



HAL
open science

Processus unifié pour la personnalisation des activités pédagogiques : méta-modèle, modèles et outils

Marie Lefevre

► **To cite this version:**

Marie Lefevre. Processus unifié pour la personnalisation des activités pédagogiques : méta-modèle, modèles et outils. Ordinateur et société [cs.CY]. Université Claude Bernard - Lyon I, 2009. Français. NNT : 2009LYO10262 . tel-00625465

HAL Id: tel-00625465

<https://theses.hal.science/tel-00625465>

Submitted on 21 Sep 2011

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Processus unifié pour la personnalisation des activités pédagogiques : méta-modèle, modèles et outils.

Thèse pour l'obtention du

DOCTORAT DE L'UNIVERSITÉ CLAUDE BERNARD LYON 1

spécialité informatique

soutenue le 1^{er} décembre 2009

par

Marie LEFEVRE

Composition du jury

Rapporteurs :

Monique GRANDBASTIEN
Philippe VIDAL

Professeur - Université Henri Poincaré Nancy 1, France
Professeur - Université Paul Sabatier Toulouse, France

Examineurs :

Thierry NODENOT
Jean-Charles MARTY

Professeur - Université de Pau et des Pays de l'Adour, France
Maître de Conférences HDR - Université de Savoie, France

Directeurs :

Alain MILLE
Stéphanie JEAN-DAUBIAS
Nathalie GUIN

Professeur - Université Claude Bernard Lyon 1, France
Maître de Conférences - Université Claude Bernard Lyon 1, France
Maître de Conférences - Université Claude Bernard Lyon 1, France

REMERCIEMENTS

Je souhaite tout d'abord remercier les membres du jury : Monique Grandbastien et Philippe Vidal d'avoir accepté de rapporter cette thèse et d'avoir eu le courage de lire ce (petit) manuscrit, ainsi que Thierry Nodenot d'avoir accepté d'être examinateur et président de jury.

Merci ensuite à Jean-Charles. J'ai eu la chance de t'avoir en tant qu'enseignant durant mes premières années de fac. Tu as fait naître chez moi l'envie d'être informaticienne.

Merci aussi à Alain de m'avoir acceptée dans ton équipe, de m'avoir laissée faire mes travaux de recherche, sans faire, ni utiliser directement des traces... même si trois ans après, je vois maintenant toute la richesse et toute la plus value que cela pourrait apporter.

Merci enfin à mes deux encadrantes. Nathalie, pour m'avoir transmis un peu de tes connaissances et pour m'avoir guidée dans l'utilisation de ce langage immonde que tu maîtrises si bien. Stéphanie, pour avoir transformé l'étudiante informaticienne en chercheuse, pour m'avoir transmis ta connaissance du domaine, pour avoir toujours cru en moi, pour m'avoir laissé cette liberté d'action et de penser, et pour tous ces crayons courageusement sacrifiés aux pieds de ces tonnes de papier... tu n'as donc jamais pensé à ces pauvres arbres qui ne nous avaient rien fait ! Merci à toutes les deux de vous être relayées, malgré les difficultés de la vie, dans mon encadrement.

Je remercie aussi les membres de l'équipe Silex pour leurs discussions et conseils lors des réunions où je présentais mon travail : Jacques, Yannick, Julien, Magali, Amélie... et plus particulièrement Pierre-Antoine pour m'avoir permis de faire des choix techniques pertinents quand la question s'est posée.

Merci enfin à Brigitte et Sylvie d'avoir rendu si facile tout le côté administratif de la recherche.

Ensuite vient le tour des amis, sans qui cette thèse aurait parue interminable. Merci Amélie pour ta présence, jour et nuit, pour tes conseils, pour notre mur de traces si riche, et pour tous les palmorals et autres délices que tu sais si bien faire. Merci Marianne d'être là depuis si longtemps maintenant, pour ces pauses interminables qui permettent de mieux repartir.

Merci aux joueurs de cartes : Erwan, Mathieu, Brice, Benoît, encore Marianne, et tous les autres. Merci à tous ceux qui ont rendu ces trois ans au labo si particulier : Magali, Damien, Carole, mais aussi Erwan, Mathieu, Brice, Benoît, Marianne, Amélie, Nathalie, Stéphanie (oui je sais, je radote).

Pour finir, la famille. Mes parents, qui m'ont permis d'arriver jusque là. Ma maman, qui a su s'occuper de mon petit homme quand ma thèse était trop prenante. Mes frangins, là depuis toujours. Dominique et Christian pour avoir cru en moi...

Enfin, et surtout mes hommes. Olivier, merci de me supporter depuis si longtemps, de m'avoir suivi à Lyon et d'être encore prêt à me suivre dans nos futurs périple. Hugo, pour avoir donné un sens à tout ça, pour m'avoir donné l'envie de terminer quand tout était si compliqué. J'espère que tu ne garderas pas de souvenir traumatisant de cette période :-).

Et pour finir, je souhaite remercier tous ces gens qui m'ont dit que ce que je voulais faire était impossible... ils m'ont donné la force d'avancer !

PLAN DU MANUSCRIT

Introduction	23
Domaine de recherche	23
Problématique	23
Contribution	24
Cadre du travail.....	25
Partenariat avec des enseignants.....	26
Scénarios d'usage	27
Plan de la thèse.....	30
Partie 1. État de l'art.....	31
Chapitre 1. Comment personnaliser l'apprentissage ?.....	33
1.1. Introduction	35
1.2. Conseiller l'enseignant sur le contenu de son enseignement.....	35
1.2.1. Proposition des thèmes à étudier par l'apprenant.....	35
1.2.2. Proposition d'outils à faire utiliser par l'apprenant	36
1.3. Créer des ressources pédagogiques	37
1.3.1. Création d'exercices	37
1.3.2. Création de listes d'exercices	45
1.3.3. Création de logiciels pédagogiques.....	45
1.4. Adapter les EIAH à chaque apprenant.....	47
1.4.1. Que peut-on personnaliser dans un EIAH ?.....	47
1.4.2. Qui personnalise l'EIAH ?.....	49
1.5. Scénariser les séances d'apprentissage	54
1.5.1. Définition du concept de scénario.....	54
1.5.2. Création et échange de scénarios	55
1.6. Synthèse des approches relatives à la personnalisation de l'apprentissage.....	55
Chapitre 2. Comment piloter un système informatique de façon externalisée ? Et plus particulièrement un EIAH ?.....	59
2.1. Introduction	61
2.2. Piloter un système informatique de façon externalisée.....	61
2.2.1. Approche des systèmes épiphytes	61
2.2.2. Approche par composants.....	62
2.2.3. Bilan	63
2.3. Décrire un EIAH à l'aide de métadonnées.....	63
2.3.1. Standards orientés description de contenus	63

2.3.2.	LSCM - Schéma de description des composants logiciels.....	67
2.3.3.	Synthèse des approches relatives à la description d'EIAH.....	70
Chapitre 3.	Analyse de l'existant, synthèse et proposition.....	71
3.1.	Retour sur les scénarios d'usage.....	73
3.2.	Verrous.....	74
3.3.	Notre approche.....	75
Partie 2.	Contributions théoriques.....	79
Chapitre 4.	PERSUA2 : un modèle pour une personnalisation unifiée des activités d'apprentissage.....	81
4.1.	Introduction.....	83
4.2.	Principe du modèle PERSUA2.....	83
4.3.	Définition formelle du modèle PERSUA2.....	84
4.4.	Principe du processus d'exploitation du modèle PERSUA2.....	87
4.5.	Comparaison de PERSUA2 avec d'autres approches utilisant des règles pédagogiques d'adaptation à l'apprenant.....	89
4.6.	Ce qu'il faut retenir.....	90
Chapitre 5.	cPMDL : un modèle de contraintes sur profils.....	91
5.1.	Introduction.....	93
5.2.	Pourquoi harmoniser la structure des profils ?.....	93
5.3.	Choix d'un formalisme commun pour décrire un profil : PMDL.....	95
5.3.1.	Structuration générale de PMDL.....	95
5.3.2.	Types de données dans PMDL.....	96
5.3.3.	Échelles dans PMDL.....	101
5.4.	Définition formelle du modèle cPMDL.....	102
5.4.1.	Contrainte sur profils portant sur une valeur.....	102
5.4.2.	Contrainte sur profils portant sur un élément.....	105
5.4.3.	Contrainte sur profils portant sur un nombre d'occurrences.....	109
5.4.4.	Combinaison de contraintes sur profils.....	110
5.5.	Ce qu'il faut retenir.....	112
Chapitre 6.	GEPPETO : des modèles et processus pour adapter des activités pédagogiques.....	113
6.1.	Introduction.....	117
6.2.	Principe de l'approche GEPPETO.....	117
6.3.	GEPPETO _p : Adapter des activités papier.....	118
6.3.1.	Personnaliser quoi et comment ?.....	119
6.3.2.	Classification des activités selon la façon de les générer.....	120
6.3.3.	Principe de l'approche GEPPETO _p	127
6.3.4.	Bilan sur l'adaptation des activités papier.....	148

6.4.	GEPPETO _S : adapter des EIAH.....	149
6.4.1.	Personnaliser quoi et comment ?	149
6.4.2.	Classification des EIAH selon leur personnalisation	152
6.4.3.	Principe de l'approche GEPPETO _S	153
6.4.4.	Bilan sur l'adaptation d'EIAH	175
6.5.	Ce qu'il faut retenir.....	177
Chapitre 7.	Processus de création de séquences personnalisées d'activités	179
7.1.	Introduction	181
7.2.	Principe d'exploitation du modèle PERSUA2	181
7.3.	Architecture du processus de création de séquences personnalisées d'activités	187
7.3.1.	Filtrage des règles d'affectation	188
7.3.2.	Création des listes d'activités à générer	189
7.3.3.	Génération des activités	189
7.3.4.	Création des séquences de travail personnalisées.....	190
7.4.	Ce qu'il faut retenir.....	191
Partie 3.	Mise en oeuvre & évaluation	195
Chapitre 8.	Adapte, ou comment aider l'enseignant à proposer des activités personnalisées à chacun de ses apprenants	197
8.1.	Introduction	201
8.2.	Fonctionnement du point de vue des utilisateurs.....	201
8.2.1.	Intégration d'un nouvel EIAH.....	202
8.2.2.	Création de séquences de travail personnalisées.....	204
8.2.3.	Conversion d'activités vers une norme pédagogique	217
8.3.	Architecture d'Adapte.....	217
8.3.1.	Processus pour l'intégration d'un nouvel EIAH.....	218
8.3.2.	Processus pour la création de séquences de travail personnalisées	218
8.3.3.	Place de nos contributions théoriques dans Adapte	222
8.4.	Modes d'utilisation d'Adapte.....	223
8.4.1.	Personnalisation des activités : avec intervention de l'enseignant et exploitation de profils d'apprenants	223
8.4.2.	Personnalisation des activités : sans intervention de l'enseignant et exploitation de profils d'apprenants	224
8.4.3.	Personnalisation des activités : avec intervention de l'enseignant sans exploitation de profils d'apprenants	225
8.5.	Retour sur les scénarios d'usage	226
Chapitre 9.	Évaluation.....	229
9.1.	Introduction	233
9.2.	Méthodologie d'évaluation.....	233
9.3.	Grille d'analyse.....	233

9.3.1.	Évaluation de notre approche de la personnalisation de l'apprentissage.....	234
9.3.2.	Évaluation du modèle PERSUA2	234
9.3.3.	Évaluation du modèle cPMDL	235
9.3.4.	Évaluation de l'approche GEPPETO.....	235
9.3.5.	Évaluation de l'approche GEPPETO _P	236
9.3.6.	Évaluation de l'approche GEPPETO _S	237
9.3.7.	Évaluation du logiciel Adapte.....	238
9.4.	Présentation des expérimentations	238
9.4.1.	Mise en oeuvre du méta-modèle AKEPI.....	239
9.4.2.	Re-génération d'activités papier existantes.....	239
9.4.3.	Questionnaire sur les pratiques éducatives.....	240
9.4.4.	Expérimentation d'Adapte avec une enseignante	242
9.5.	Synthèse.....	243
9.5.1.	Évaluation de notre approche de la personnalisation de l'apprentissage.....	243
9.5.2.	Évaluation du modèle PERSUA2	245
9.5.3.	Évaluation du modèle cPMDL	245
9.5.4.	Évaluation de l'approche GEPPETO.....	246
9.5.5.	Évaluation de l'approche GEPPETO _P	246
9.5.6.	Évaluation de l'approche GEPPETO _S	248
9.5.7.	Évaluation du logiciel Adapte.....	249
9.6.	Limites et perspectives d'évaluation	250
9.6.1.	Évaluation du modèle PERSUA2, du modèle cPMDL et de l'approche GEPPETO.....	251
9.6.2.	Évaluation de la typologie d'activités papier	251
9.6.3.	Évaluation du logiciel Adapte.....	252
Conclusion.....		255
Références bibliographiques.....		261
Références netographiques		275
Annexes.....		281
Annexe A.	Corpus de logiciels pédagogiques.....	283
Annexe B.	Exemple de profil exprimé dans le langage PMDL	285
Annexe C.	Formules de conversion d'échelles.....	289
Annexe D.	Exemples d'activités papier	291
Annexe E.	Sémantique de la représentation graphique utilisée pour présenter les modèles.....	305
Annexe F.	Les patrons d'exercices de l'approche GEPPETO _P	307
Annexe G.	Les générateurs d'exercices de l'approche GEPPETO _P	333

Annexe H. Exemple de contenu de fichiers de configuration et identification des invariants du méta-modèle AKEPI.....	345
Annexe I. Modèle OKEP/ABALECT	349
Annexe J. Technologies utilisées lors de la mise en œuvre des processus de GEPPETO _s	361
Annexe K. Questionnaire sur les pratiques éducatives	363
Résumé	368
Abstract	369

TABLE DES ILLUSTRATIONS

Figure 0-1 : Architecture de l'environnement EPROFILEA	25
Figure 1-1 : Méthode pédagogique ATRIUM dans le contexte de la robotique pédagogique [Bruillard et al. 2000].	36
Figure 1-2 : Questions générées par READING TUTOR [Mostow et al. 2004].	38
Figure 1-3 : Patron d'exercices du générateur d'exercices d'APLUSIX [Bouhineau et al. 2005b]. ...	38
Figure 1-4 : Tirages aléatoires pour le patron d'exercices d'APLUSIX de la Figure 1-3 [Bouhineau et al. 2005b].	39
Figure 1-5 : Carte de tests d'APLUSIX [Aplusix 2009].	39
Figure 1-6 : Page énoncé d'un exercice de cinétique chimique créé avec GENEVAL [GenEval 1997].	40
Figure 1-7 : Pages énoncé d'un exercice de mécanique (en arrière plan) et réponse (au premier plan, en bas à droite) créé avec GENEVAL [GenEval 1997].	40
Figure 1-8 : Exercice créé avec EXERCISE GENERATOR PLUS [ExerciseGeneratorPlus].	41
Figure 1-9 : Écran de bilan du générateur de problèmes du module AMBRE-ENSEIGNANT de AMBRE-ADD [Duclosson et al. 2005a].	43
Figure 1-10 : Interface de création d'un exercice de type « Prestidigitateur » dans PÉPIGEN [Prévit et al. 2007].	43
Figure 1-11 : Interface de définition d'un test dans SIETTE [Conejo et al. 2004].	44
Figure 1-12 : Interface de définition d'une question dans SIETTE [Conejo et al. 2004].	44
Figure 1-13 : Éditeur de simulations dans SIMQUEST [Van Joolingen et al. 2003].	46
Figure 1-14 : Simulateur créé avec SIMQUEST [De Jong 2004].	46
Figure 1-15 : KEIGO TUTOR, tuteur intelligent pour l'apprentissage du japonais, créé avec EON [Murray 2003a].	47
Figure 1-16 : CHEMISTRY WORKBENCH, tuteur intelligent de chimie, créé avec EON [Murray 2003a].	47
Figure 1-17 : Écrans d'ANDES permettant à l'apprenant de sélectionner le problème qu'il va résoudre [VanLehn et al. 2005].	49
Figure 1-18 : Menu de LILIMATH permettant à l'apprenant d'obtenir une série d'exercices en fonction d'un thème [LiliMath 2000].	50
Figure 1-19 : Interface d'ACTIVEMATH avec sur la gauche la liste des thèmes disponibles et sur la droite le cours et les exercices associés au thème choisi par l'apprenant [ActiveMath 2007].	50
Figure 1-20 : Interface de description du contenu pédagogique à utiliser dans REDEEM [Ainsworth 2000].	51
Figure 1-21 : Interface de définition d'une stratégie pédagogique dans REDEEM [Ainsworth 2000].	51
Figure 1-22 : Éditeur de méta-stratégies dans EON [Murray 2003a].	52
Figure 2-1 : Métadonnées du Dublin Core [DublinCoreFr 2000].	64
Figure 2-2 : Catégories du LOM [LOM 2002].	65
Figure 2-3 : Métadonnées du LOM [De la Passardière et al. 2004].	66
Figure 2-4 : Rubriques de la section générique « Software Component Metadata » du schéma LSCM [Rebaï 2006].	68
Figure 2-5 : Rubriques de la section spécifique « Learning Metadata » du schéma LSCM [Rebaï 2006].	69
Figure 3-1 : Principe de la personnalisation par un enseignant des activités pédagogiques pour un apprenant.	75
Figure 3-2 : Principe d'une personnalisation unifiée des activités pédagogiques.	76

