tester_uah(1,3,1)	afficher_test_uah(1,3,1)
103	tester_uah(1,3,2)	afficher_test_uah(1,3,2)
104	tester_uah(1,3,3)	afficher_test_uah(1,3,3)
105	tester_uah(1,3,4)	afficher_test_uah(1,3,4)
106	tester_uah(2,1,1)	afficher_test_uah(2,1,1)
107	tester_uah(2,1,4)	afficher_test_uah(2,1,4)
108	tester_uah(2,2,2)	afficher_test_uah(2,2,2)
109	tester_uah(2,2,3)	afficher_test_uah(2,2,3)
110	tester_uah(2,2,4)	afficher_test_uah(2,2,4)

111	tester_uah(2,2,6)	afficher_test_uah(2,2,6)
112	tester_uah(3,1,1)	afficher_test_uah(3,1,1)
113	tester_uah(3,2,3)	afficher_test_uah(3,2,3)

Base de règles de négociation		
NR	Condition	Conclusion
1	commencer_par(1)	tester_par(0)
2	commencer_par(2)	tester_par(1)
3	commencer_par(3)	tester_par(2)
4	commencer_par(4)	tester_par(3)
5	commencer_chap(1,1)	tester_par(0)
6	commencer_chap(2,1)	tester_par(1)
7	commencer_chap(3,1)	tester_par(2)
8	commencer_chap(4,1)	tester_par(3)
9	commencer_chap(1,2)	tester_chap(1,1)
10	commencer_chap(1,3)	tester_chap(1,2)
11	commencer_chap(2,2)	tester_chap(2,1)
12	commencer_chap(3,2)	tester_chap(3,1)
13	commencer_uah(1,1,1)	tester_par(0)
14	commencer_uah(2,1,1)	tester_par(1)
15	commencer_uah(3,1,1)	tester_par(2)
16	commencer_uah(4,1,1)	tester_par(3)
17	commencer_uah(1,2,1)	tester_chap(1,1)
18	commencer_uah(1,3,1)	tester_chap(1,2)
19	commencer_uah(2,2,1)	tester_chap(2,1)
20	commencer_uah(3,2,1)	tester_chap(3,1)
21	commencer_uah(1,1,2)	tester_uah(1,1,1)
22	commencer_uah(1,1,3)	tester_uah(1,1,2)
23	commencer_uah(1,1,4)	tester_uah(1,1,3)
24	commencer_uah(1,1,5)	tester_uah(1,1,4)
25	commencer_uah(1,2,2)	tester_uah(1,2,1)
26	commencer_uah(1,2,3)	tester_uah(1,2,2)
27	commencer_uah(1,2,4)	tester_uah(1,2,3)
28	commencer_uah(1,3,2)	tester_uah(1,3,1)
29	commencer_uah(1,3,3)	tester_uah(1,3,2)
30	commencer_uah(1,3,4)	tester_uah(1,3,3)
31	commencer_uah(2,1,2)	tester_uah(2,1,1)
32	commencer_uah(2,1,3)	tester_uah(2,1,2)
33	commencer_uah(2,1,4)	tester_uah(2,1,3)
34	commencer_uah(2,1,5)	tester_uah(2,1,4)
35	commencer_uah(2,2,2)	tester_uah(2,2,1)