Figure 4-1 : Principe du modèle PERSUA2.....	83
Figure 4-2 : Représentation graphique du modèle PERSUA2.	87
Figure 4-3 : Principe du processus d'exploitation du modèle PERSUA2.	88
Figure 5-1 : Représentation graphique de la structuration générale de PMDL [Jean-Daubias et al. 2009a].....	96
Figure 5-2 : Profils d'élèves dans J'ADE [J'ADE 2007b].	97
Figure 5-3 : Représentation graphique de l'élément <i>liste_composante</i> de PMDL [Eyssautier-Bavay 2008]......	98
Figure 5-4 : Profil de l'apprenant dans la version mobile de MOREMATHS [Bull et al. 2003]......	99
Figure 5-5 : Représentation graphique de l'élément <i>liste_repartition</i> de PMDL [Eyssautier-Bavay 2008]......	99
Figure 5-6 : Profil de l'apprenant dans PÉPITE [Jean 2000].	100
Figure 5-7 : Représentation graphique de l'élément <i>graphe</i> de PMDL [Eyssautier-Bavay 2008].	100
Figure 5-8 : Profil de l'apprenant issu de pratiques d'enseignants.....	101
Figure 5-9 : Représentation graphique de l'élément <i>echelles</i> de PMDL [Eyssautier-Bavay 2008].	101
Figure 5-10 : Contrainte sur profils portant sur une valeur numérique.....	104
Figure 5-11 : Contrainte sur profils portant sur un entier positif.	104
Figure 5-12 : Représentation graphique de l'élément E4 « Mathématiques » du profil exprimé avec PMDL de l'Annexe B.....	104
Figure 5-13 : Contrainte sur profils portant sur une valeur numérique.....	105
Figure 5-14 : Contrainte sur profils portant sur une valeur textuelle.	105
Figure 5-15 : Exemple d'un profil fictif illustrant les différents éléments de PMDL sur lesquels il est possible de définir une contrainte sur profils portant sur un élément.	106
Figure 5-16 : Types possibles de l'échelle résultante dans une contrainte sur profils portant sur un élément, en fonction des échelles des valeurs contenues dans les sous-éléments.	107
Figure 5-17 : Contrainte sur profils portant sur un élément.	108
Figure 5-18 : Contrainte sur profils portant sur un élément.	109
Figure 5-19 : Contrainte sur profils portant sur un nombre d'occurrences.....	110
Figure 5-20 : Contrainte sur profils portant sur un nombre d'occurrences.....	110
Figure 5-21 : Exemple de combinaison de contraintes sur profils.....	110
Figure 5-22 : Contraintes sur profils portant sur une valeur numérique utilisées dans la contrainte de la Figure 5-21.	111
Figure 5-23 : Contraintes sur profils portant sur une valeur textuelle utilisées dans la contrainte de la Figure 5-21.....	111
Figure 6-1 : Représentation graphique de l'approche GEPPETO.....	118
Figure 6-2 : Typologie des activités papier selon les éléments nécessaires à leur génération....	120
Figure 6-3 : Étapes de génération des activités de la catégorie « Travail sur texte ».	122
Figure 6-4 : Représentation graphique de l'approche GEPPETO _p	128
Figure 6-5 : Personnalisation d'activités papier avec l'approche GEPPETO _p	129
Figure 6-6 : Métadonnées de la structure de patrons.....	130
Figure 6-7 : Patron d'exercices « Travail sur texte ».	131
Figure 6-8 : Choix d'une image dans un patron d'exercices.....	132
Figure 6-9 : Choix d'un texte dans un patron d'exercices.	132
Figure 6-10 : Patron opérationnel « Identification des parties du texte ».	134
Figure 6-11 : Structure d'exercices issue du patron opérationnel « Identification des parties du texte » associé au patron « Travail sur texte ».	135
Figure 6-12 : Structure d'exercices issue du patron opérationnel « Identification des parties du texte » associé au patron « Travail sur texte ».	136
Figure 6-13 : Patron opérationnel « Analyse du texte ».	136

Figure 6-14 : Structure d'exercices issue du patron opérationnel « Analyse du texte » associé au patron « Travail sur texte ».....	137
Figure 6-15 : Patron opérationnel « Sens du texte ».....	138
Figure 6-16 : Structure d'exercices issue du patron opérationnel « Sens du texte » associé au patron « Travail sur texte ».....	139
Figure 6-17 : Structure d'exercices issue du patron opérationnel « Sens du texte » associé au patron « Travail sur texte ».....	139
Figure 6-18 : Patron opérationnel « Transposition du texte ».....	140
Figure 6-19 : Structure d'exercices issue du patron opérationnel « Transposition du texte » associé au patron « Travail sur texte ».....	141
Figure 6-20 : Architecture générique des générateurs d'exercices de l'approche GEPPETo _p	144
Figure 6-21 : Architecture du générateur « Travail sur texte ».....	145
Figure 6-22 : Extrait de la base de textes d'Adapte.....	146
Figure 6-23 : Parallèle entre l'approche GEPPETo et sa déclinaison GEPPETo _p	148
Figure 6-24 : Facettes de la personnalisation d'un EIAH.....	150
Figure 6-25 : Exemples d'EIAH illustrant les moyens de mettre en œuvre les différentes facettes de la personnalisation des EIAH.....	151
Figure 6-26 : Classification des EIAH selon la manière de les personnaliser à travers un système externe à l'EIAH.....	152
Figure 6-27 : Représentation graphique de l'approche GEPPETo _s	154
Figure 6-28 : Étape de rédaction de la solution du problème dans AMBRE-ADD [AMBRE-add 2005].....	155
Figure 6-29 : Écran bilan de AMBRE-ENSEIGNANT récapitulant les contraintes de génération d'un problème [AMBRE-enseignant 2004].....	155
Figure 6-30 : Fichier de configuration du générateur GENAMBRE [AMBRE-enseignant 2004]..	155
Figure 6-31 : Écran de création de séquences d'exercices de AMBRE-ENSEIGNANT [AMBRE-enseignant 2004].....	156
Figure 6-32 : Écran de sauvegarde d'une séquence d'exercices de AMBRE-ENSEIGNANT [AMBRE-enseignant 2004].....	157
Figure 6-33 : Fichier de configuration de AMBRE-ADD contenant les séquences d'exercices créées avec AMBRE-ENSEIGNANT [AMBRE-add 2005].....	157
Figure 6-34 : Écran d'affectation d'une séquence d'exercices à un élève dans AMBRE-ENSEIGNANT [AMBRE-enseignant 2004].....	157
Figure 6-35 : Fichier de configuration de AMBRE-ADD contenant la liste des élèves d'un enseignant et les séquences d'exercices affectées aux élèves dans AMBRE-ENSEIGNANT [AMBRE-add 2005].....	158
Figure 6-36 : Activité de TRI [Jean-Daubias 2007].....	158
Figure 6-37 : Fichier de configuration des séquences de travail de TRI [Jean-Daubias 2007]....	159
Figure 6-38 : Écran de choix d'un exercice par l'apprenant dans ABALECT [Chevé 2005].....	159
Figure 6-39 : Fenêtres de choix d'un texte selon son niveau dans ABALECT [Chevé 2005].....	160
Figure 6-40 : Modèle des propriétés pédagogiques du méta-modèle AKEPI.....	161
Figure 6-41 : Élément <i>PedagogicalContent</i> du méta-modèle AKEPI.....	161
Figure 6-42 : Élément <i>Parameter</i> du méta-modèle AKEPI.....	162
Figure 6-43 : Description d'une échelle de type <i>ScaleList</i> dans le méta-modèle AKEPI.....	163
Figure 6-44 : Description d'une échelle de type <i>ScaleTree</i> dans le méta-modèle AKEPI.....	163
Figure 6-45 : Description d'une échelle de type <i>ScaleBoolean</i> dans le méta-modèle AKEPI.....	163
Figure 6-46 : Description d'une échelle de type <i>ScaleNumerical</i> dans le méta-modèle AKEPI...	163
Figure 6-47 : Description d'une échelle de type <i>ScaleInterval</i> dans le méta-modèle AKEPI.....	164
Figure 6-48 : Description d'une échelle de type <i>ScaleText</i> dans le méta-modèle AKEPI.....	164
Figure 6-49 : Élément <i>PedagogicalOrganization</i> du méta-modèle AKEPI.....	164
Figure 6-50 : Éléments <i>Functionalities</i> , <i>Feedback</i> et <i>Customization</i> du méta-modèle AKEPI.....	165

Figure 6-51 : Format des conditions (partie gauche) et des conclusions (partie droite) des règles pédagogiques du méta-modèle AKEPI.	165
Figure 6-52 : Modèle des propriétés techniques du méta-modèle AKEPI.	166
Figure 6-53: Processus AKASI utilisant le méta-modèle AKEPI pour créer un modèle OKEP/x propre à l'EIAH x.	167
Figure 6-54 : Extrait des propriétés pédagogiques du modèle OKEP/ABALECT.	169
Figure 6-55 : Extrait des règles pédagogiques du modèle OKEP/ABALECT.	170
Figure 6-56 : Propriétés techniques du modèle OKEP/ ABALECT.	171
Figure 6-57 : Règles techniques du modèle OKEP/ABALECT.	171
Figure 6-58 : Extrait des paramètres de création d'un problème avec le générateur GENAMBRE.	172
Figure 6-59 : Extrait des règles pédagogiques du modèle OKEP/AMBRE-ADD.	173
Figure 6-60 : Extrait de la feuille de style créée à partir de la description technique du modèle OKEP/AMBRE-ADD.	174
Figure 6-61 : Processus OPIKSI utilisant le modèle OKEP/x pour personnaliser l'EIAH x.	175
Figure 6-62 : Parallèle entre l'approche GEPPETO et sa déclinaison en GEPPETO _s	176
Figure 6-63 : Synthèse de l'approche GEPPETO et ses deux déclinaisons permettant de personnaliser les activités papier et logicielles.	177
Figure 7-1 : Profils de Jules et Mathilde.	181
Figure 7-2 : Exemple d'un modèle de personnalisation.	183
Figure 7-3 : Schématisation du modèle de personnalisation de la Figure 7-2.	183
Figure 7-4 : Liste des règles d'affectation activées par le profil de Jules et des contraintes sur activités associées.	184
Figure 7-5 : Liste des règles d'affectation activées par le profil de Mathilde et des contraintes sur activités associées.	184
Figure 7-6 : Liste des activités à générer pour Jules.	185
Figure 7-7 : Liste des activités à générer pour Mathilde.	185
Figure 7-8 : Résultat du processus de création de séquences d'activités personnalisées.	186
Figure 7-9 : Principe du modèle de personnalisation et de son processus d'exploitation.	187
Figure 7-10 : Architecture du processus de génération de séquences d'activités personnalisées.	188
Figure 8-1 : Fonctionnement du logiciel Adapte.	201
Figure 8-2 : Écran d'Adapte permettant de définir les propriétés de la description pédagogique du modèle OKEP/x d'un EIAH x.	202
Figure 8-3 : Écran d'Adapte permettant de définir un paramètre du modèle OKEP/x d'un EIAH x.	203
Figure 8-4 : Écran d'Adapte permettant de définir les règles de la description pédagogique du modèle OKEP/x d'un EIAH x.	204
Figure 8-5 : Écran de choix d'un patron d'exercices dans Adapte.	206
Figure 8-6 : Écran d'Adapte permettant de définir des contraintes sur le patron opérationnel « Table de conjugaison » du générateur « Tableau à double entrée ».	207
Figure 8-7 : Écran d'Adapte permettant de définir des contraintes sur une activité logicielle... ..	208
Figure 8-8 : Écran d'Adapte permettant de définir des contraintes sur profils portant sur une valeur numérique.	209
Figure 8-9 : Écran d'Adapte permettant de définir des contraintes sur profils portant sur une valeur textuelle.	210
Figure 8-10 : Écran d'Adapte permettant de définir des contraintes sur profils portant sur un élément.	211
Figure 8-11 : Écran d'Adapte permettant de définir des règles d'affectation des activités aux apprenants.	212
Figure 8-12 : Écran d'Adapte permettant de définir une stratégie pédagogique.	213

Figure 8-13 : Écran d'Adapte permettant de définir un contexte d'utilisation pour une stratégie pédagogique donnée.....	214
Figure 8-14 : Séquence personnalisée d'activités papier générée par Adapte.....	215
Figure 8-15 : Séquences personnalisées d'activités logicielles proposées par Adapte composées d'une liste d'instructions pour l'apprenant et d'un fichier de configuration pour ABALECT.	216
Figure 8-16 : Écran de validation des séquences de travail personnalisées dans Adapte.....	216
Figure 8-17 : Architecture du logiciel Adapte.	217
Figure 8-18 : Métadonnées d'une structure d'activités.....	219
Figure 8-19 : Métadonnées d'une stratégie pédagogique.....	220
Figure 8-20 : Métadonnées d'un contexte d'utilisation.....	221
Figure 8-21 : Lien entre l'architecture d'Adapte et nos contributions théoriques.....	222
Figure 8-22 : Utilisation d'Adapte par l'enseignant avec des profils d'apprenants.....	223
Figure 8-23 : Utilisation d'Adapte en boucle automatique à partir des profils d'apprenants.....	224
Figure 8-24 : Utilisation d'Adapte par l'enseignant sans profils d'apprenants.....	225
Figure 9-1 : Extrait des résultats du questionnaire sur les pratiques éducatives diffusé en juin 2009.....	241
Figure A-1 : Patron d'exercices « Travail sur illustration ».....	307
Figure A-2 : Patron opérationnel « Annotation d'illustration ».....	307
Figure A-3 : Choix d'un mot dans un patron d'exercices.	308
Figure A-4 : Structure d'exercices issue du patron opérationnel « Annotation d'illustration » associé au patron « Travail sur illustration ».....	309
Figure A-5 : Structure d'exercices issue du patron opérationnel « Annotation d'illustration » associé au patron « Travail sur illustration ».....	310
Figure A-6 : Patron opérationnel « Copie d'un modèle ».....	310
Figure A-7 : Choix d'une lettre dans un patron d'exercices.....	311
Figure A-8 : Structure d'exercices issue du patron opérationnel « Copie d'un modèle » associé au patron « Travail sur illustration ».....	312
Figure A-9 : Patron d'exercices « Organisation d'éléments ».....	312
Figure A-10 : Patron opérationnel « Classement d'objets ».....	312
Figure A-11 : Sélection d'un objet de type <i>Text</i> dans le patron opérationnel « Classement d'objets ».....	313
Figure A-12 : Sélection d'un objet de type <i>Word</i> dans le patron opérationnel « Classement d'objets ».....	313
Figure A-13 : Sélection d'un objet de type <i>Letter</i> dans le patron opérationnel « Classement d'objets ».....	314
Figure A-14 : Sélection d'un objet de type <i>Picture</i> dans le patron opérationnel « Classement d'objets ».....	314
Figure A-15 : Sélection d'un objet de type <i>Number</i> dans le patron opérationnel « Classement d'objets ».....	314
Figure A-16 : Choix d'un nombre dans un patron d'exercices.	315
Figure A-17 : Structure d'exercices issue du patron opérationnel « Classement d'objets » associé au patron « Organisation d'éléments ».....	315
Figure A-18 : Structure d'exercices issue du patron opérationnel « Classement d'objets » associé au patron « Organisation d'éléments ».....	316
Figure A-19 : Patron opérationnel « Groupement d'objets ».....	317
Figure A-20 : Critères possibles pour associer des textes dans le patron opérationnel « Groupement d'objets ».....	317
Figure A-21 : Critères possibles pour associer des mots dans le patron opérationnel « Groupement d'objets ».....	317
Figure A-22 : Critères possibles pour associer des lettres dans le patron opérationnel « Groupement d'objets ».....	318

Figure A-23 : Critères possibles pour associer des illustrations dans le patron opérationnel « Groupement d'objets ».....	318
Figure A-24 : Critères possibles pour associer des nombres dans le patron opérationnel « Groupement d'objets ».....	318
Figure A-25 : Structure d'exercices issue du patron opérationnel « Groupement d'objets » associé au patron « Organisation d'éléments ».....	319
Figure A-26 : Patron d'exercices « Tableau à double entrée ».....	320
Figure A-27 : Patron opérationnel « Table de conjugaison ».....	320
Figure A-28 : Structure d'exercices issue du patron opérationnel « Table de conjugaison » associé au patron « Tableau à double entrée ».....	320
Figure A-29: Patron opérationnel « Table de mathématiques ».....	321
Figure A-30 : Structure d'exercices issue du patron opérationnel « Table de mathématiques » associé au patron « Tableau à double entrée ».....	322
Figure A-31: Patron opérationnel « Tableaux à double entrées ».....	322
Figure A-32 : Définition de la taille d'un tableau dans le patron opérationnel « Autre tableau à double entrée ».....	322
Figure A-33 : Définition des titres des colonnes d'un tableau dans le patron opérationnel « Autre tableau à double entrée ».....	323
Figure A-34 : Définition d'une expression mathématique dans le patron opérationnel « Autre tableau à double entrée ».....	323
Figure A-35 : Définition des titres des lignes d'un tableau dans le patron opérationnel « Autre tableau à double entrée ».....	324
Figure A-36 : Définition des cellules d'un tableau dans le patron opérationnel « Autre tableau à double entrée ».....	324
Figure A-37 : Élément <i>ToDelete</i> du patron opérationnel « Autre tableau à double entrée ».....	324
Figure A-38 : Structure d'exercices issue du patron opérationnel « Autre tableau à double entrée » associé au patron « Tableau à double entrée ».....	325
Figure A-39 : Structure d'exercices issue du patron opérationnel « Autre tableau à double entrée » associé au patron « Tableau à double entrée ».....	326
Figure A-40 : Structure d'exercices issue du patron opérationnel « Autre tableau à double entrée » associé au patron « Tableau à double entrée ».....	327
Figure A-41 : Représentation de la totalité du tableau décrit dans la structure d'exercices de la Figure A-40.....	327
Figure A-42 : Patron d'exercices « Questions / réponses ».....	328
Figure A-43 : Élément <i>Question</i> du patron d'exercices « Questions / réponses ».....	329
Figure A-44 : Élément <i>Answers</i> du patron d'exercices « Questions / réponses ».....	329
Figure A-45 : Structure d'exercices issue du patron « Questions / réponses ».....	330
Figure A-46 : Structure d'exercices issue du patron « Questions / réponses ».....	330
Figure A-47 : Structure d'exercices issue du patron « Questions / réponses ».....	331
Figure A-48 : Architecture du générateur « Travail sur illustration ».....	333
Figure A-49 : Architecture du générateur « Organisation d'éléments ».....	335
Figure A-50 : Architecture du générateur « Tableau à double entrée ».....	337
Figure A-51 : Architecture du générateur « Expression algébrique ».....	339
Figure A-52 : Architecture du générateur « Problème scientifique».....	341
Figure A-53 : Architecture du générateur « Questions / réponses ».....	343
Figure A-54 : Extrait du fichier de configuration pour la génération d'exercices dans l'EIAH AMBRE-ADD.....	345
Figure A-55 : Extrait du fichier de configuration pour l'interface du logiciel EXOMATIKS.....	346
Figure A-56 : Technologies utilisées dans le processus AKASI.....	361
Figure A-57 : Technologies utilisées dans le processus OPIKSI.....	362

TABLES DES ÉQUATIONS

Équation 4-1 : Modèle de la personnalisation souhaité par un agent donné.....	84
Équation 4-2 : Modèle d'une stratégie pédagogique.....	84
Équation 4-3 : Modèle d'une règle d'affectation.....	85
Équation 4-4 : Équivalence du modèle d'une règle d'affectation.....	85
Équation 4-5 : Modèle d'un contexte d'utilisation.....	85
Équation 4-6 : Modèle des contraintes de création d'une séquence d'activités.....	85
Équation 4-7 : Modèle du support d'une séquence d'activités.....	86
Équation 4-8 : Modèle des contraintes de présentation d'une personnalisation.....	87
Équation 5-1 : Modèle d'une contrainte sur profils portant sur une valeur textuelle.....	103
Équation 5-2 : Modèle d'une contrainte sur profils portant sur une valeur numérique.....	103
Équation 5-3 : Modèle d'une contrainte sur profils portant sur un entier positif.....	103
Équation 5-4 : Modèle d'une contrainte sur profils portant sur un élément.....	107
Équation 5-5 : Modèle d'une contrainte sur profils portant sur un nombre d'occurrences.....	109
Équation 9-1 : Nombre maximum de découpes possibles dans une liste de mots.....	314
Equation A-1 : Formule permettant convertir une valeur donnée dans une échelle numérique vers une autre échelle numérique.....	289
Equation A-2 : Formule permettant convertir une valeur donnée dans une échelle textuelle vers une échelle numérique.....	289