36	commencer_uah(2,2,3)	tester_uah(2,2,2)
37	commencer_uah(2,2,4)	tester_uah(2,2,3)
38	commencer_uah(2,2,5)	tester_uah(2,2,4)
39	commencer_uah(2,2,6)	tester_uah(2,2,5)
40	commencer_uah(3,1,2)	tester_uah(3,1,1)
41	commencer_uah(3,1,3)	tester_uah(3,1,2)
42	commencer_uah(3,1,4)	tester_uah(3,1,3)
43	commencer_uah(3,1,5)	tester_uah(3,1,4)
44	commencer_uah(3,2,2)	tester_uah(3,2,1)
45	commencer_uah(3,2,3)	tester_uah(3,2,2)
46	commencer_uah(4,1,2)	tester_uah(4,1,1)
47	commencer_uah(4,1,3)	tester_uah(4,1,2)
48	tester_par(0)	tester_chap(0,0)
49	tester_par(1)	tester_chap(1,3)
50	tester_par(2)	tester_chap(2,2)
51	tester_par(3)	tester_chap(3,2)
52	tester_par(4)	tester_chap(4,1)
53	tester_chap(0,0)	tester_uah(0,0,0)
54	tester_chap(1,1)	tester_uah(1,1,5)
55	tester_chap(1,2)	tester_uah(1,2,4)
56	tester_chap(1,3)	tester_uah(1,3,4)
57	tester_chap(2,1)	tester_uah(2,1,5)
58	tester_chap(2,2)	tester_uah(2,2,6)
59	tester_chap(3,1)	tester_uah(3,1,5)
60	tester_chap(3,2)	tester_uah(3,2,3)
61	tester_chap(4,1)	tester_uah(4,1,3)
62	tester_uah(1,1,1)	uah_acquis(1,1,1)
63	not uah_acquis(1,1,1)	tester_par(0)
64	not uah_acquis(1,1,1)	tester_par(0)
65	tester_uah(1,1,2)	uah_acquis(1,1,2)
66	uah_acquis(1,1,2)	tester_uah(1,1,1)
67	tester_uah(1,1,4)	uah_acquis(1,1,4)
68	uah_acquis(1,1,4)	tester_uah(1,1,3)
69	tester_uah(1,1,5)	uah_acquis(1,1,5)
70	uah_acquis(1,1,5)	tester_uah(1,1,4)
71	tester_uah(1,2,2)	uah_acquis(1,2,2)
72	uah_acquis(1,2,2)	tester_uah(1,2,1)
73	tester_uah(1,2,4)	uah_acquis(1,2,4)
74	uah_acquis(1,2,4)	tester_uah(1,2,3)
75	tester_uah(2,1,2)	uah_acquis(2,1,2)
76	uah_acquis(2,1,2)	tester_uah(2,1,1)
77	tester_uah(2,1,3)	uah_acquis(2,1,3)

78	uah_acquis(2,1,3)	tester_uah(2,1,2)
79	tester_uah(2,1,5)	uah_acquis(2,1,5)
80	uah_acquis(2,1,5)	tester_uah(2,1,4)
81	tester_uah(2,2,1)	uah_acquis(2,2,1)
82	uah_acquis(2,2,1)	tester_chap(2,1)
83	tester_uah(2,2,5)	uah_acquis(2,2,5)
84	uah_acquis(2,2,5)	tester_uah(2,2,4)
85	tester_uah(3,1,2)	uah_acquis(3,1,2)
86	uah_acquis(3,1,2)	tester_uah(3,1,1)
87	tester_uah(3,1,3)	uah_acquis(3,1,3)
88	uah_acquis(3,1,3)	tester_uah(3,1,2)
89	tester_uah(3,1,4)	uah_acquis(3,1,4)
90	uah_acquis(3,1,4)	tester_uah(3,1,3)
91	tester_uah(3,1,5)	uah_acquis(3,1,5)
92	uah_acquis(3,1,5)	tester_uah(3,1,4)
93	tester_uah(3,2,1)	uah_acquis(3,2,1)
94	uah_acquis(3,2,1)	tester_chap(3,1)
95	tester_uah(3,2,2)	uah_acquis(3,2,2)
96	uah_acquis(3,2,2)	tester_uah(3,2,1)
97	tester_uah(4,1,1)	uah_acquis(4,1,1)
98	uah_acquis(4,1,1)	tester_par(3)
99	tester_uah(4,1,2)	uah_acquis(4,1,2)
100	uah_acquis(4,1,2)	tester_uah(4,1,1)
101	tester_uah(4,1,3)	uah_acquis(4,1,3)
102	uah_acquis(4,1,3)	tester_uah(4,1,2)
103	not uah_acquis(1,1,1)	tester_par(0)
104	not uah_acquis(2,1,1)	tester_par(1)
105	not uah_acquis(3,1,1)	tester_par(2)
106	not uah_acquis(4,1,1)	tester_par(3)
107	not uah_acquis(1,2,1)	tester_chap(1,1)
108	not uah_acquis(1,3,1)	tester_chap(1,2)
109	not uah_acquis(2,2,1)	tester_chap(2,1)
110	not uah_acquis(3,2,1)	tester_chap(3,1)
111	not uah_acquis(1,1,2)	tester_uah(1,1,1)
112	not uah_acquis(1,1,3)	tester_uah(1,1,2)
113	not uah_acquis(1,1,4)	tester_uah(1,1,3)
114	not uah_acquis(1,1,5)	tester_uah(1,1,4)
115	not uah_acquis(1,2,2)	tester_uah(1,2,1)
116	not uah_acquis(1,2,3)	tester_uah(1,2,2)
117	not uah_acquis(1,2,4)	tester_uah(1,2,3)
118	not uah_acquis(1,3,2)	tester_uah(1,3,1)
119	not uah_acquis(1,3,3)	tester_uah(1,3,2)

120	not uah_acquis(1,3,4)	tester_uah(1,3,3)
121	not uah_acquis(2,1,2)	tester_uah(2,1,1)
122	not uah_acquis(2,1,3)	tester_uah(2,1,2)
123	not uah_acquis(2,1,4)	tester_uah(2,1,3)
124	not uah_acquis(2,1,5)	tester_uah(2,1,4)
125	not uah_acquis(2,2,2)	tester_uah(2,2,1)
126	not uah_acquis(2,2,3)	tester_uah(2,2,2)
127	not uah_acquis(2,2,4)	tester_uah(2,2,3)
128	not uah_acquis(2,2,5)	tester_uah(2,2,4)
129	not uah_acquis(2,2,6)	tester_uah(2,2,5)
130	not uah_acquis(3,1,2)	tester_uah(3,1,1)
131	not uah_acquis(3,1,3)	tester_uah(3,1,2)
132	not uah_acquis(3,1,4)	tester_uah(3,1,3)
133	not uah_acquis(3,1,5)	tester_uah(3,1,4)
134	not uah_acquis(3,2,2)	tester_uah(3,2,1)
135	not uah_acquis(3,2,3)	tester_uah(3,2,2)
136	not uah_acquis(4,1,2)	tester_uah(4,1,1)
137	not uah_acquis(4,1,3)	tester_uah(4,1,2)
138	commencer_module(1)	pretest_module(1)

Base de règles de planification

NR	Condition	Conclusion
1	Demarrage	tester_uah(0,0,0)
2	uah_acquis(0,0,0)	Chap_acquis(0,0)
3	not uah_acquis(0,0,0)	tester_uah(0,0,0)
4	chap_acquis(0,0)	par_acquis(0)
5	par_acquis(0)	presenter_resume_par(1)
6	par_acquis(1)	presenter_resume_par(2)
7	par_acquis(2)	presenter_resume_par(3)
8	par_acquis(3)	presenter_resume_par(4)
9	par_acquis(4)	Cours_acquis
10	resume_par_presente(1)	presenter_resume_chap(1,1)
11	resume_par_presente(2)	presenter_resume_chap(2,1)
12	resume_par_presente(3)	presenter_resume_chap(3,1)
13	resume_par_presente(4)	presenter_resume_chap(4,1)
14	chap_acquis(1,1)	presenter_resume_chap(1,2)
15	chap_acquis(1,2)	presenter_resume_chap(1,3)
16	chap_acquis(2,1)	presenter_resume_chap(2,2)
17	chap_acquis(3,1)	presenter_resume_chap(3,2)
18	resume_chap_presente(1,1)	presenter_uah(1,1,1)
19	resume_chap_presente(1,2)	presenter_uah(1,2,1)