TABLES DES EXEMPLES

Exemple A-1 : Exercice à trous avec liste des mots à placer [Claudal 2007].....	291
Exemple A-2 : Exercice à trous sans liste des mots à placer, issu du logiciel Adapte	291
Exemple A-3 : Exercice à trous avec pour chaque trou un indice sur le mot à placer [Camus 2007].....	292
Exemple A-4 : Exercice à trous sans liste de mots à placer, issu des pratiques d'enseignants...	292
Exemple A-5 : Exercice d'identification de noms dans un texte [Madoré 2007].....	293
Exemple A-6 : Exercice d'identification des éléments pertinents d'un texte [Madoré 2007].....	293
Exemple A-7 : Exercice d'analyse grammaticale de texte [Madoré 2007].....	293
Exemple A-8 : Exercice d'analyse grammaticale de texte [Madoré 2007].....	293
Exemple A-9 : Exercice de traduction, issu du logiciel Adapte.....	294
Exemple A-10 : Exercice de traduction, issu du logiciel Adapte.....	294
Exemple A-11 : Exercice de définition, issu du logiciel Adapte.....	294
Exemple A-12 : Exercice de définition, issu du logiciel Adapte.....	294
Exemple A-13 : Exercice de correction orthographique [Madoré 2007].....	295
Exemple A-14 : Exercice de transposition au pluriel [Madoré 2007].....	295
Exemple A-15 : Exercice de changement de temps [UniversDesExperts 2003].....	295
Exemple A-16 : Exercice de changement de temps [Enseignons.be 2004].....	295
Exemple A-17 : Exercice d'annotation d'illustration [ConnaisTuLAnatomie 2002].....	295
Exemple A-18 : Exercice d'annotation d'illustration [Dromer 2009].....	296
Exemple A-19 : Exercice de symétrie, issu des pratiques d'enseignants.....	297
Exemple A-20 : Exercice de symétrie, issu des pratiques d'enseignants.....	297
Exemple A-21 : Exercice d'écriture, issu des pratiques d'enseignants.....	297
Exemple A-22 : Exercice d'écriture, issu des pratiques d'enseignants.....	297
Exemple A-23 : Exercice de classement de nombres, issu des pratiques d'enseignants.....	298
Exemple A-24 : Exercice de texte en désordre, issu des pratiques d'enseignants.....	298
Exemple A-25 : Exercice de texte en désordre [Camus 2007].....	298
Exemple A-26 : Exercice de texte en désordre [Camus 2007].....	298
Exemple A-27 : Exercice d'association de forme, issu des pratiques d'enseignants.....	298
Exemple A-28 : Exercice d'association des images à des mots [Ortholud 2004].....	299
Exemple A-29 : Exercice de détection de familles de mots [Madoré 2007].....	299
Exemple A-30 : Exercice de conjugaison française, issu des pratiques d'enseignants.....	299
Exemple A-31 : Exercice de conjugaison française [Madoré 2007].....	299
Exemple A-32 : Exercice de conjugaison française, issu des pratiques d'enseignants.....	300
Exemple A-33 : Exercice de table de multiplication, issu des pratiques d'enseignants.....	300
Exemple A-34 : Exercice de complétion d'un tableau à double entrée [IUFM_Paris 2005].....	300
Exemple A-35 : Exercice de complétion d'un tableau à double entrée [Jobin 2007].....	301
Exemple A-36 : Exercice de complétion d'un tableau à double entrée [Madoré 2007].....	301
Exemple A-37 : Exercice d'utilisation d'un tableau à double entrée, issu des pratiques d'enseignants.....	301
Exemple A-38 Exercice de complétion d'un tableau à double entrée, issu des pratiques d'enseignants.....	301
Exemple A-39 : Exercice portant sur une expression algébrique, des pratiques d'enseignants.	302
Exemple A-40 : Exercice portant sur une expression algébrique, des pratiques d'enseignants.	302
Exemple A-41 : Exercice portant sur une expression algébrique, des pratiques d'enseignants.	302
Exemple A-42 : Exercice portant sur une expression algébrique, des pratiques d'enseignants.	302
Exemple A-43 : Exercice portant sur un problème additif [Duclosson et al. 2005a].....	302

Exemple A-44 Exercice portant sur un problème de dénombrement [Guin 1997].	302
Exemple A-45 Exercice portant sur un problème de dénombrement [Guin 1997].	302
Exemple A-46 Exercice portant sur un problème de chimie [Guin 1997].....	302
Exemple A-47 : Exercice de démonstration [Sesamath 2002].....	303
Exemple A-48 : QCM, issu des pratiques enseignantes.	303
Exemple A-49 : QCM [Ortholud 2004].....	303
Exemple A-50 : Extrait d'un QCM [Dromer 2009].....	303
Exemple A-51 : QROC, issu des pratiques enseignantes.....	304

INTRODUCTION

DOMAINE DE RECHERCHE

Cette thèse se situe dans le domaine des Environnements Informatiques pour l'Apprentissage Humain (EIAH). Ce domaine de recherche vise à fournir des modèles et des outils informatiques pour accompagner l'apprentissage et l'enseignement. Le champ des EIAH est un domaine pluridisciplinaire à la croisée de l'informatique, de la psychologie cognitive, de la didactique et des sciences de l'éducation. Dans notre travail, nous étudions plus particulièrement la personnalisation de l'apprentissage d'un point de vue informatique.

La personnalisation de l'apprentissage consiste à adapter aux spécificités de chaque apprenant le contenu, mais aussi la manière de proposer un apprentissage. Cet apprentissage peut se faire dans le contexte classique de la classe ou à distance, en autonomie ou sous la supervision d'un tuteur. Dans tous les cas, le support des activités d'apprentissage peut être soit des feuilles de papier, soit des logiciels informatiques.

PROBLÉMATIQUE

La personnalisation de l'apprentissage consiste à modifier les activités que l'on propose à un apprenant, activités papier ou activités sur un logiciel pédagogique, en fonction d'une situation pédagogique donnée. Cette situation pédagogique comprend les spécificités de l'apprenant, les buts pédagogiques de l'enseignant et le contexte dans lequel se déroule l'apprentissage (en classe, en autonomie, avec ou sans logiciel pédagogique, etc.) [Touchnikine 2009, p. 32]. Nous considérons donc que personnaliser l'apprentissage revient à appliquer des règles dont la condition comporte des contraintes sur la situation pédagogique et la conclusion indique les modifications à produire sur les activités.

Cette personnalisation est une tâche complexe et longue à mettre en œuvre d'une part parce qu'il existe une grande variété d'activités papier, ainsi qu'une grande variété des logiciels pédagogiques portant sur des sujets d'études différents, et d'autre part car les situations pédagogiques sont très diverses.

Dans le cadre de cette thèse, nous avons cherché à identifier un processus qui permette à la fois de personnaliser des séances de travail sur papier et des séances de travail sur des logiciels pédagogiques. Ce processus doit permettre de prendre en compte les spécificités de chaque apprenant, en s'appuyant sur son profil, mais il doit également prendre en compte les buts et les habitudes pédagogiques des enseignants.

Notre problématique est donc la suivante :

QUEL PROCESSUS GÉNÉRIQUE PEUT PERMETTRE DE PERSONNALISER LES ACTIVITÉS PÉDAGOGIQUES PROPOSÉES AUX APPRENANTS, À PARTIR DE LEURS PROFILS, TOUT EN ÉTANT GUIDÉ PAR LES SPÉCIFICITÉS DU CONTEXTE PÉDAGOGIQUE ?

Cette problématique peut être décomposée en trois points, correspondant aux trois facettes de la création des règles utilisées lors de la personnalisation.

Tout d'abord, **comment exploiter les profils d'apprenants pour prendre en compte les individualités des apprenants ?** Un profil d'apprenant rassemble des indications sur l'apprenant sous la forme d'un ensemble d'informations caractérisant ses connaissances, compétences et/ou conceptions. Le profil peut être généré par un système (un EIAH disposant d'un module élaborant le profil de l'apprenant) ou conçu et utilisé directement sous forme papier-crayon par un enseignant [Jean-Daubias 2003]. La question est donc de savoir comment sélectionner dans les profils les informations utiles à la personnalisation. Du point de vue des règles de personnalisation, la question qui se pose est alors : comment peut-on utiliser des profils d'apprenants pour créer les conditions contenues dans ces règles ?

Ensuite, **comment adapter une activité pour prendre en compte les besoins et habitudes pédagogiques d'un enseignant ?** Cette question concerne à la fois l'adaptation des activités papier et des activités logicielles. Pour les activités papier, la question est de savoir comment générer un exercice correspondant au besoin d'un enseignant. Pour les activités logicielles, il faut savoir comment paramétrer un logiciel pédagogique pour que son contenu, ainsi que l'environnement qu'il propose correspondent aux attentes de l'enseignant. Du point de vue des règles de personnalisation, la question soulevée est alors : comment peut-on définir les conclusions contenues dans ces règles ?

Enfin, **comment attribuer une activité à un apprenant ?** C'est-à-dire, qui possède les connaissances nécessaires pour faire le choix de faire travailler un apprenant donné sur une activité donnée ? Quelles sont ces connaissances en jeu et comment sont-elles utilisées ? Du point de vue des règles de personnalisation, la question qui se pose est alors : qui associe, et comment, une condition et une conclusion pour former un règle ?

Ces trois facettes doivent être traitées dans l'objectif de proposer un processus générique. Nous nous posons donc comme contrainte de proposer un processus qui puisse être implémenté dans un environnement informatique indépendant des logiciels pédagogiques proposant des activités. Ce processus, et les modèles qui lui sont rattachés, doivent permettre une personnalisation externalisée des activités pédagogiques.

CONTRIBUTION

Pour répondre au premier point de notre problématique, nous proposons le modèle cPMDL. Cette extension du langage de modélisation des profils PMDL permet de contraindre les profils des apprenants afin de sélectionner ceux ayant les caractéristiques requises dans les contraintes. cPMDL nous permet donc d'exploiter les informations contenues dans les profils dans le processus de personnalisation.

Pour répondre au second point, nous proposons l'approche GEPPETO. Cette approche s'appuie sur des modèles et des processus génériques permettant d'adapter les activités en fonction des intentions pédagogiques des enseignants grâce à la définition de contraintes sur activités. Nous avons décliné cette approche pour permettre l'adaptation des activités papier (GEPPETO_P) ainsi que pour l'adaptation d'activités logicielles au sein d'EIAH (GEPPETO_S).

Pour répondre au troisième point, nous proposons le modèle PERSUA2 qui permet de lier les contraintes sur profils de cPMDL aux contraintes sur activités de GEPPETO. Ces liens, nommés règles d'affectation, sont ensuite hiérarchisés selon leur degré de priorité pour former une

stratégie pédagogique qui sera associée à un ou plusieurs contextes d'utilisation. Le modèle PERSUA2 reflète ainsi les objectifs pédagogiques d'un enseignant. Son exploitation permet de créer des séquences de travail propres à chaque apprenant.

CADRE DU TRAVAIL

Cette thèse se situe dans le cadre du projet PERLEA (Profils d'Élèves Réutilisés pour L'Enseignant et l'Apprenant) [Jean-Daubias 2003] qui vise à proposer d'une part des modèles pour la réutilisation des profils d'apprenants par les différents acteurs de la situation d'apprentissage (enseignant, apprenant, famille, institution et chercheur) et d'autre part un environnement informatique à destination des enseignants mettant en œuvre ces modèles.

Cet environnement informatique, nommé EPROFILEA, est constitué de deux parties : la préparation des profils et les différentes exploitations pédagogiques de ces profils. L'architecture de cet environnement est présentée sur la figure ci-dessous.

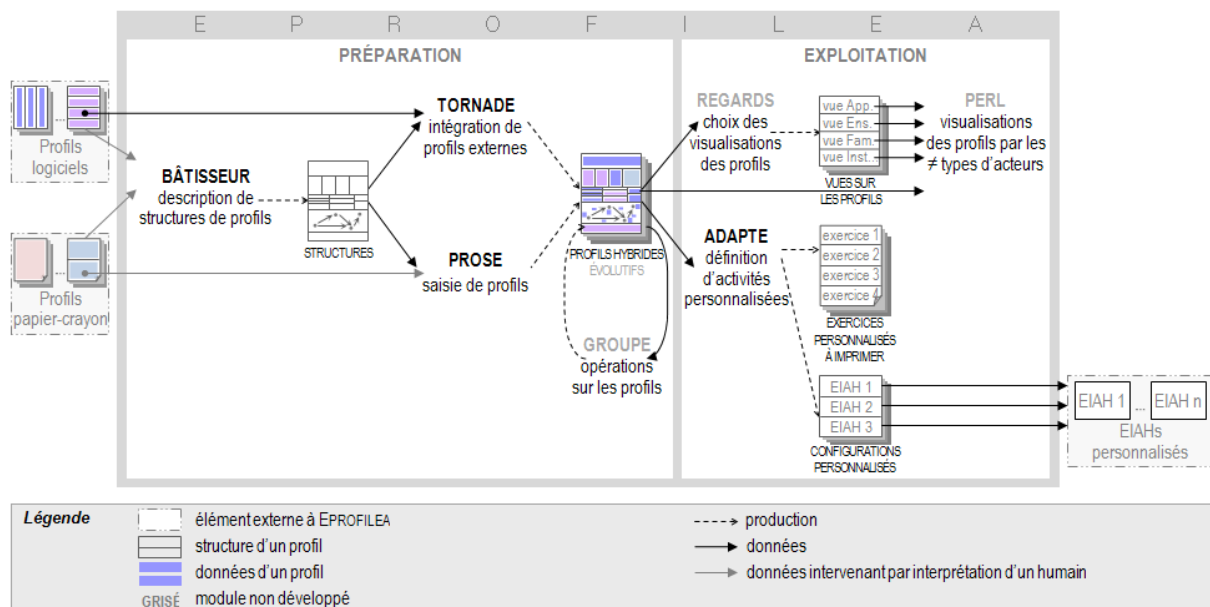


Figure 0-1 : Architecture de l'environnement EPROFILEA.

La phase de préparation des profils permet de prendre en compte toute la diversité des profils d'apprenants. Ceci est primordial car les profils d'apprenants, créés par des systèmes informatiques ou par les enseignants, peuvent être établis à la demande de différentes personnes et contenir des informations très variées. Pour un même apprenant, il peut exister plusieurs profils parfois complémentaires, parfois contradictoires, dans une ou plusieurs disciplines. Ces profils sont très divers, tant par leur contenu que par leur structure. Cette hétérogénéité des profils rend difficile leur utilisation hors du contexte pour lequel ils ont été créés. Afin de traiter cette diversité des profils, la phase de préparation s'appuie sur un travail théorique qui a abouti à la proposition de modèles génériques permettant la réutilisation de profils d'apprenants [Eyssautier-Bavay 2008]. La préparation des profils consiste ainsi à établir la structure des profils que l'on souhaite manipuler, avant d'y intégrer les données issues des profils externes pour constituer des profils d'apprenants conformes au souhait de l'enseignant et respectant le formalisme d'EPROFILEA.

La description de la structure des profils que l'enseignant souhaite exploiter dans l'environnement EPROFILEA est élaborée par l'enseignant dans le module Bâtitseur, selon une

opérationnalisation du langage de modélisation de profils PMDL. Grâce à ce langage, Bâtitseur permet à l'enseignant de décrire la structure des profils préexistants, qu'ils soient papier-crayon ou issus d'EIAH, quels que soient les types d'informations qu'ils contiennent [Jean-Daubias et al. 2009a]. Compléter la structure de profils créée dans Bâtitseur avec les données personnelles des apprenants pour constituer les profils EPROFILEA se fait de façons différentes selon que les données sont issues de profils papier-crayon ou d'un EIAH. Dans le cas de profils papier-crayon, EPROFILEA comporte un module, Prose, permettant à l'enseignant de saisir les données de chacun de ses apprenants selon la structure de profils définie dans Bâtitseur. Dans le cas de profils issus de logiciels, EPROFILEA propose des systèmes de conversion de profils, interfaces entre les logiciels externes et EPROFILEA, ainsi qu'un module, Tornade, assistant un enseignant-expert dans la constitution de tels adapteurs. Les profils ainsi créés peuvent être hybrides, c'est-à-dire qu'une structure de profils peut être instanciée à la fois dans Prose avec des données papier-crayon et dans Tornade avec des données provenant de profils informatisés.

La phase d'exploitation pédagogique des profils donne les moyens aux enseignants de réintégrer leur travail sur les profils d'apprenants avec l'environnement EPROFILEA, dans leur pratique de classe. En effet, la phase d'exploitation permet de proposer aux différents acteurs de la situation d'apprentissage des activités sur les profils et des activités pédagogiques définies en fonction du contenu des profils.

À partir des profils résultants de la phase de préparation, l'enseignant peut établir dans le module Regards différentes vues d'un même profil, adaptées à chaque acteur de la situation d'apprentissage. Pour construire ces vues, l'enseignant choisit les parties du profil qui seront consultables par l'acteur concerné, le vocabulaire utilisé ou encore le mode de présentation des informations à l'interface. Le module Perl permet la visualisation interactive des profils par les différents acteurs selon les vues déterminées par l'enseignant dans Regards. Il propose également aux apprenants des activités sur leurs profils (reformulation, pose d'objectifs, etc.) permettant à l'apprenant d'entrer dans une démarche réflexive par rapport à son apprentissage et ainsi, de mieux assimiler et exploiter les informations qui lui sont fournies. Enfin, le module Adapte permet de proposer aux apprenants des activités adaptées à leur profil : activités papier ou activités logicielles gérées par un EIAH externe à l'environnement [Lefevre et al. 2009a]. Ces activités pourront servir en remédiation ou pour acquérir de nouvelles compétences. C'est dans ce module que les résultats théoriques proposés lors de cette thèse ont été mis en œuvre.

PARTENARIAT AVEC DES ENSEIGNANTS

Cette thèse s'intéresse à la personnalisation des activités pédagogiques fournies aux apprenants, personnalisation correspondant aux besoins et aux habitudes pédagogiques des enseignants. Pour avoir un aperçu des pratiques des enseignants, nous avons travaillé avec trois d'entre eux. Cette collaboration s'étant faite dans le cadre du projet PERLEA, dans la suite de ce manuscrit, nous nommons ces enseignants « les enseignants partenaires du projet PERLEA ». Cette collaboration a eu lieu en deux temps.

Dans un premier temps, nous avons travaillé avec deux enseignants de l'enseignement primaire, Agnès et Patrick, afin de recueillir des informations sur leurs pratiques. Nous avons travaillé ensemble pour identifier les mécanismes implicites qu'ils utilisent pour personnaliser les activités pédagogiques. Nous avons également discuté des données utiles à cette personnalisation : profils d'apprenants, résultats d'évaluations, jugements personnels, etc. Nous avons enfin collecté les activités pédagogiques au format papier qu'eux ou leurs collègues proposent à leurs élèves. C'est à partir de ce travail préliminaire que nous avons construit notre modèle de personnalisation et notre typologie des exercices papier. Nous leur avons ensuite

présenté un prototype mettant en œuvre nos propositions pour la personnalisation des activités papier. Cette présentation avait pour but de voir si le logiciel correspondait à leurs habitudes de travail et quelle aide il pouvait leur apporter. Nous avons de plus travaillé sur l'interface de celui-ci et sur la typologie d'exercices. Nous avons ainsi pu corriger nos modèles et le prototype associé. Nous avons fait cinq réunions communes avec Agnès et Patrick et une réunion supplémentaire avec Patrick. Ces réunions ont eu lieu en 2006 et ont chacune duré une demi-journée.

Dans un deuxième temps, nous avons travaillé avec une enseignante de collège, Bérengère, dans le but de tester nos propositions, ainsi que le logiciel Adapte qui les met en œuvre. Cette collaboration nous a permis d'avoir un regard extérieur puisque Bérengère n'est jamais intervenue dans la phase de conception d'Adapte. Nous avons fait six séances de travail de trois jours environ entre 2007 et 2009. Durant ces séances, Bérengère a utilisé le logiciel Adapte, mais aussi tous les modules de l'environnement EPROFILEA dans lequel il s'intègre. Elle a pour cela utilisé ses propres profils d'apprenants (surtout des relevés de notes et des commentaires) et défini ses propres règles de personnalisation, tout d'abord sans le logiciel puis avec le logiciel. De plus, nous avons pu récolter par son intermédiaire les activités pédagogiques que ses collègues utilisent afin de les confronter à notre typologie d'exercices papier. Bérengère a ensuite diffusé auprès de ses collègues un questionnaire dans lequel nous cherchions, entre autres, à connaître les pratiques de ces enseignants de collège en termes d'utilisation de logiciels éducatifs en classe. Le faible nombre de réponses à ce questionnaire nous a seulement permis de lister les logiciels utilisés par ces enseignants.

Pour compléter ces collaborations, nous avons diffusé un questionnaire auprès d'enseignants de tous niveaux (maternelle, primaire, collège, lycée et université) et de diverses disciplines, pour connaître leurs besoins et leurs habitudes en termes de personnalisation d'activités pédagogiques, d'utilisation de profils d'apprenants et d'utilisation de logiciels pédagogiques. Nous reviendrons sur les résultats de ce questionnaire dans le chapitre consacré à l'évaluation de nos propositions.

SCÉNARIOS D'USAGE

Nous présentons dans cette section six scénarios montrant les enjeux de la personnalisation des activités pédagogiques dans les pratiques des enseignants. Nous reviendrons sur ces scénarios à la fin de notre état de l'art pour montrer leurs limites actuelles (cf. section 3.1), puis lors de l'évaluation de nos propositions pour montrer en quoi elles peuvent changer ces situations (cf. section 8.5).

SCÉNARIO 1 : AIDE PERSONNALISÉE À L'ÉCOLE PRIMAIRE

Alphonse, enseignant de CE2, imprime en début d'année les bilans personnels de ses élèves proposés par le logiciel J'ADE [J'ADE 2007a]. Ces bilans correspondent aux résultats des évaluations nationales en français et en mathématiques que ses élèves ont passées en fin d'année de CE1, à la demande de l'institution scolaire. Il fait également régulièrement utiliser le logiciel AMBRE-ADD [Guin-Duclosson et al. 2002] à ses élèves pour la résolution de problèmes additifs en autonomie lorsqu'ils ont terminé leur travail.

Dans l'établissement où enseigne Alphonse, une aide personnalisée est proposée aux élèves en difficulté. Pour cela, les enseignants identifient les élèves concernés et forment des groupes de six élèves. Chaque élève bénéficiera d'une heure de soutien en plus des heures de cours, dispensée par un autre enseignant que le sien et d'une heure de soutien pendant les heures de

cours. Pour préparer le contenu de ces séances de soutien, les enseignants se rencontrent une heure par semaine. Ainsi, chaque enseignant fixe les objectifs à atteindre par ses élèves et fournit à ses collègues la liste des exercices à faire contenant des exercices tirés des livres d'exercices qu'il possède.

Alphonse souhaiterait s'appuyer sur toutes les informations dont il dispose sur les connaissances de ses élèves (données issues des évaluations nationales et profils d'élèves créés par le logiciel AMBRE-ADD) pour proposer des séances de remédiation soit sous forme papier, soit avec des logiciels d'apprentissage adaptés à chacun des élèves concernés par les séances de soutien.

SCÉNARIO 2 : COMBINER REMÉDIATION ET APPROFONDISSEMENT

Fred, professeur d'anglais au collège, ne fait actuellement ni remédiation, ni approfondissement avec ses classes de trente élèves.

Il souhaiterait, avec chacune de ses classes, faire une séance d'une heure en salle informatique durant laquelle il séparerait la classe en deux groupes : les élèves ayant des difficultés dans le premier et les autres dans le second. La salle informatique de son établissement contient d'un côté des ordinateurs et de l'autre des tables avec un tableau.

Pendant la première demi-heure de chaque séance, le groupe d'élèves sans difficulté apprendrait du nouveau vocabulaire grâce aux logiciels PLANET ENGLISH [English] et VOCABONE [VocabOne 2007]. De son côté, il ferait travailler le groupe d'élèves en difficulté sur des points non compris précédemment grâce à des exercices papier et des discussions orales.

Durant la seconde demi-heure, il échangerait les groupes. Les élèves en difficulté passeraient sur ordinateur pour faire des activités adaptées à leurs difficultés de prononciation sur le logiciel SEPHONICS [Sephonics 2008], tandis que lui travaillerait avec les autres élèves pour approfondir des notions vues en cours.

Dans tous les cas, il souhaiterait que les logiciels s'adaptent automatiquement aux besoins des élèves.

SCÉNARIO 3 : FORMATION À DISTANCE PERSONNALISÉE

Amélie, professeur de français au collège, donne des cours à distance, via une plateforme informatique, pour des élèves ne pouvant être scolarisés. Elle accompagne ainsi une vingtaine d'élèves dans leur apprentissage du français.

Actuellement, lorsqu'elle donne des exercices à ses élèves, elle fournit les mêmes exercices à chacun de manière à pouvoir faire une correction commune lors de séances de travail synchrones avec tous ses élèves. De plus, elle leur demande chaque semaine de faire une rédaction pour laquelle elle fournit une correction personnalisée.

Elle souhaiterait maintenant construire un profil pour chaque élève avec les difficultés identifiées lors de la correction des rédactions. Elle se servirait de ces profils pour faire travailler les élèves sur des exercices ayant comme support leur propre rédaction, mais avec des énoncés adaptés à leur profil. Par exemple, leur demander à tous de réécrire leur rédaction, mais en mettant l'accent pour l'un sur la ponctuation, pour l'autre sur le passé simple, etc.

SCÉNARIO 4 : APPRENDRE À APPRENDRE

Manuela, professeur de biologie au collège, participe à l'encadrement de l'étude du soir pour les élèves de 6^{ème} et de 5^{ème}. Le but de cette étude est de donner à tous les élèves des outils méthodologiques. Pour cela, les élèves reprennent des exercices déjà faits dans des disciplines variées et les refont avec les conseils des enseignants présents.

Ce soutien serait plus efficace si les élèves ne refaisaient pas exactement le même exercice, mais un exercice similaire. Cela permettrait de leur donner des conseils méthodologiques qu'ils pourraient appliquer à un exercice dont ils ne connaissent pas déjà la solution. Actuellement, Manuela ne peut pas mettre en place cette solution idéale, car elle n'a pas les connaissances nécessaires pour créer des exercices de mathématiques, de français, d'anglais ou encore d'histoire.

SCÉNARIO 5 : GROUPE DE RÉVISION POUR LE BREVET DES COLLÈGES

Louise et Léon sont professeurs de français au collège. Ils possèdent un profil pour chacun de leurs élèves. Ces profils, sous forme papier, contiennent la connaissance qu'ils ont de leurs élèves et les résultats aux évaluations de chacun d'entre eux.

Dans leur établissement, les élèves de 3^{ème} participent, sur la base du volontariat, à des séances de révisions pour le brevet des collèges. Il y a trois séances d'une heure par semaine : une pour réviser le français, une pour les mathématiques et une pour l'histoire-géographie. En début de semaine, les professeurs de chaque matière proposent aux élèves la liste des points pouvant être abordés dans la prochaine séance. Chaque élève peut s'inscrire en précisant les points qui l'intéressent. À partir de ces inscriptions, deux groupes sont formés par matière pour travailler chacun sur un thème.

Louise et Léon encadrent chacun un groupe de révision pour le français. Ils souhaiteraient obtenir le profil de leur groupe, constitué à partir des profils individuels des élèves, afin de créer une feuille de travail commune pour un groupe, mais adaptée au profil du groupe.

De la même manière, leurs collègues de mathématiques, qui possèdent des profils individuels sur les connaissances en mathématiques de leurs élèves, souhaiteraient également obtenir des profils de groupes. Ils se serviraient de ces profils pour adapter le logiciel MATHENPOCHE [Mathenpoche 2002] au besoin de chaque groupe d'élèves.

SCÉNARIO 6 : PLAN LICENCE À L'UNIVERSITÉ

Lucie est maître de conférences en informatique. Dans le cadre du plan licence à l'université, elle organise trois séances de soutien d'une heure et demi durant chaque semestre et doit faire passer un examen, sous forme de QROC (Questionnaire à Réponses Ouvertes et Courtes), à ses étudiants lors de chaque séance de travaux dirigés.

Actuellement, lors des séances de soutien, elle aide les étudiants à refaire les exercices non compris précédemment en travaux dirigés. Elle répond de plus aux questions que peuvent se poser les étudiants à l'issue du cours. Il serait également possible de faire, lors de ces séances de soutien, des exercices similaires à ceux des travaux dirigés.

De plus, pour les QROC qu'elle doit faire passer à la fin des travaux dirigés, elle crée leurs énoncés elle-même. Pour gagner du temps, elle souhaiterait posséder un logiciel lui fournissant des propositions d'énoncés.

Ces différents scénarios montrent une grande variété de besoins non pourvus par les outils actuellement à la disposition des enseignants. Ils montrent de plus la lourdeur du travail à faire si un enseignant souhaite créer des séances adaptées aux informations qu'il a sur ses apprenants : celui-ci doit d'une part traiter lui-même ces informations et d'autre part, apprendre à manipuler différents outils (générateurs et/ou interfaces de paramétrage) pour obtenir des ressources adaptées à ses élèves et à ses buts pédagogiques.

PLAN DE LA THÈSE

La première partie de la thèse est consacrée à l'état de l'art relatif à notre problématique de recherche. Ainsi, nous avons étudié les différentes approches existantes pour personnaliser l'apprentissage, tant au niveau de la prise en compte des apprenants, que de la prise en compte des enseignants. La personnalisation de l'apprentissage telle que nous la concevons doit permettre de personnaliser les activités papier proposées aux apprenants, mais aussi d'adapter les sessions des logiciels pédagogiques qu'ils utilisent. Nous avons donc complété notre état de l'art en étudiant les différentes manières de piloter un système informatique et plus particulièrement un EIAH. Nous terminons cet état de l'art en reprenant chacun des scénarios d'usage présentés en introduction pour montrer les limites de leur mise en œuvre avec les outils actuellement à disposition des enseignants.

La seconde partie présente les différentes contributions théoriques que nous avons faites. Ces propositions se déclinent en quatre axes. Tout d'abord, nous proposons le modèle PERSUA2 permettant aux enseignants de décrire comment ils souhaitent affecter une activité à un apprenant en fonction de son profil. Ensuite, nous expliquons comment un enseignant peut contraindre un profil pour sélectionner les éléments qui lui sont utiles lors de la personnalisation. Nous nous appuyons pour cela sur le langage de description de profils PMDL [Eyssautier-Bavay 2008, Jean-Daubias et al. 2009a]. Par la suite, nous exposons notre approche GEPPETO permettant d'adapter les activités des apprenants aux buts pédagogiques de chaque enseignant. Nous expliquons les principes généraux de cette approche avant de la décliner d'une part pour l'adaptation d'activités papier-crayon et d'autre part pour l'adaptation de sessions sur des logiciels pédagogiques. Enfin, nous présentons le processus permettant d'exploiter le modèle PERSUA2 afin de créer des séances de travail adaptées à chaque apprenant. Pour conclure cette partie, nous faisons un bilan de nos propositions en mettant en évidence leurs limites, ainsi que ce qui est réutilisable dans d'autres contextes.

La troisième partie est consacrée à la mise en œuvre de nos contributions théoriques au sein du logiciel Adapte et à leur évaluation. Nous expliquons le fonctionnement du logiciel avant de présenter son architecture en situant les propositions théoriques au sein de celle-ci. Nous illustrons ensuite l'utilisation du logiciel en listant tout d'abord les différents modes d'utilisation d'Adapte, puis en revenant sur les scénarios de l'introduction pour montrer en quoi Adapte peut permettre de les mener à bien. Nous exposons enfin les différentes évaluations que nous avons faites de nos modèles et de leurs mises en œuvre dans le logiciel Adapte.

Pour conclure cette thèse, nous proposons un bilan de notre travail en mettant en évidence ses avantages, ses limites, ainsi que les perspectives que soulève ce travail.

PARTIE 1. ÉTAT DE L'ART

PLAN DE LA PARTIE

Dans cette partie, nous faisons une revue de l'existant se rapportant à notre problématique de recherche. Celle-ci porte sur la définition d'un processus générique permettant de personnaliser des activités proposées aux apprenants en respectant les buts pédagogiques des enseignants.

Cette problématique s'intéresse d'une part aux trois facettes de la personnalisation de l'apprentissage (la prise en compte des individualités des apprenants, la prise en compte des besoins pédagogiques des enseignants et la façon d'associer une activité donnée à un apprenant donné), et d'autre part, à la définition d'un processus générique pouvant être implémenté dans un module indépendant personnalisant les activités papier ou les activités contenues dans un logiciel pédagogique existant.

Dans le premier chapitre, nous présentons les quatre approches existantes permettant de personnaliser l'apprentissage. La première consiste à conseiller l'enseignant sur le contenu de son enseignement. La seconde aide l'enseignant à créer les ressources pédagogiques dont il a besoin dans son enseignement. La troisième adapte les logiciels pédagogiques au besoin de chaque apprenant. Enfin, la dernière donne la possibilité à l'enseignant de scénariser ses situations d'apprentissage. Nous terminons ce chapitre par un bilan reprenant les trois facettes de la personnalisation de l'apprentissage contenues dans notre problématique. Pour chacune d'elles, nous mettons en évidence les points forts et les faiblesses des approches présentées.

Dans le second chapitre, nous présentons les deux approches permettant d'agir ou de piloter, de manière externalisée, un logiciel informatique et plus précisément un EIAH : l'approche des systèmes épiphytes et l'approche par composant. Nous présentons ensuite les différents standards permettant de décrire une ressource pédagogique pour voir si ces standards suffisent à décrire un logiciel pédagogique dans un but de personnalisation externalisée.

Nous concluons cet état de l'art en revenant sur certains scénarios d'usage présentés dans l'introduction du manuscrit pour souligner les lacunes des approches existantes permettant de personnaliser l'apprentissage. Nous indiquons ensuite les différences entre ces approches et la nôtre. Nous montrons de plus en quoi notre approche peut s'inscrire dans une démarche d'Ingénierie Dirigée par les Modèles (IDM).

CHAPITRE 1. COMMENT PERSONNALISER L'APPRENTISSAGE ?

PLAN DU CHAPITRE

1.1.	Introduction.....	35
1.2.	Conseiller l'enseignant sur le contenu de son enseignement	35
1.2.1.	Proposition des thèmes à étudier par l'apprenant	35
1.2.2.	Proposition d'outils à faire utiliser par l'apprenant.....	36
1.3.	Créer des ressources pédagogiques.....	37
1.3.1.	Création d'exercices.....	37
a.	Générateurs d'exercices automatiques.....	37
b.	Générateurs d'exercices manuels.....	40
c.	Générateurs d'exercices semi-automatiques	41
d.	Synthèse sur les générateurs d'exercices	44
1.3.2.	Création de listes d'exercices.....	45
1.3.3.	Création de logiciels pédagogiques	45
1.4.	Adapter les EIAH à chaque apprenant.....	47
1.4.1.	Que peut-on personnaliser dans un EIAH ?	47
1.4.2.	Qui personnalise l'EIAH ?	49
a.	Personnalisation par l'apprenant.....	49
b.	Personnalisation par l'enseignant	51
c.	Personnalisation par le système.....	53
d.	Synthèse sur les acteurs de la personnalisation	53
1.5.	Scénariser les séances d'apprentissage.....	54
1.5.1.	Définition du concept de scénario	54
1.5.2.	Création et échange de scénarios.....	55
1.6.	Synthèse des approches relatives à la personnalisation de l'apprentissage	55

1.1. INTRODUCTION

Personnaliser l'apprentissage consiste à prendre en compte les spécificités de chaque apprenant, ainsi que les buts pédagogiques des enseignants, pour proposer à partir de ces deux contraintes une adaptation des activités fournies aux apprenants. Différentes approches prenant en compte au moins une de ces deux contraintes pour fournir une aide lors de la personnalisation ou directement une personnalisation des activités fournies aux apprenants ont été proposées.

Dans ce chapitre, nous présentons ces approches en les regroupant selon leur but : conseiller l'enseignant sur le contenu de son enseignement, créer des ressources pédagogiques, adapter les EIAH à chaque apprenant ou scénariser les séances d'apprentissage. Nous concluons ce chapitre par une analyse de ces approches au regard de notre problématique de recherche.

1.2. CONSEILLER L'ENSEIGNANT SUR LE CONTENU DE SON ENSEIGNEMENT

Pour aider l'enseignant dans sa tâche de personnalisation, une première approche consiste à le conseiller dans le choix des thèmes à aborder avec chaque apprenant ou dans le choix des outils permettant de faire travailler chacun d'entre eux. Nous illustrons les deux versants de cette approche avec, respectivement, le projet LINGOT et la méthode ATRIUM.

1.2.1. PROPOSITION DES THÈMES À ÉTUDIER PAR L'APPRENANT

Le projet LINGOT [Lingot 1994] a pour but de concevoir des logiciels qui secondent les enseignants de collège et lycée dans la régulation des apprentissages en algèbre. Pour cela, il propose des outils aux enseignants leur permettant d'obtenir des stratégies d'apprentissage en fonction du profil cognitif de leurs élèves en algèbre élémentaire.

Ce projet s'appuie sur deux hypothèses. D'une part, identifier les compétences des apprenants permet aux enseignants de concevoir des séquences d'apprentissage ou de remédiation adaptées. Les séquences d'apprentissage contiennent des activités pour développer les compétences des élèves tandis que les séquences de remédiation contiennent des activités pour déstabiliser des fonctionnements inadéquats. D'autre part, en s'appuyant sur des recherches en didactiques, il est possible de concevoir un logiciel informatique capable d'aider l'enseignant à identifier les compétences des apprenants et capable de proposer des activités adaptées [Delozanne et al. 2002].

Un premier logiciel, PÉPITE, s'appuyant sur un travail didactique en algèbre élémentaire [Grugeon 1995], permet d'obtenir un profil cognitif dans ce domaine pour chaque élève, ainsi qu'un profil de groupe [Jean 2000].

Afin d'aider à proposer des stratégies d'enseignement pour remédier aux difficultés diagnostiquées, un second logiciel, PÉPISTÉRÉO [Vincent et al. 2005], classe les élèves par stéréotype en fonction de leur profil individuel. Un stéréotype est, pour les auteurs, la synthèse d'un profil auquel est associé un ensemble d'objectifs d'apprentissage prioritaires. Avec PÉPISTÉRÉO, l'enseignant dispose donc de groupes d'élèves ayant des compétences voisines en algèbre. PÉPISTÉRÉO propose ensuite des objectifs prioritaires d'apprentissage adaptés à chacun de ces groupements par stéréotype.