20	resume_chap_presente(1,3)	presenter_uah(1,3,1)
21	resume_chap_presente(2,1)	presenter_uah(2,1,1)
22	resume_chap_presente(2,2)	presenter_uah(2,2,1)
23	resume_chap_presente(3,1)	presenter_uah(3,1,1)
24	resume_chap_presente(3,2)	presenter_uah(3,2,1)
25	resume_chap_presente(4,1)	presenter_uah(4,1,1)
26	uah_acquis(1,1,1)	presenter_uah(1,1,2)
27	uah_acquis(1,1,2)	presenter_uah(1,1,3)
28	uah_acquis(1,1,3)	presenter_uah(1,1,4)
29	uah_acquis(1,1,4)	presenter_uah(1,1,5)
30	not uah_acquis(1,1,3)	presenter_uah(1,1,3)
31	uah_acquis(1,2,1)	presenter_uah(1,2,2)
32	uah_acquis(1,2,2)	presenter_uah(1,2,3)
33	uah_acquis(1,2,3)	presenter_uah(1,2,4)
34	not uah_acquis(1,2,1)	presenter_uah(1,2,1)
35	not uah_acquis(1,2,3)	presenter_uah(1,2,3)
36	uah_acquis(1,3,1)	presenter_uah(1,3,2)
37	uah_acquis(1,3,2)	presenter_uah(1,3,3)
38	uah_acquis(1,3,3)	presenter_uah(1,3,4)
39	not uah_acquis(1,3,1)	presenter_uah(1,3,1)
40	not uah_acquis(1,3,2)	presenter_uah(1,3,2)
41	not uah_acquis(1,3,3)	presenter_uah(1,3,3)
42	not uah_acquis(1,3,4)	presenter_uah(1,3,4)
43	uah_acquis(2,1,1)	presenter_uah(2,1,2)
44	uah_acquis(2,1,2)	presenter_uah(2,1,3)
45	uah_acquis(2,1,3)	presenter_uah(2,1,4)
46	uah_acquis(2,1,4)	presenter_uah(2,1,5)
47	not uah_acquis(2,1,1)	presenter_uah(2,1,1)
48	not uah_acquis(2,1,4)	presenter_uah(2,1,4)
49	uah_acquis(2,2,1)	presenter_uah(2,2,2)
50	uah_acquis(2,2,2)	presenter_uah(2,2,3)
51	uah_acquis(2,2,3)	presenter_uah(2,2,4)
52	uah_acquis(2,2,4)	presenter_uah(2,2,5)
53	uah_acquis(2,2,5)	presenter_uah(2,2,6)
54	not uah_acquis(2,2,2)	presenter_uah(2,2,2)
55	not uah_acquis(2,2,3)	presenter_uah(2,2,3)
56	not uah_acquis(2,2,4)	presenter_uah(2,2,4)
57	not uah_acquis(2,2,6)	presenter_uah(2,2,6)
58	uah_acquis(3,1,1)	presenter_uah(3,1,2)
59	uah_acquis(3,1,2)	presenter_uah(3,1,3)
60	uah_acquis(3,1,3)	presenter_uah(3,1,4)
61	uah_acquis(3,1,4)	presenter_uah(3,1,5)