Avec cette approche, l'affectation d'une séquence d'apprentissage à un apprenant se fait automatiquement par le système en fonction du diagnostic fait suite à une évaluation de l'élève. Ce diagnostic permet de classer un élève dans un groupe ayant des caractéristiques comparables. Ainsi, l'affectation de la séquence est la même pour tous les élèves du groupe. Il y a donc une perte au niveau de la prise en compte de l'individualité de chaque élève. De plus, le système propose une séquence d'apprentissage contenant les thèmes que doit aborder le groupe d'élèves, mais cette séquence ne contient pas les activités qui permettraient de faire travailler les élèves sur ces thèmes. Chaque enseignant doit donc, en fonction des thèmes, créer des activités ou choisir des activités existantes permettant de faire travailler les élèves.

1.2.2. PROPOSITION D'OUTILS À FAIRE UTILISER PAR L'APPRENANT

La méthode pédagogique ATRIUM [Bruillard et al. 2000, Leroux 2002] offre un cadre de réflexion méthodologique aux enseignants afin de définir, pour une situation d'apprentissage donnée, les compétences à travailler, les activités à mettre en place pour travailler les compétences, et afin de déterminer les outils qui supporteront les activités en adéquation avec les compétences abordées. Le principe de la méthode est de partir des compétences à travailler pour aller jusqu'au choix des outils.

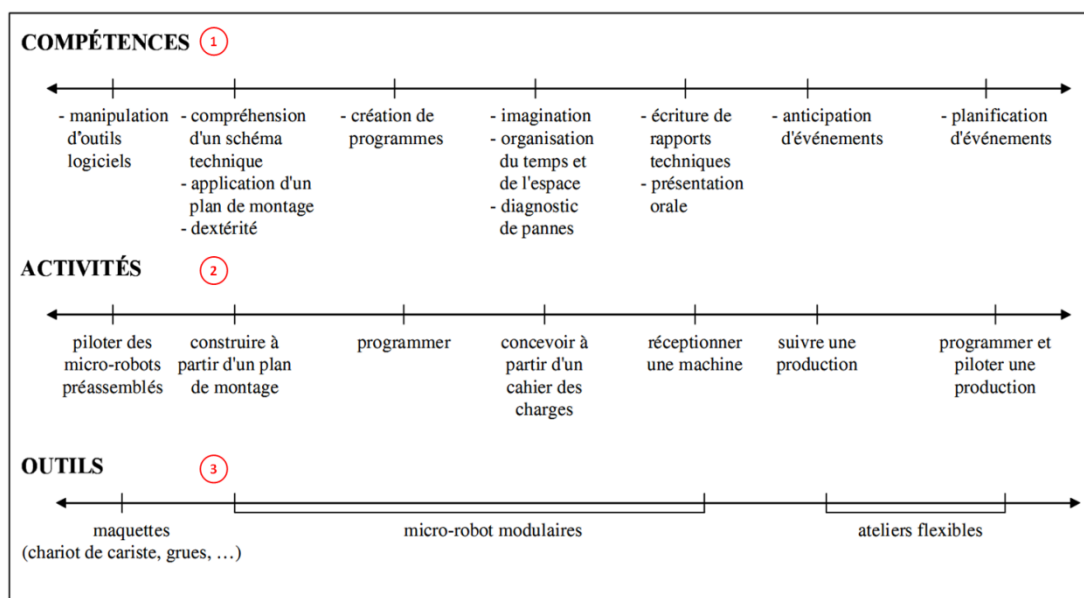


Figure 1-1 : Méthode pédagogique ATRIUM dans le contexte de la robotique pédagogique [Bruillard et al. 2000].

Ainsi, pour une formation donnée, un enseignant définit la liste des compétences visées (cf. ① sur la Figure 1-1). Pour chaque compétence, il indique la ou les activités à mettre en place et les outils qui supporteront ces activités (cf. ② sur la Figure 1-1). À partir de ce modèle de formation, l'enseignant pourra plus facilement choisir sur quel outil faire travailler un apprenant qui doit acquérir une compétence précise (cf. ③ sur la Figure 1-1).

Dans cette approche, l'enseignant définit lui-même quelle activité doit être effectuée pour travailler chaque compétence. Il définit de plus quelle compétence doit être travaillée par chaque apprenant. La méthode ATRIUM lui propose une aide méthodologique, mais ne l'aide aucunement lors de l'identification des besoins des apprenants, en termes d'apprentissage d'une compétence précise, ni lors de la définition des activités à fournir sur un outil donné.

1.3. CRÉER DES RESSOURCES PÉDAGOGIQUES

Pour aider l'enseignant dans sa tâche de personnalisation, une seconde approche consiste à créer les ressources pédagogiques dont il a besoin. Ces ressources peuvent être des exercices, des listes d'exercices, ou encore des logiciels pédagogiques. Lors de la création de ces ressources, l'enseignant peut être plus ou moins actif. Nous détaillons à présent les différentes propositions permettant d'aider l'enseignant lors de cette création de ressources, en précisant à chaque fois son degré d'intervention.

1.3.1. CRÉATION D'EXERCICES

Dans la plupart des EIAH fondés sur la résolution de problèmes, les exercices proposés aux apprenants sont issus d'une bibliothèque prédéfinie. Ce mode de fonctionnement a deux inconvénients majeurs. Le premier est que le nombre d'exercices et leur diversité s'en trouvent limités. Le second, et sûrement le plus important, est que les exercices ne sont pas toujours adaptés aux besoins des enseignants et à leur contexte de travail. Il y a donc un problème d'adaptation et donc d'intégration des EIAH à leur démarche pédagogique. Une des solutions est de créer des générateurs d'exercices.

Les premiers systèmes génératifs ont vu le jour dans les années 1970 avec l'intégration des techniques d'intelligence artificielle dans les environnements d'apprentissage. Ces systèmes génèrent des exercices et leurs réponses. Ils permettent de rendre innombrables la quantité d'exercices proposés et dans certains cas de paramétrer ces exercices [Bruillard 1997].

Une étude des générateurs d'exercices existants nous permet d'identifier trois types de générateurs. Tout d'abord, les générateurs automatiques, qui vont générer des exercices sans aucune intervention de l'utilisateur. À l'opposé, les générateurs manuels, appelés outils auteurs, qui vont aider et guider l'utilisateur lors de la conception des exercices. Et enfin, entre les deux, les générateurs semi-automatiques, qui vont générer des exercices automatiquement, mais selon des contraintes définies par l'utilisateur.

Nous définissons à présent plus précisément ces types de générateurs en donnant des exemples pour chacun d'eux.

A. GÉNÉRATEURS D'EXERCICES AUTOMATIQUES

Les générateurs d'exercices automatiques permettent de créer des exercices automatiquement, sans que l'utilisateur ne puisse influencer les choix du système. Avec ces générateurs, la personnalisation porte uniquement sur le thème général de l'exercice (domaine d'étude, notion précise d'un cours, connaissance, compétence, etc.). L'utilisateur ne peut aucunement personnaliser le contenu proprement dit de l'exercice (vocabulaire, valeurs, etc.). La fonction de génération peut être intégrée dans un EIAH (dans ce cas l'utilisateur correspond au module pédagogique de l'EIAH) ou utilisable en tant que telle par un enseignant.

Dans le cas où l'utilisateur correspond au module pédagogique d'un EIAH, le générateur produit des exercices qui seront proposés à l'apprenant par l'EIAH lui-même.

Un premier exemple de générateur piloté par le module pédagogique d'un EIAH est celui intégré au système `DEBUGGY` [Burton 1982]. Cet EIAH, diagnostiquant les causes d'erreur dans la soustraction écrite, crée un exercice adapté après chaque erreur de l'apprenant.

Un second exemple est le générateur contenu dans le système READING TUTOR [Mostow et al. 2004] du projet LISTEN [LISTEN 1993, Mostow et al. 2001]. Cet EIAH propose, au fur et à mesure de la lecture d'un texte par l'apprenant, des questions de compréhension. Une question correspond soit à une phrase du texte dont un mot a été supprimé (cf. partie gauche de la Figure 1-2), soit à une expression du texte dont il faut trouver le sens (cf. partie droite de la Figure 1-2). À chaque fois, des propositions de réponses sont fournies. Pour faire varier le niveau de difficulté des questions, le générateur étiquète les mots du texte selon leur fréquence d'apparition dans la langue employée dans le texte (dans ce cas, l'anglais). La difficulté d'une question est alors, entre autres, inversement proportionnelle à la fréquence des mots sur lesquels elle porte.

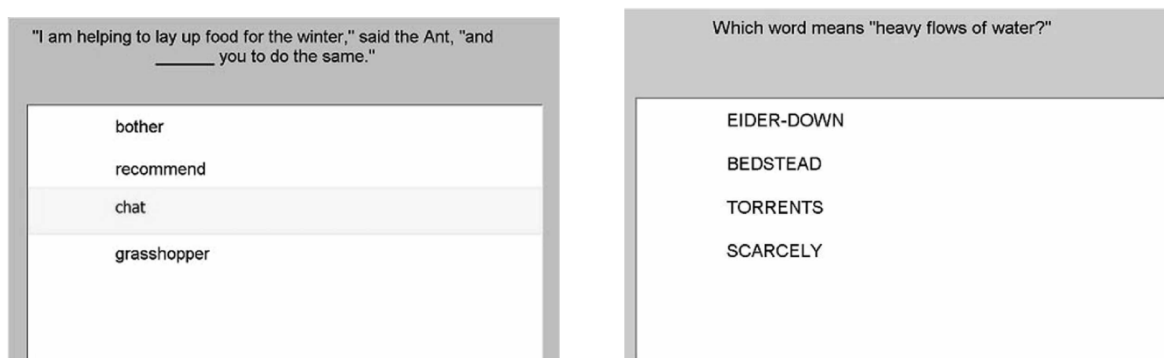


Figure 1-2 : Questions générées par READING TUTOR [Mostow et al. 2004].

Dans le cas où l'utilisateur est un enseignant, le générateur produit des exercices à la demande de l'enseignant qui les proposera ensuite à ses élèves.

Un exemple est le générateur d'exercices contenu dans le micromonde APLUSIX [Bouhineau et al. 2005b]. Ce micromonde est destiné à l'apprentissage de l'algèbre élémentaire dans les lycées et collèges [Nicaud 1993, Aplusix 2009]. Le but du générateur est de construire automatiquement des exercices pour faire travailler les apprenants sur les différents types de problèmes algébriques (calculer, développer, factoriser, résoudre). Il est composé d'une hiérarchie de patrons d'exercices qui définit l'ensemble des exercices possibles et d'une interface proposant une carte de tests.

```
{[nom FactorDistSTD23Z]
[sorteDe FactorDistSTD]
[patron <<ax^2+bx>>]
[domaine ((c entier+ petit)(d entier+ petit)(e entier* petit))]
[avec ((<> c 1) (:= a (* d c)) (:= b (* e c)))]}
```

Figure 1-3 : Patron d'exercices du générateur d'exercices d'APLUSIX [Bouhineau et al. 2005b].

Chaque patron équivaut à une expression algébrique comportant des paramètres. Comme pour le patron de la Figure 1-3, chaque paramètre possède un domaine de définition et chaque patron possède des contraintes de validité. Les expressions varient en complexité en fonction de la diversité des règles utilisées, du nombre de transformations à faire pour les résoudre ainsi que du degré et de la forme des polynômes utilisés. À l'exécution, le patron est instancié en prenant, pour chaque paramètre, une valeur du domaine au hasard et en vérifiant que les contraintes de validité sont respectées. Ensuite, un algorithme de mise en forme est utilisé pour éliminer les éléments neutres, les parenthèses inutiles, etc. De cette manière, un patron d'exercices permet de créer de multiples exercices. La Figure 1-4 illustre cette diversité en montrant deux exercices créés à partir d'un même patron d'exercices.

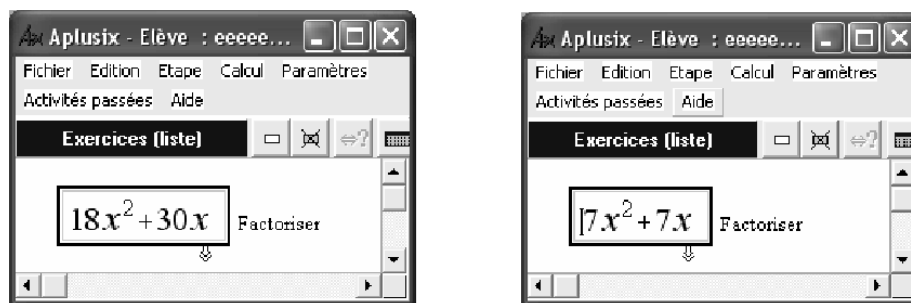


Figure 1-4 : Tirages aléatoires pour le patron d'exercices d'APLUSIX de la Figure 1-3 [Bouhineau et al. 2005b].

La carte des tests regroupe l'ensemble des exercices définis. Elle présente ces exercices en les regroupant par famille, et fournit une description pour chacune d'elle (cf. Figure 1-5). Cette carte des tests permet de lancer une session d'entraînement ou de test sur une liste d'exercices produite automatiquement par le générateur.

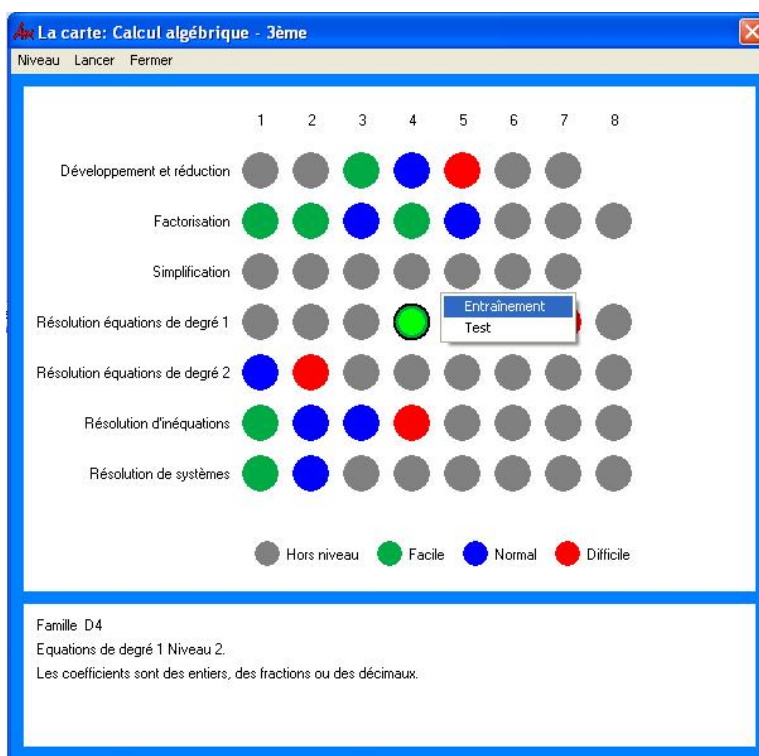


Figure 1-5 : Carte de tests d'APLUSIX [Aplusix 2009].

Dans certains cas, le générateur d'exercices peut être appelé directement par un module pédagogique d'EIAH ou être utilisé avec une interface propre à l'enseignant. C'est le cas du générateur présenté dans [Pecego 1996] qui crée des exercices de géométrie en partant d'un théorème de base. C'est le générateur qui choisit aléatoirement le théorème de base, mais il peut être guidé par l'utilisateur ou le module pédagogique d'un EIAH. Le générateur fonctionne grâce à un mécanisme de chaînage arrière à partir d'une base de connaissances. En partant du théorème de base, le générateur applique un mécanisme de complications successives et peut ainsi fournir l'énoncé des exercices, mais également leur solution en donnant la trace inversée du processus de création.

L'avantage majeur des générateurs automatiques est qu'ils peuvent créer rapidement une grande quantité d'exercices. Mais leur principale limite est que les utilisateurs ne peuvent pas

intervenir dans la génération des exercices, les enseignants ne peuvent donc pas adapter ceux-ci à leurs habitudes de travail.

B. GÉNÉRATEURS D'EXERCICES MANUELS

Les générateurs d'exercices manuels, ou outils auteurs, ont pour principale caractéristique le fait qu'ils ne construisent pas eux-mêmes les énoncés des problèmes, ni leurs solutions. Cette tâche revient à l'enseignant. Il construit chaque énoncé et fournit la ou les réponses attendues. L'enseignant est ainsi auteur de ses exercices. Les générateurs manuels peuvent créer de nombreux exercices, comme des exercices à trous, des reconstitutions de textes, des appariements, des exercices sous forme de jeux (mots croisés, puzzle, quiz...), des exercices multimédias (lecture, écoute...) et ce dans divers domaines d'enseignement. Les exercices créés vont être de deux types : à réponse ouverte (exercice demandant une rédaction de l'apprenant sur ordinateur ou sur papier) ou fermée (exercice possédant une liste finie de réponses possibles, comme les exercices à trous, les questionnaires à choix multiples...).

Il faut noter qu'il s'agit d'outils destinés aux enseignants non informaticiens, leur seconde caractéristique est que leur interface est donc adaptée à ce public.

Enfin, leur dernière caractéristique est qu'ils ne possèdent aucune connaissance sur les exercices qu'ils proposent et sur les réponses préenregistrées. Ils ne possèdent donc pas de fonctionnalité d'aide (sauf si celle-ci est préalablement définie par l'enseignant) et ne peuvent proposer qu'un diagnostic très limité. Dans le cas des exercices à réponses fermées, le générateur peut fournir une correction automatique des réponses de l'apprenant, dans la mesure où l'utilisateur du système auteur a décrit la manière de les corriger. À l'inverse, les exercices à réponses ouvertes nécessitent une correction de l'enseignant car le générateur ne possédant pas de représentation du problème et des réponses de l'apprenant, il ne pourra pas les corriger.

Nous pouvons citer plusieurs exemples de générateurs manuels. Le premier est l'outil auteur GENEVAL [David et al. 1996, GenEval 1997, Cogne et al. 1998], issu du projet européen de recherche ARIADNE [ARIADNE 2006]. Il permet de créer des exercices hypermédias dans n'importe quel domaine. La Figure 1-6 en montre un en chimie et la Figure 1-7 un en mécanique. Les exercices créés ont tous la même structure : un énoncé et la réponse associée qui peut être donnée selon trois niveaux progressifs, qui correspondent à deux niveaux d'aide, puis à une réponse détaillée. À l'inverse d'autres systèmes, les apprenants devront résoudre sur papier les exercices proposés avant de comparer leur réponse à celle(s) indiquée(s) par le système.

Figure 1-6 : Page énoncé d'un exercice de cinétique chimique créé avec GENEVAL [GenEval 1997].

Figure 1-7 : Pages énoncé d'un exercice de mécanique (en arrière plan) et réponse (au premier plan, en bas à droite) créé avec GENEVAL [GenEval 1997].

Un second exemple est le logiciel CREEEXO [Creexo 2007] issu des pratiques enseignantes. Il permet de créer des exercices dans divers domaines (mathématiques, géographie...). Dans ce système, l'enseignant construit l'écran correspondant à l'exercice grâce à des éléments disponibles à l'interface (zones de saisie, zones de texte, cases à cocher...). Il spécifie ensuite la fonction des objets placés, puis donne les diverses réponses attendues. Une fois l'exercice créé et sauvegardé dans CREEEXO, l'apprenant prend connaissance de l'énoncé et résout les exercices dans READEXO, l'environnement apprenant associé.

Un dernier exemple est le logiciel EXERCISE GENERATOR PLUS [ExerciseGeneratorPlus], issu du commerce, qui permet à un enseignant de créer divers types d'exercices à trous ou de vocabulaire (cf. Figure 1-8). Ces exercices portent sur les textes de la bibliothèque du logiciel ou sur des textes fournis par l'enseignant. Une fois créé, l'exercice se voit attribuer un niveau de difficulté. L'apprenant peut alors travailler sur un exercice de son niveau. Il le résout sur papier ou dans le logiciel.

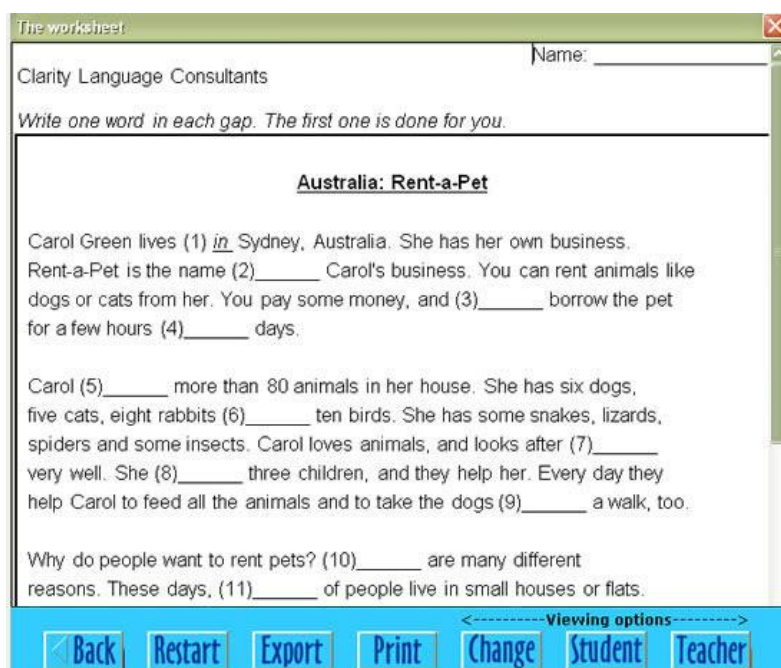


Figure 1-8 : Exercice créé avec EXERCISE GENERATOR PLUS [ExerciseGeneratorPlus].

L'avantage majeur des générateurs manuels est qu'ils laissent une totale liberté à l'enseignant tant dans le domaine d'application que dans le contenu pédagogique de l'exercice. En contrepartie, ils ont une limite extrêmement forte du fait que l'enseignant doit définir entièrement l'exercice et ses solutions.

C. GÉNÉRATEURS D'EXERCICES SEMI-AUTOMATIQUES

Les générateurs d'exercices semi-automatiques sont un intermédiaire aux deux types de générateurs que nous venons de présenter. Ces générateurs construisent eux-mêmes les énoncés des exercices, mais en laissant l'utilisateur intervenir dans le processus de création. Ainsi, l'utilisateur influe sur les exercices générés en spécifiant un ensemble de contraintes. Cet utilisateur peut être soit le module pédagogique d'un EIAH, soit un enseignant.

Le système CEP [Giroire 1989] est un premier exemple de générateur de ce type. Il a pour but de générer des énoncés d'exercices dans des domaines variés, liés au monde réel, comme la physique ou la chimie. Dans ce système, la création d'un exercice se fait en trois grandes étapes. La première est la création de la structure de l'exercice à concevoir. Pour cela, il faut définir les

grandeurs impliquées dans l'énoncé (volume, tension...) et les relier à des types d'objets (récipient, résistor...). L'utilisateur indique alors la grandeur ou la formule sur laquelle portera le problème et choisit une stratégie de construction. Cette stratégie va définir la classe de problème à laquelle le problème généré doit appartenir. La seconde étape est l'instanciation de la structure d'exercice. Il s'agit de remplacer tous les types d'objets par des objets réels (par exemple un récipient devient un flacon) et de choisir une valeur pour chaque grandeur. Ces valeurs sont issues d'intervalles ou sélectionnées dans des ensembles énumérés de valeurs. Enfin la dernière étape est l'habillage en langue naturelle.

Dans la continuité du travail entrepris avec CEP, le système à base de connaissances SYGEP [Pecego 1998] crée des énoncés de problèmes dans des domaines divers utilisant des dispositifs ou des expérimentations pour donner lieu à des questions (circuits électriques, expériences de mécanique...). Dans ce système, basé essentiellement sur la manipulation des formules du domaine, l'utilisateur fixe ses objectifs pédagogiques en spécifiant un ensemble de contraintes.

Autre exemple de génération semi-automatique d'exercices, le système GENAMBRE du projet AMBRE. Ce projet a pour but de concevoir des EIAH utilisant un processus d'apprentissage inspiré du cycle du Raisonnement à Partir de Cas pour l'enseignement de méthodes [AMBRE 2000, Guin-Duclosson et al. 2002]. Un EIAH AMBRE comporte un module apprenant et un module enseignant. AMBRE-ENSEIGNANT [Duclosson et al. 2005b, Jean-Daubias et al. 2009b] est un outil d'assistance à l'enseignant contenant, entre autres, le générateur de problèmes GENAMBRE, pour créer les exercices de son choix et des séquences d'apprentissage personnalisées. Ce générateur semi-automatique propose, pour chaque problème généré, un énoncé en langue naturelle, ainsi qu'une description du problème compatible avec le résolveur utilisé par le module apprenant. La génération du problème respecte un certain nombre de caractéristiques que doit comporter le problème. Ces caractéristiques, spécifiées par l'enseignant, portent sur la structure du problème (cf. ① sur la Figure 1-9), les traits de surface de l'énoncé (cf. ② sur la Figure 1-9), les valeurs contenues dans l'énoncé (cf. ③ sur la Figure 1-9) et les éléments de complication linguistique de l'énoncé (cf. ④ sur la Figure 1-9). Les caractéristiques non définies par l'enseignant le seront par GENAMBRE au moment de la génération du problème. AMBRE-ENSEIGNANT permet ainsi d'aller d'une génération entièrement aléatoire des problèmes (lorsque l'enseignant ne définit aucune caractéristique des problèmes) à une génération entièrement contrainte (lorsque l'enseignant définit toutes les caractéristiques des problèmes).

Autre exemple, au sein du projet LINGOT [Lingot 1994], qui s'intéresse au diagnostic de la compétence algébrique des élèves en fin de collège, le système PÉPIGEN [Prévit et al. 2007] crée des exercices de diagnostic à partir de modèles paramétrés d'exercices. Les modèles paramétrés d'exercices sont mis au point par des informaticiens à partir de spécifications établies par un didacticien. Ces modèles sont utilisés par un enseignant ou didacticien qui fixe les paramètres et fait générer par le système PÉPIGEN un exercice particulier. La Figure 1-10 montre l'interface de PÉPIGEN permettant à un enseignant de saisir un énoncé à l'aide d'une palette comportant des mots, des mots de liaison, des signes de ponctuation et des chiffres. PÉPIGEN calcule au fur et à mesure l'expression algébrique traduisant l'énoncé saisi. Lorsque l'enseignant a terminé de saisir l'énoncé, PÉPIGEN le complète à l'aide de la forme réduite de l'expression algébrique.

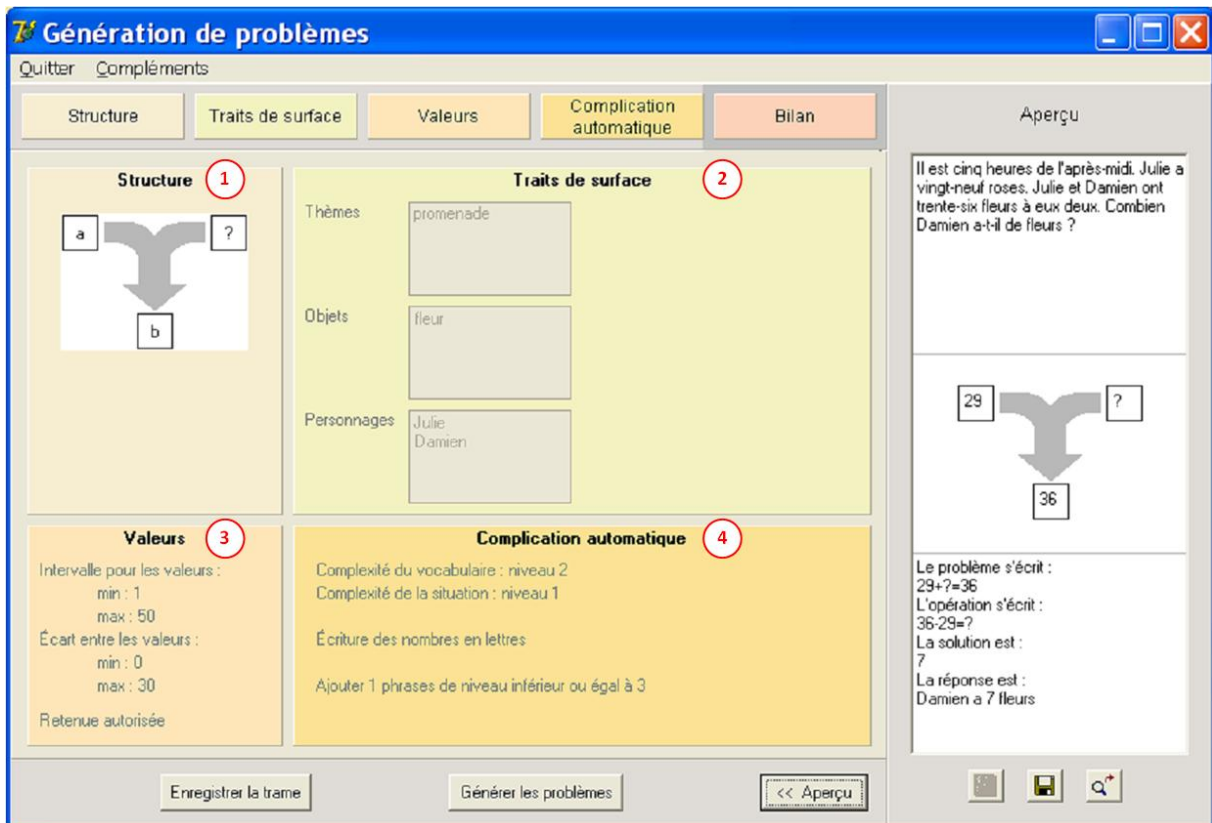


Figure 1-9 : Écran de bilan du générateur de problèmes du module AMBRE-ENSEIGNANT de AMBRE-ADD [Duclosson et al. 2005a].

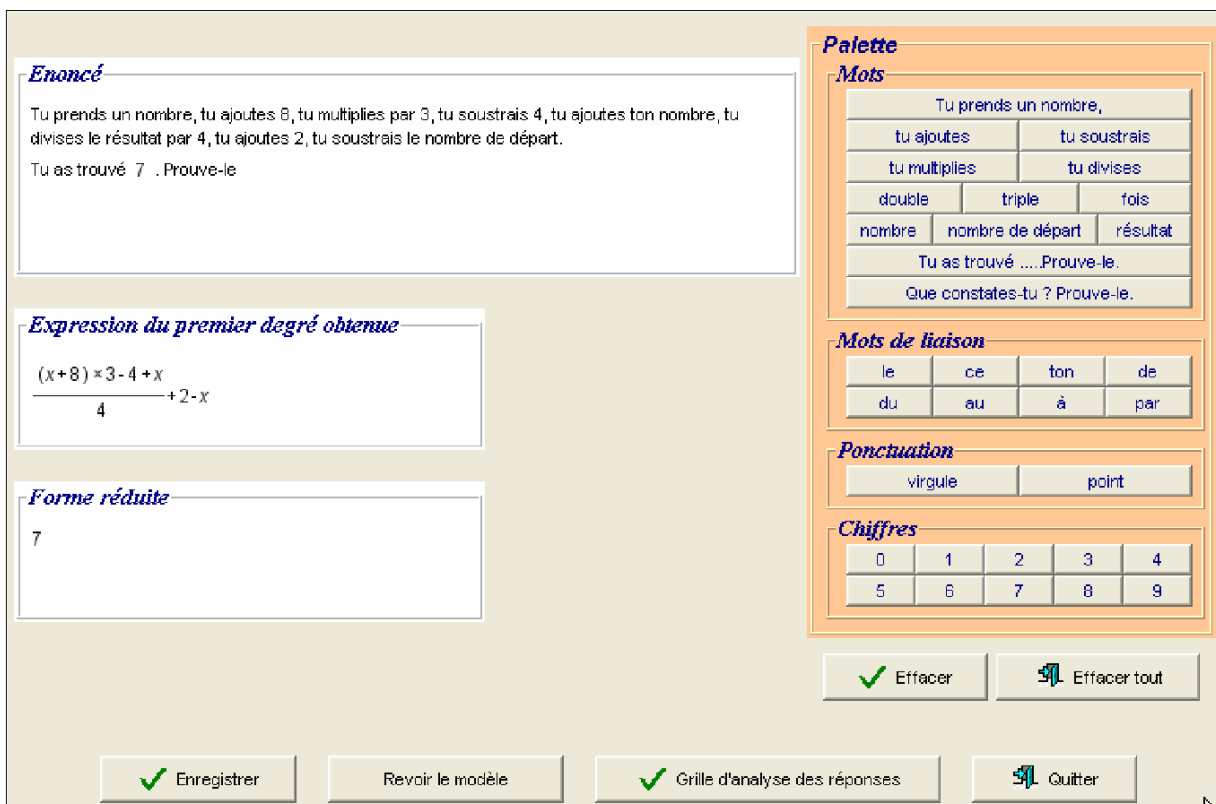


Figure 1-10 : Interface de création d'un exercice de type « Prestidigitateur » dans PÉPIGEN [Prévit et al. 2007].

Prenons un dernier exemple de générateur d'exercices semi-automatiques. Au sein du projet TREE ayant pour but le développement d'un tuteur intelligent pour la classification et l'identification des différentes espèces végétales européennes, le système SIETTE [Conejo et al. 2004] permet de générer des tests pour aider les étudiants à identifier les connaissances qu'ils maîtrisent. Dans SIETTE, les enseignants ou les experts du domaine spécifient des tests en précisant les questions (cf. Figure 1-12), les sujets étudiés et divers paramètres (cf. Figure 1-11). Ensuite, lorsqu'un apprenant cherche à obtenir un test, celui-ci est créé dynamiquement à partir des contraintes définies précédemment.

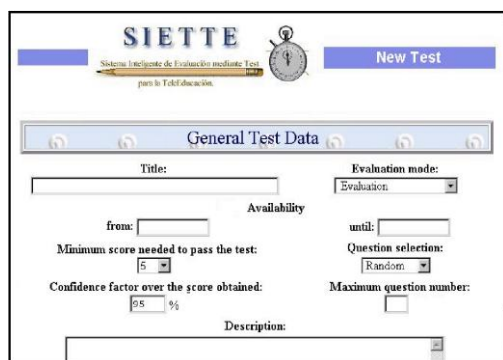


Figure 1-11 : Interface de définition d'un test dans SIETTE [Conejo et al. 2004].

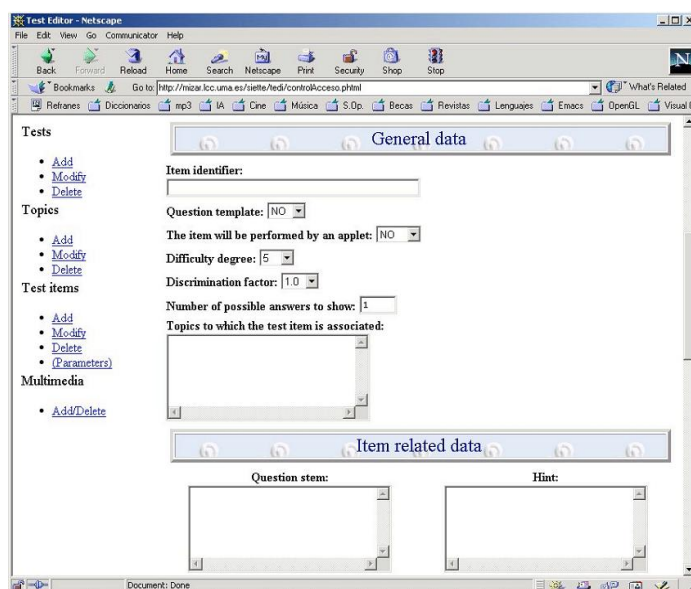


Figure 1-12 : Interface de définition d'une question dans SIETTE [Conejo et al. 2004].

Les générateurs semi-automatiques possèdent donc les avantages des générateurs automatiques (génération rapide d'une quantité importante d'exercices), mais en donnant une solution à leur principale limite : les enseignants peuvent paramétrer les exercices générés et donc les adapter à leurs besoins.

D. SYNTHÈSE SUR LES GÉNÉRATEURS D'EXERCICES

Dans cette section, nous faisons un bilan des forces et faiblesses de chacun des trois types de générateurs que nous venons de présenter. Dans ce bilan, nous utilisons le terme *utilisateur* pour parler indifféremment d'un enseignant ou d'un EIAH utilisant les générateurs.

En prenant comme premier critère l'implication de l'utilisateur lors de la génération des exercices, nous retenons que les générateurs automatiques ne laissent aucune possibilité d'intervention de l'utilisateur, tandis que les générateurs semi-automatiques ou manuels laissent intervenir l'utilisateur, quoiqu'à des degrés divers, dans le processus de génération. Dans les générateurs semi-automatiques, les utilisateurs peuvent contraindre le contenu pédagogique de l'exercice. Ainsi, l'utilisateur pourra générer des exercices allant du tout aléatoire au tout déterminé. Ces générateurs ne permettent néanmoins de créer des exercices que dans des domaines pour lesquels ils possèdent les bases de connaissances. En revanche, les générateurs manuels laissent beaucoup plus de possibilités à l'enseignant. Celui-ci ne contraint pas le contenu pédagogique de l'exercice, mais le définit entièrement.

En prenant comme second critère l'efficacité, nous retenons que les générateurs automatiques ou semi-automatiques permettent de créer rapidement un grand nombre d'exercices. À l'inverse, les générateurs manuels demandent à l'enseignant de créer l'énoncé et la correction de chaque exercice, ce qui est beaucoup plus coûteux en temps.

En conclusion de cette revue de l'existant sur les différents générateurs d'exercices, nous retenons que seuls les générateurs semi-automatiques d'exercices permettent à la fois une prise en compte de besoins de l'utilisateur et une génération rapide d'un grand nombre d'exercices. Dans notre contexte de personnalisation en fonction des buts pédagogiques de l'enseignant, ce type de générateurs semble donc le plus pertinent.

1.3.2. CRÉATION DE LISTES D'EXERCICES

Nous venons de voir comment un exercice pouvait être généré, avec ou sans intervention extérieure. Nous regardons maintenant comment certains systèmes proposent de combiner des exercices existants pour fournir à l'apprenant ou à l'enseignant des listes d'exercices correspondant à un thème à étudier, une connaissance à tester, etc.