62	not_uah_acquis(3,1,1)	presenter_uah(3,1,1)
63	uah_acquis(3,2,1)	presenter_uah(3,2,2)
64	uah_acquis(3,2,2)	presenter_uah(3,2,3)
65	not_uah_acquis(3,2,3)	presenter_uah(3,2,3)
66	uah_acquis(4,1,1)	presenter_uah(4,1,2)
67	uah_acquis(4,1,2)	presenter_uah(4,1,3)
68	uah_presente(1,1,1)	uah_acquis(1,1,1)
69	uah_presente(1,1,2)	uah_acquis(1,1,2)
70	uah_presente(1,1,3)	Tester_uah(1,1,3)
71	uah_presente(1,1,4)	uah_acquis(1,1,4)
72	uah_presente(1,1,5)	uah_acquis(1,1,5)
73	uah_presente(1,2,1)	Tester_uah(1,2,1)
74	uah_presente(1,2,2)	uah_acquis(1,2,2)
75	uah_presente(1,2,3)	Tester_uah(1,2,3)
76	uah_presente(1,2,4)	uah_acquis(1,2,4)
77	uah_presente(1,3,1)	Tester_uah(1,3,1)
78	uah_presente(1,3,2)	Tester_uah(1,3,2)
79	uah_presente(1,3,3)	Tester_uah(1,3,3)
80	uah_presente(1,3,4)	Tester_uah(1,3,4)
81	uah_presente(2,1,1)	Tester_uah(2,1,1)
82	uah_presente(2,1,2)	uah_acquis(2,1,2)
83	uah_presente(2,1,3)	uah_acquis(2,1,3)
84	uah_presente(2,1,4)	Tester_uah(2,1,4)
85	uah_presente(2,1,5)	uah_acquis(2,1,5)
86	uah_presente(2,2,1)	uah_acquis(2,2,1)
87	uah_presente(2,2,2)	Tester_uah(2,2,2)
88	uah_presente(2,2,3)	Tester_uah(2,2,3)
89	uah_presente(2,2,4)	Tester_uah(2,2,4)
90	uah_presente(2,2,5)	uah_acquis(2,2,5)
91	uah_presente(2,2,6)	Tester_uah(2,2,6)
92	uah_presente(3,1,1)	Tester_uah(3,1,1)
93	uah_presente(3,1,2)	uah_acquis(3,1,2)
94	uah_presente(3,1,3)	uah_acquis(3,1,3)
95	uah_presente(3,1,4)	uah_acquis(3,1,4)
96	uah_presente(3,1,5)	uah_acquis(3,1,5)
97	uah_presente(3,2,1)	uah_acquis(3,2,1)
98	uah_presente(3,2,2)	uah_acquis(3,2,2)
99	uah_presente(3,2,3)	Tester_uah(3,2,3)
100	uah_presente(4,1,1)	uah_acquis(4,1,1)
101	uah_presente(4,1,2)	uah_acquis(4,1,2)
102	uah_presente(4,1,3)	uah_acquis(4,1,3)
103	uah_acquis(1,1,5)	Chap_acquis(1,1)

104	uah_acquis(1,2,4)	Chap_acquis(1,2)
105	uah_acquis(1,3,4)	Chap_acquis(1,3)
106	uah_acquis(2,1,5)	Chap_acquis(2,1)
107	uah_acquis(2,2,6)	Chap_acquis(2,2)
108	uah_acquis(3,1,5)	Chap_acquis(3,1)
109	uah_acquis(3,2,3)	Chap_acquis(3,2)
110	uah_acquis(4,1,3)	Chap_acquis(4,1)
111	chap_acquis(1,3)	par_acquis(1)
112	chap_acquis(2,2)	par_acquis(2)
113	chap_acquis(3,2)	par_acquis(3)
114	chap_acquis(4,1)	par_acquis(4)