Pour générer automatiquement une feuille d'exercices portant sur des connaissances données, [Hibou et al. 2003] proposent d'indexer les exercices en fonction des connaissances utiles pour les résoudre. Pour cela, les auteurs utilisent le démonstrateur automatique de théorèmes ARGOS [Spagnol 2001] afin de réaliser une indexation d'exercices de géométrie à partir des théorèmes et propriétés qui servent à leur résolution. Il est alors possible d'obtenir une liste d'exercices sur un sujet donné, dans le cas présent un théorème de géométrie, en interrogeant la base de données. Cette approche implémentée pour la géométrie pourrait être réutilisée pour aider la planification d'activités dans un EIAH dans la mesure où celui-ci intègre un résolveur de problèmes et dans la mesure où il est possible de le faire communiquer avec un module d'indexation d'exercices.

L'hypermédia adaptatif de langue basque HEZINET [Lopez-Cuadrado et al. 2007] propose une évaluation intelligente qui se fait par l'intermédiaire d'un test adaptatif regroupant un ensemble de questions nommées items. Les items proposés sont issus d'une banque de 252 items avec à chaque fois quatre réponses possibles. Pour pouvoir choisir un item dans la base, la banque d'items a été calibrée selon trois paramètres : la difficulté, la discrimination et la pseudo-chance. La difficulté représente le fait qu'un apprenant possédant une habileté de niveau N doit savoir répondre à des items de difficulté N. La discrimination indique si un item catalogue une habileté de l'apprenant. La pseudo-chance représente la chance de réponse juste en n'ayant pas l'habileté requise. Lors du passage du test par l'apprenant, l'item suivant dépend des réponses fournies précédemment et est choisi avec un niveau de difficulté proche de l'habileté à estimer, une capacité de discrimination élevée et une probabilité de pseudo-chance basse.

Ces exemples illustrent la nécessité d'indexer les bases d'exercices en fonction d'un certain nombre de métadonnées (compétences mises en jeu, niveau de difficulté, etc.). Lors de la sélection des exercices, l'enseignant ou le module pédagogique d'un EIAH peuvent ainsi contraindre les choix du système en spécifiant des valeurs pour une ou plusieurs de ces métadonnées.

1.3.3. CRÉATION DE LOGICIELS PÉDAGOGIQUES

La troisième facette de la création de ressources pédagogiques concerne les logiciels pédagogiques. De nombreux outils auteurs ont été proposés pour aider l'enseignant à concevoir

un logiciel correspondant à ses besoins pédagogiques. Nous présentons dans cette section des exemples de ces systèmes.

Le système auteur SIMQUEST [SimQuest 1996, Van Joolingen et al. 2003, De Jong 2004] permet de créer des tuteurs intelligents de simulation. Il fournit aux enseignants un cadre à la fois conceptuel (permettant de créer le modèle de simulation) et technique (permettant de créer l'interface du simulateur) lors de la création des logiciels. Son éditeur (cf. Figure 1-13) propose une liste de composants. Chaque composant est lié à une variable du modèle de simulation ou à une action spécifique (lecture, pause, etc.). SIMQUEST a permis de créer une vingtaine de logiciels de simulation dans différents domaines scientifiques, comme le simulateur de génie mécanique MOMENT présenté sur la Figure 1-14.

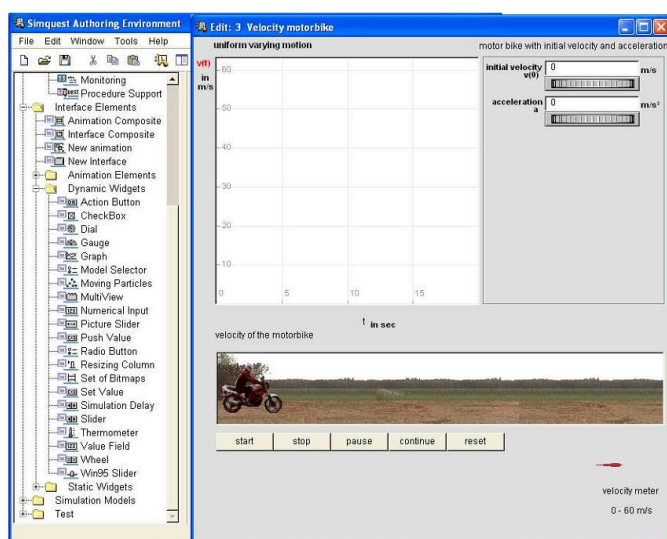


Figure 1-13 : Éditeur de simulations dans SIMQUEST [Van Joolingen et al. 2003].

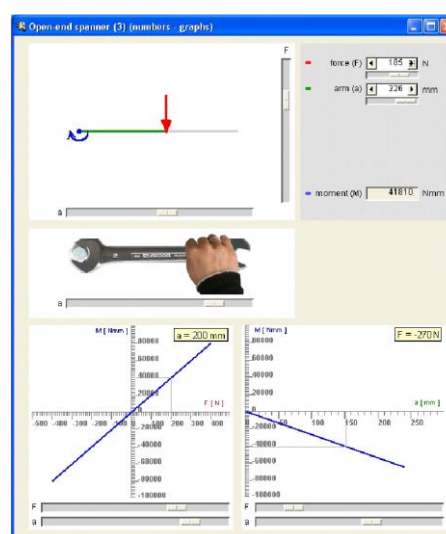


Figure 1-14 : Simulateur créé avec SIMQUEST [De Jong 2004].

L'atelier d'ingénierie pédagogique ADISA (anciennement AGD ou AGDI) [Paquette et al. 1996a] est destiné à outiller les différentes étapes du processus de construction des systèmes d'apprentissage. Il contient des éditeurs facilitant la modélisation des connaissances enseignées, permettant la définition des événements d'apprentissage et leur structuration en différents scénarios, ainsi que la définition des publics cibles, de leur niveau d'habileté de départ et du niveau visé, etc. Cet atelier s'appuie sur la méthodologie d'ingénierie des systèmes d'apprentissage MISA [Paquette et al. 1997a, Paquette et al. 1997b] qui permet de modéliser la tâche de conception de cours se déroulant sur des logiciels ou des sites internet pédagogiques. Pour aider l'enseignant lors de l'utilisation de l'atelier ADISA, celui-ci a été doté d'un système conseiller. Un système conseiller est un système informatique qui propose une aide active « intelligente » aux utilisateurs d'un logiciel particulier, conseils fondés sur une analyse des actions et des productions de l'utilisateur ([Winkels 1992] cité dans [Paquette et al. 2002]). Le système conseiller d'ADISA [Paquette et al. 1996a] analyse les interactions entre l'utilisateur (le concepteur du système d'apprentissage) et l'atelier logiciel pour aider cet utilisateur à mener à bien sa tâche, en l'occurrence à concevoir son système d'apprentissage.

Plus récemment, l'idée de méta-outils auteurs est apparue [Bourdeau et al. 2002, Murray 2003a]. Un méta-outil auteur correspond à un outil créateur ou paramétreur d'outils auteurs génériques et permet à un utilisateur expert de créer une ontologie correspondant aux besoins d'un enseignant ou d'un formateur spécifique. Une ontologie est une spécification explicite d'une conceptualisation [Gruber 1993], c'est-à-dire un ensemble de termes représentant le sens des concepts d'un domaine et des relations qui les lient. Une fois qu'un expert a créé une ontologie

spécifiquement pour un enseignant et paramétré l'outil auteur générique à l'aide des méta-outils, l'outil auteur est utilisable par l'enseignant lui-même. Ainsi, l'enseignant dispose d'un outil adapté à ses besoins. Le système auteur EON [Murray 2003a] a été conçu selon ce principe. Cet outil générique, une fois paramétré par un expert, devient un outil auteur spécifique à un enseignant. Dans EON, l'expert peut adapter le vocabulaire utilisé pour décrire le domaine de l'enseignant, pour créer le modèle de l'apprenant, ainsi que celui de l'éditeur de stratégies d'enseignement. EON a été utilisé pour construire plusieurs prototypes de tuteurs dans des domaines aussi variés que l'apprentissage du japonais (cf. Figure 1-15) ou de la chimie (cf. Figure 1-16) et mettant en œuvre des stratégies d'enseignements différentes.



Figure 1-15 : KEIGO TUTOR, tuteur intelligent pour l'apprentissage du japonais, créé avec EON [Murray 2003a].

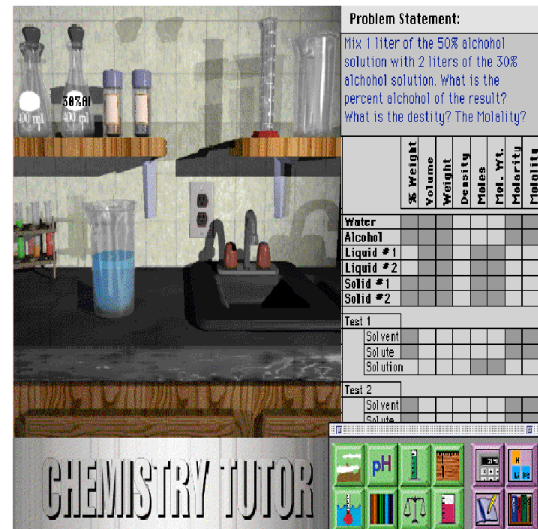


Figure 1-16 : CHEMISTRY WORKBENCH, tuteur intelligent de chimie, créé avec EON [Murray 2003a].

Les outils actuellement à disposition des enseignants pour les aider à concevoir des logiciels, éventuellement augmentés de systèmes conseillers ou paramétrés par un expert, possèdent les mêmes forces et les mêmes faiblesses que les générateurs manuels d'exercices : ils laissent une totale liberté à l'enseignant tant dans le domaine d'application que dans le contenu pédagogique du logiciel créé, mais demandent une forte implication de l'enseignant et requièrent un temps d'utilisation assez important.

1.4. ADAPTER LES EIAH À CHAQUE APPRENANT

Pour aider l'enseignant dans sa tâche de personnalisation, une troisième approche consiste à adapter les logiciels pédagogiques aux spécificités de chaque apprenant. Nous indiquons à présent, en nous appuyant sur des exemples, ce qui peut être personnalisé dans un EIAH, avant de parler de l'acteur de cette personnalisation.

1.4.1. QUE PEUT-ON PERSONNALISER DANS UN EIAH ?

Le paramétrage d'un EIAH « concerne la possibilité de contrôler, de modifier à son gré les caractéristiques du logiciel, non seulement pour l'adapter à des situations d'apprentissage particulières, mais aussi pour en adapter le contenu, l'agencement ou le comportement

pédagogique » [Hû 2001]. La personnalisation d'un EIAH peut donc porter sur plusieurs facettes de l'EIAH.

Concernant les tuteurs intelligents et les hypermédias adaptatifs, [Brusilovsky 1999] présente un inventaire des technologies utilisées pour adapter ces systèmes. Cet inventaire se focalise sur les technologies mais permet, à travers les éléments cibles de l'adaptation, de connaître les éléments personnalisables dans les systèmes étudiés.

L'auteur regroupe les technologies permettant l'adaptation en deux grandes catégories : celles agissant sur le parcours des apprenants dans les systèmes et celles concernant l'aide à la résolution de problèmes.

Les technologies agissant sur le parcours des apprenants permettent : d'adapter l'ordre des leçons et des exercices à faire pour chaque apprenant, comme dans SIETTE [Ríos et al. 1999] ou ILESA [López et al. 1998] ; de guider l'apprenant en lui indiquant quel élément de la formation serait à faire en fonction de ses connaissances actuelles et de son but d'apprentissage, comme dans INTERBOOK [Brusilovsky et al. 1998] ou ELM-ART [Brusilovsky et al. 1996] ; de masquer ou rendre inactifs des liens qui mènent vers des informations que l'apprenant ne pourra pas encore comprendre, comme dans REMEDIAL MULTIMEDIA SYSTEM [Anjaneyulu 1997] ; et de présenter la même information différemment selon l'apprenant en détaillant, masquant, résumant ou illustrant des parties de la formation, comme dans PT [Kay et al. 1997], AHA [De Bra et al. 1998] ou METALINKS [Murray et al. 1998, Murray 2003b].

Les technologies de cette catégorie agissent sur l'organisation des activités, mais également sur le choix des activités elles-mêmes, et sur l'interface du logiciel.

Les technologies concernant l'aide à la résolution de problèmes permettent : de corriger et d'aider l'apprenant au cours de la résolution, comme dans LISP-TUTOR [Anderson et al. 1985] ; de déterminer, à partir de l'analyse des réponses de l'apprenant (si une réponse finale est correcte/incorrecte, si les connaissances utilisées sont correctes, etc.), les rétroactions à fournir et de mettre à jour le modèle de l'apprenant, comme dans PROUST [Johnson 1986] ; de fournir un exemple de résolution expliqué ou des problèmes résolus du même type, comme dans ELM-PE [Weber 1996] ou ELM-ART [Brusilovsky et al. 1996].

Les technologies de cette catégorie agissent sur les fonctionnalités du logiciel (aide, diagnostic, etc.) et permettent dans certains cas de modifier les rétroactions à fournir à l'apprenant.

En complément de cet inventaire centré sur les tuteurs intelligents et les hypermédias adaptatifs, nous avons étudié trente logiciels pédagogiques. Ces logiciels sont très variés de part leurs types (tuteurs intelligents, micromondes, simulateurs, exercices, applications web), leurs provenances (issus de la recherche, des pratiques enseignantes ou du commerce), leurs publics (formation scolaire, universitaire ou professionnelle) mais également de part les domaines d'apprentissage traités. La liste complète des logiciels étudiés et leur description sont fournies en Annexe A page 283. Nous avons établi que dans ces logiciels, la personnalisation peut porter sur :

- les **activités** proposées, en générant de nouvelles activités ou en choisissant une activité adaptée parmi celles contenues dans le logiciel ;
- la **séquence des activités**, en choisissant le nombre d'activités et l'ordre dans lequel les activités sont proposées ;
- les **fonctionnalités**, en définissant celles disponibles (par exemple autoriser l'accès à l'aide, au diagnostic, aux compagnons, aux outils de dessins, de calculs) et selon quelle temporalité (constamment, à certains moments, avec un nombre d'accès autorisés, etc.) ;

- les **rétroactions** proposées aux apprenants, en changeant l'accessibilité et le contenu des messages (adapter le vocabulaire ou la présentation de ceux-ci) ;
- l'**interface** du logiciel, en choisissant la langue du logiciel, en paramétrant la police (taille, style, etc.), en adaptant les couleurs (pour les dyslexiques par exemple), en activant la synthèse vocale, etc.

Nous reviendrons plus en détail sur l'identification de ces cinq facettes dans la section 6.4.1.

1.4.2. QUI PERSONNALISE L'EIAH ?

Nous venons de voir ce qui pouvait être personnalisé dans un EIAH. Nous nous intéressons à présent aux différents acteurs pouvant mettre en place une personnalisation : l'apprenant lui-même, l'enseignant et le module pédagogique d'un EIAH. Pour chaque type d'acteur, nous donnons des exemples avant de résumer les possibilités offertes, les forces, ainsi que les faiblesses de la personnalisation par un acteur donné.

A. PERSONNALISATION PAR L'APPRENANT

Le principal acteur de l'apprentissage est l'apprenant lui-même. C'est pourquoi certains systèmes guident l'apprenant dans le choix des activités qu'il va effectuer, voire le laissent choisir seul ces activités.

Un exemple de système permettant à l'apprenant de choisir précisément l'activité qu'il va effectuer est le système ANDES [VanLehn et al. 2005, Andes 2009], où l'apprenant choisit un thème, puis sélectionne un problème, parmi ceux associés à ce thème, d'après son énoncé. Sur la Figure 1-17, l'écran de gauche contient la liste des thèmes proposés à l'apprenant et l'écran de droite contient la liste des problèmes associés au thème « Vectors ». Sur ce deuxième écran, l'apprenant peut sélectionner un problème pour avoir un aperçu de son énoncé.

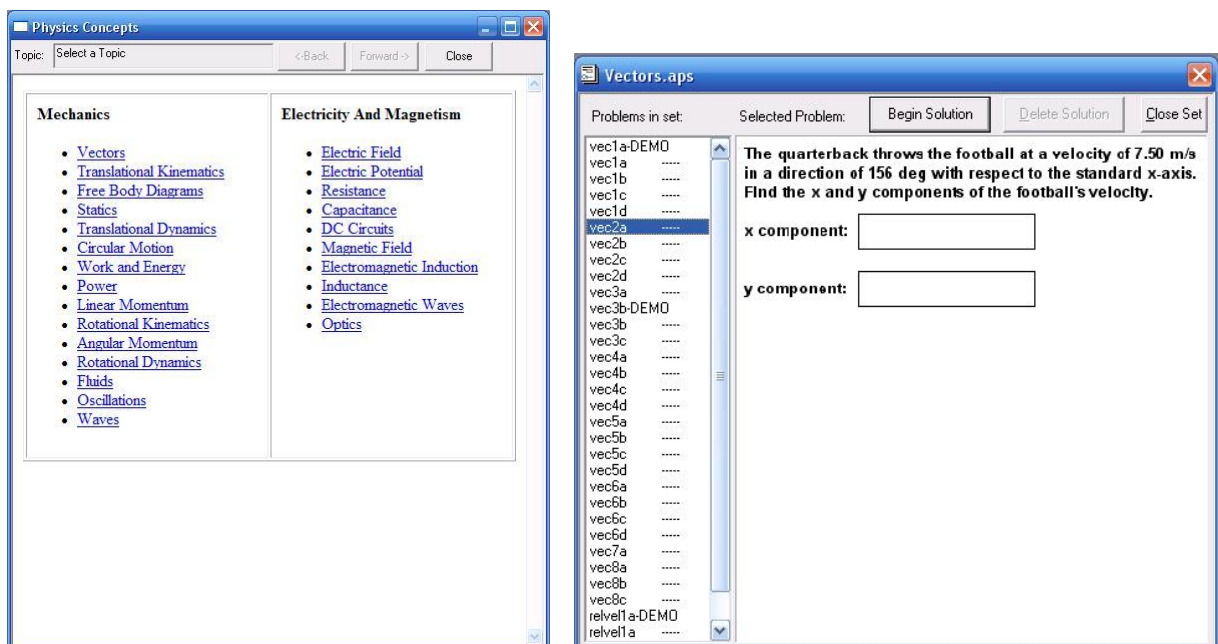


Figure 1-17 : Écrans d'ANDES permettant à l'apprenant de sélectionner le problème qu'il va résoudre [VanLehn et al. 2005].

D'autres systèmes ne laissent à l'apprenant que la possibilité d'un choix partiel concernant les activités. Ainsi, dans les systèmes APLUSIX [Bouhineau et al. 2005b, Bouhineau et al. 2005a] ou LILIMATH [LiliMath 2000], les apprenants choisissent un thème sur lequel ils souhaitent travailler (cf. Figure 1-5 pour avoir un aperçu des thèmes d'APLUSIX et Figure 1-18 pour ceux de LILIMATH) et le système génère une série d'exercices sur ce thème.