Base de règles d'acquisition		
NR	Condition	Conclusion
1	par_acquis(1)	par_acquis(0)
2	par_acquis(2)	par_acquis(1)
3	par_acquis(3)	par_acquis(2)
4	par_acquis(4)	par_acquis(3)
5	par_acquis(5)	par_acquis(4)
6	par_acquis(0)	Chap_acquis(0,0)
7	par_acquis(1)	Chap_acquis(1,3)
8	par_acquis(2)	Chap_acquis(2,2)
9	par_acquis(3)	Chap_acquis(3,2)
10	par_acquis(4)	Chap_acquis(4,1)
11	chap_acquis(1,1)	par_acquis(0)
12	chap_acquis(2,1)	par_acquis(1)
13	chap_acquis(3,1)	par_acquis(2)
14	chap_acquis(4,1)	par_acquis(3)
15	chap_acquis(1,2)	Chap_acquis(1,1)
16	chap_acquis(1,3)	Chap_acquis(1,2)
17	chap_acquis(2,2)	Chap_acquis(2,1)
18	chap_acquis(3,2)	Chap_acquis(3,1)
19	chap_acquis(0,0)	uah_acquis(0,0,0)
20	chap_acquis(1,1)	uah_acquis(1,1,5)
21	chap_acquis(1,2)	uah_acquis(1,2,4)
22	chap_acquis(1,3)	uah_acquis(1,3,4)
23	chap_acquis(2,1)	uah_acquis(2,1,5)
24	chap_acquis(2,2)	uah_acquis(2,2,6)
25	chap_acquis(3,1)	uah_acquis(3,1,5)
26	chap_acquis(3,2)	uah_acquis(3,2,3)
27	chap_acquis(4,1)	uah_acquis(4,1,3)

28	uah_acquis(1,1,1)	par_acquis(0)
29	uah_acquis(2,1,1)	par_acquis(1)
30	uah_acquis(3,1,1)	par_acquis(2)
31	uah_acquis(4,1,1)	par_acquis(3)
32	uah_acquis(1,2,1)	Chap_acquis(1,1)
33	uah_acquis(1,3,1)	Chap_acquis(1,2)
34	uah_acquis(2,2,1)	Chap_acquis(2,1)
35	uah_acquis(3,2,1)	Chap_acquis(3,1)
36	uah_acquis(1,1,2)	uah_acquis(1,1,1)
37	uah_acquis(1,1,3)	uah_acquis(1,1,2)
38	uah_acquis(1,1,4)	uah_acquis(1,1,3)
39	uah_acquis(1,1,5)	uah_acquis(1,1,4)
40	uah_acquis(1,2,2)	uah_acquis(1,2,1)
41	uah_acquis(1,2,3)	uah_acquis(1,2,2)
42	uah_acquis(1,2,4)	uah_acquis(1,2,3)
43	uah_acquis(1,3,2)	uah_acquis(1,3,1)
44	uah_acquis(1,3,3)	uah_acquis(1,3,2)
45	uah_acquis(1,3,4)	uah_acquis(1,3,3)
46	uah_acquis(2,1,2)	uah_acquis(2,1,1)
47	uah_acquis(2,1,3)	uah_acquis(2,1,2)
48	uah_acquis(2,1,4)	uah_acquis(2,1,3)
49	uah_acquis(2,1,5)	uah_acquis(2,1,4)
50	uah_acquis(2,2,2)	uah_acquis(2,2,1)
51	uah_acquis(2,2,3)	uah_acquis(2,2,2)
52	uah_acquis(2,2,4)	uah_acquis(2,2,3)
53	uah_acquis(2,2,5)	uah_acquis(2,2,4)
54	uah_acquis(2,2,6)	uah_acquis(2,2,5)
55	uah_acquis(3,1,2)	uah_acquis(3,1,1)
56	uah_acquis(3,1,3)	uah_acquis(3,1,2)
57	uah_acquis(3,1,4)	uah_acquis(3,1,3)
58	uah_acquis(3,1,5)	uah_acquis(3,1,4)
59	uah_acquis(3,2,2)	uah_acquis(3,2,1)
60	uah_acquis(3,2,3)	uah_acquis(3,2,2)
61	uah_acquis(4,1,2)	uah_acquis(4,1,1)
62	uah_acquis(4,1,3)	uah_acquis(4,1,2)