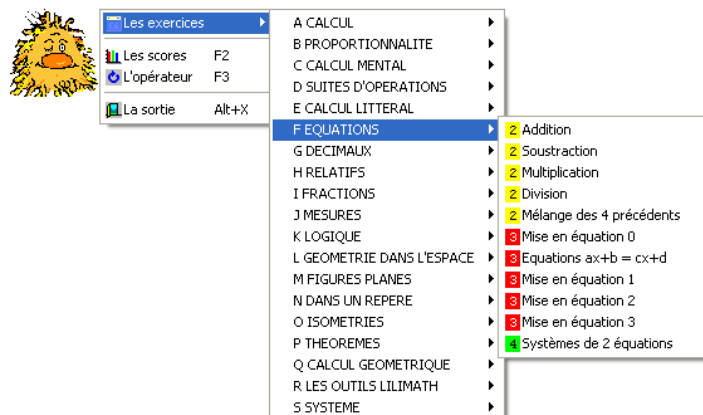


Figure 1-18 : Menu de LILIMATH permettant à l'apprenant d'obtenir une série d'exercices en fonction d'un thème [LiliMath 2000].

De la même façon, sur le site web ACTIVEMATH [Melis et al. 2001, ActiveMath 2007], l'apprenant choisit un thème à étudier (cf. partie gauche de la Figure 1-19) pour avoir accès au cours, ainsi qu'aux exercices prédéfinis associés (cf. partie droite de la Figure 1-19).

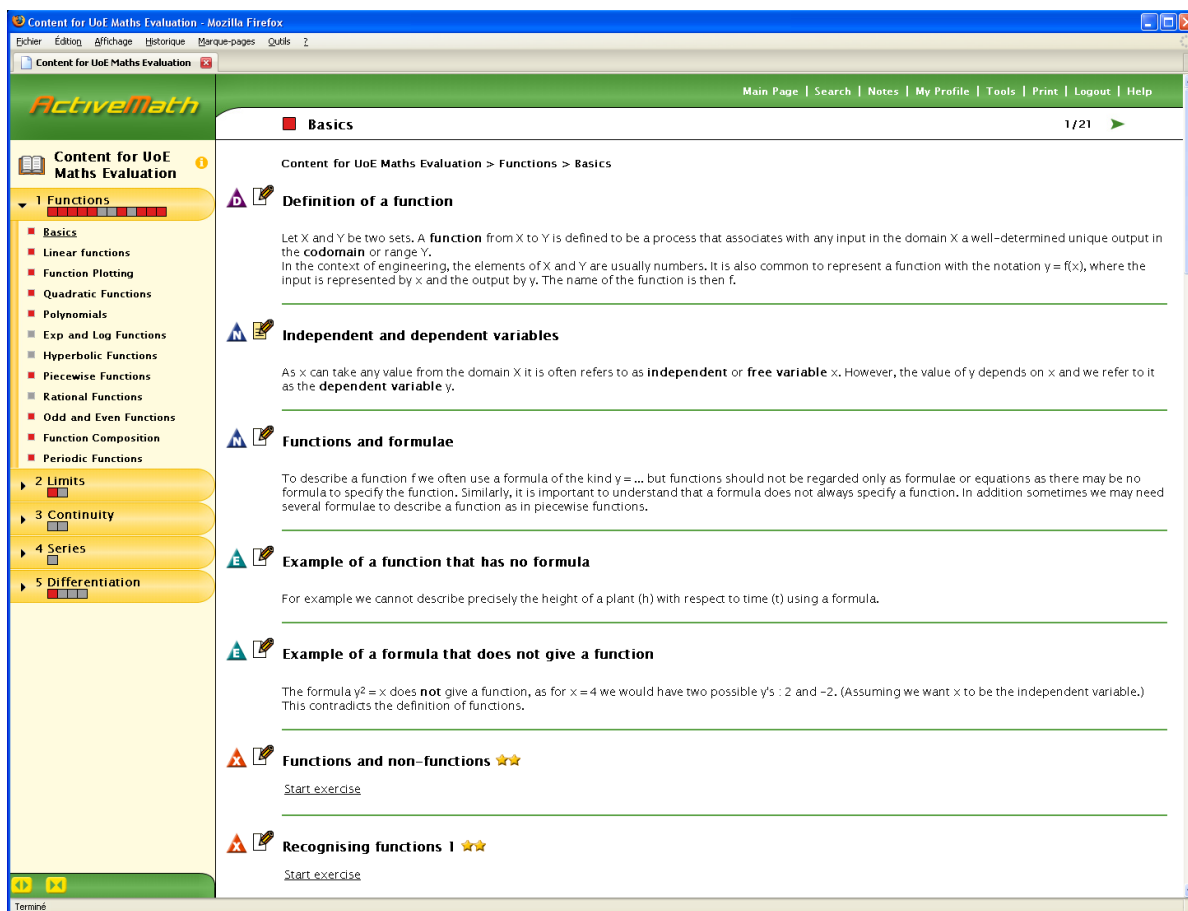


Figure 1-19 : Interface d'ACTIVEMATH avec sur la gauche la liste des thèmes disponibles et sur la droite le cours et les exercices associés au thème choisi par l'apprenant [ActiveMath 2007].

B. PERSONNALISATION PAR L'ENSEIGNANT

Avant de présenter les approches permettant aux enseignants de personnaliser un EIAH, il est important de rappeler que les enseignants n'ont pas toujours eu un rôle prévu lors de l'utilisation des logiciels pédagogiques. En France, les premiers tutoriels installés dans les écoles dans les années 1970-80 réduisaient l'intervention des enseignants à la détermination de la durée des séances, des délais de réflexion laissés pour les réponses et du temps de pause entre les questions. Ceci a eu pour conséquence que, vingt ans plus tard, encore un tiers des enseignants étaient réfractaires à l'intégration des EIAH dans leurs cours [Burnier 2000]. Le désintérêt des enseignants vis-à-vis des EIAH est accentué par deux facteurs : leur manque d'expérience concernant l'utilisation des nouvelles technologies et le manque d'adéquation entre le contenu pédagogique du logiciel et les attentes des enseignants [Girard et al. 2007]. Suite à ce constat d'échec, les chercheurs en EIAH se sont attachés à créer des EIAH non seulement utiles pour l'apprentissage d'une discipline, mais également intégrables aux méthodes d'apprentissage traditionnelles des enseignants. Cette démarche nécessite d'attribuer un rôle à l'enseignant au sein des EIAH. Pour cela, plusieurs stratégies sont possibles : intégrer des enseignants lors de la conception des logiciels, développer des outils permettant aux enseignants de créer eux-mêmes leurs logiciels pédagogiques ou développer une partie des logiciels spécifiquement pour l'enseignant [Jean-Daubias 2004].

La première stratégie, qui consiste à prendre en compte les enseignants lors de la conception, est essentielle, mais elle ne leur permet pas de personnaliser par la suite les logiciels créés, sauf s'ils possèdent un partie destinée à l'enseignant. Nous ne nous attarderons donc pas sur cette stratégie.

La seconde stratégie consiste à développer des outils auteurs permettant aux enseignants de créer leurs logiciels. Des outils auteurs ont déjà été présentés dans la section 1.3.3, mais certains systèmes permettent de surcroît de prédéfinir plusieurs stratégies pédagogiques et de les appliquer selon les performances de l'apprenant.

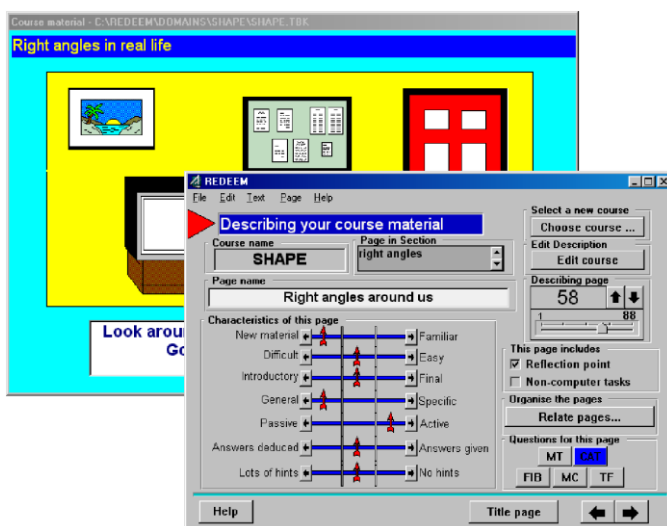


Figure 1-20 : Interface de description du contenu pédagogique à utiliser dans REDEEM [Ainsworth 2000].

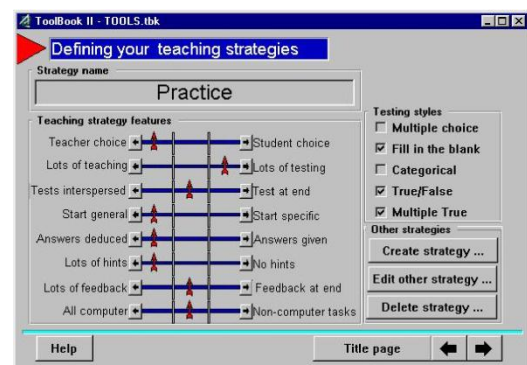


Figure 1-21 : Interface de définition d'une stratégie pédagogique dans REDEEM [Ainsworth 2000].

Ainsi, le système REDEEM [Ainsworth 2000] permet aux enseignants d'intégrer le matériel pédagogique créé par un autre outil auteur afin de l'utiliser avec leurs stratégies pédagogiques. Pour cela, un enseignant décrit le matériel pédagogique qu'il souhaite utiliser (cf. Figure 1-20), forme des groupes avec ses apprenants, puis définit les stratégies pédagogiques qu'il veut

mettre en place pour que le matériel pédagogique soit proposé de façon adéquate à chaque groupe d'apprenants (cf. Figure 1-21).

Le système EON [Murray 2003a] permet à l'enseignant de définir des méta-stratégies afin de choisir la stratégie pédagogique appropriée pour enseigner un type de connaissances. Chaque méta-stratégie est une règle « si-alors » avec pour prémisse une condition d'application et pour conclusion, des valeurs pour les paramètres du système. La Figure 1-22 montre un exemple de méta-stratégie définie par un enseignant. Dans cet exemple, si la variable du modèle de l'apprenant « RecentWrong » a une valeur supérieure à 50%, c'est-à-dire si l'étudiant se trompe une fois sur deux, alors le nombre de conseils sera de deux, le niveau de difficulté sera croissant, etc.

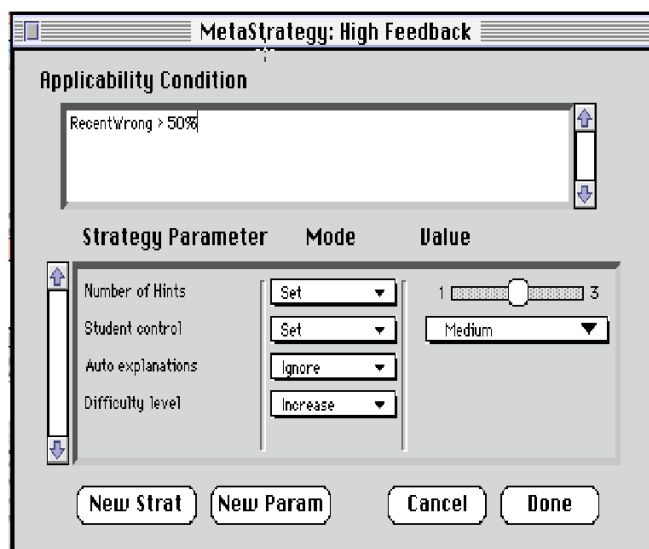


Figure 1-22 : Éditeur de méta-stratégies dans EON [Murray 2003a].

La troisième stratégie pour attribuer un rôle à l'enseignant au sein des EIAH est de leur réserver une partie de l'EIAH. Nous allons à présent voir les différentes possibilités offertes aux enseignants pour adapter les EIAH possédant une partie qui leur est consacrée.

Une première possibilité réside dans la personnalisation des buts à atteindre par les apprenants. Dans DIVIDINGQUEST [Girard et al. 2007], l'enseignant fournit pour chaque apprenant son niveau habituel concernant l'apprentissage des divisions. Le système classe, à partir de cette information, l'apprenant dans un groupe auquel correspondent des buts pédagogiques à atteindre. Ces buts pédagogiques peuvent être modifiés par l'enseignant.

Une seconde possibilité consiste à permettre à l'enseignant d'agir directement sur les activités fournies aux apprenants et sur l'ordre dans lequel elles sont proposées aux apprenants. Ainsi, dans APLUSIX [Nicaud et al. 2003], l'enseignant peut créer des exercices, pour ensuite demander à ses élèves de les faire. Dans AMBRE-ENSEIGNANT [Duclosson et al. 2005b, Jean-Daubias et al. 2009b], l'enseignant crée des problèmes semi-automatiquement, c'est-à-dire en contraignant la génération des problèmes (cf. Figure 1-9), puis définit, pour chaque apprenant, des séquences contenant les exercices qu'il souhaite que l'élève résolve et leur ordre. Enfin, dans ROBOTTEACH [Leroux 2002, Leroux 2006], l'enseignant définit une suite d'activités à faire grâce à une interface de gestion de séquences pédagogiques. Il peut compléter cette suite d'activités de créneaux où l'enseignant devra être appelé par l'apprenant pour interagir sur les activités passées ou à venir.

Une dernière possibilité permet à l'enseignant d'agir sur les fonctionnalités accessibles à l'apprenant, ainsi que sur l'interface du logiciel.

C. PERSONNALISATION PAR LE SYSTÈME

La personnalisation d'un EIAH peut être faite automatiquement par celui-ci afin que son contenu soit adapté à chaque apprenant. Cette personnalisation peut se faire de plusieurs manières, mais exploite toujours les informations que le système possède sur l'apprenant.

Les informations sur l'apprenant peuvent être représentées par des stéréotypes auxquels les apprenants sont associés [Girard et al. 2007] ou dans un modèle de l'apprenant contenu dans l'EIAH. Le modèle de l'apprenant peut être créé par l'apprenant lui-même, comme dans CREEK-TUTOR [Sørmo et al. 2002, Aamodt 2005], où l'apprenant dessine la carte des concepts qu'il possède sur un sujet particulier, ou calculé par le système, comme dans SQL-TUTOR [Mitrovic 1998].

Lorsqu'il est calculé par le système, le modèle de l'apprenant peut contenir plusieurs types d'informations. Tout d'abord, les informations peuvent être, tout simplement, les réponses de l'apprenant. Ainsi, le système IDEBUGGY [Burton 1982] génère un nouveau problème après chaque erreur de l'apprenant. De façon analogue, APPELEC [Guéraud et al. 2007] contient 22 activités dont l'enchaînement est géré en fonction des réponses fournies progressivement par l'élève. Un autre type d'informations sur l'apprenant peut être le comportement de celui-ci lorsqu'il interagit avec le logiciel. Ainsi, le didacticiel ESCARRE [Hofmann 1990] s'adapte aux réactions de l'apprenant. De même, le simulateur TELEOS [Teleos 2003, Mufti-Alchawafa et al. 2004] interprète le comportement de l'apprenant pendant le processus de résolution des problèmes pour lui fournir un feedback approprié. Enfin, les informations sur l'apprenant peuvent être relatives à ses connaissances ou compétences. Ainsi, le logiciel MOREMATH [Bull et al. 2003] construit un modèle des connaissances de l'apprenant pour, entre autres, lui fournir une séquence de révision adaptée (contenant des cours et des exercices). Les modèles de l'apprenant peuvent contenir d'autres informations (comme les préférences de l'apprenant), nous avons présenté uniquement les trois types d'informations les plus couramment utilisées.

D. SYNTHÈSE SUR LES ACTEURS DE LA PERSONNALISATION

Dans cette section, nous indiquons les avantages et les inconvénients de la personnalisation par chacun des acteurs.

Tout d'abord, lorsque l'apprenant personnalise lui-même un EIAH, cela lui permet de se l'approprier pleinement et de choisir les thèmes qu'il souhaite aborder. Il peut ainsi, selon les cas, choisir un cours à suivre, un exercice à résoudre ou un thème à aborder. L'inconvénient de cette approche est qu'elle est difficilement intégrable dans le cadre d'un enseignement en classe. Un enseignant ne pourra pas contrôler ce que fait chaque apprenant, et ainsi être sûr que celui-ci fera les activités nécessaires pour maîtriser une notion qu'il souhaite lui faire acquérir.

À l'inverse, lorsque l'enseignant personnalise les EIAH avant qu'ils soient utilisés par l'apprenant, il peut maîtriser le contenu du logiciel pour chaque élève ou groupe d'élèves. Le problème est que cette personnalisation n'est possible que sur un nombre restreint de systèmes. De plus, dans la majorité des EIAH concernés, aucune aide n'est fournie à l'enseignant sur la personnalisation à mettre en place pour un élève donné, en exploitant par exemple les réponses de l'apprenant ou un modèle des connaissances de celui-ci.

Enfin, lorsque la personnalisation de l'EIAH est faite automatiquement par le système, elle l'est en fonction de la connaissance que le système a de l'apprenant, mais elle n'est pas forcément adaptée aux buts pédagogiques des enseignants.

1.5. SCÉNARISER LES SÉANCES D'APPRENTISSAGE

Pour aider l'enseignant dans sa tâche de personnalisation, une dernière approche propose des modèles et des outils permettant à l'enseignant de définir des *scénarios*. Nous allons définir le concept de scénario avant de présenter les principaux standards permettant la définition et l'échange de scénarios.

1.5.1. DÉFINITION DU CONCEPT DE SCÉNARIO

Le concept de scénario complète ou remplace d'autres termes plus couramment utilisés par les enseignants : cours, séquence, situation d'apprentissage [Villiot-Leclercq 2005]. De nombreuses définitions, souvent associées à des déclinaisons différentes du terme scénario (pédagogique, d'apprentissage, d'usage...) ont été proposées pour préciser ce concept. Nous en retiendrons deux pour illustrer ce que nous entendons par scénario :

« Un scénario d'apprentissage représente la description, effectuée *a priori* ou *a posteriori*, du déroulement d'une situation d'apprentissage ou unité d'apprentissage visant l'appropriation d'un ensemble précis de connaissances, en précisant les rôles, les activités, ainsi que les ressources de manipulation de connaissances, outils et services nécessaires à la mise en œuvre des activités. » [Pernin et al. 2004]

« Description plus ou moins formelle d'une séquence d'enseignement définissant les objectifs pédagogiques cibles et les moyens à mettre en œuvre pour atteindre ces objectifs. Un scénario pédagogique décrit généralement les acteurs impliqués (apprenant, enseignant, tuteur, etc.), les ressources pédagogiques (documents, logiciels, etc.), les tâches que les apprenants doivent réaliser, les rôles des différents acteurs et les contraintes à respecter. » [Tchounikine 2009]

Ces définitions générales d'un scénario peuvent être spécifiées en fonction des buts ou des propriétés du scénario. Ainsi, [Pernin et al. 2004] distinguent scénario prédictif et scénario descriptif. Un scénario *prédictif* est établi *a priori* par un concepteur en vue de la mise en place d'une situation d'apprentissage, instrumentée ou non par les technologies numériques, tandis qu'un scénario *descriptif* décrit *a posteriori* le déroulement effectif d'une situation d'apprentissage, en y incluant en particulier les traces de l'activité des acteurs et leurs productions.

De plus, un scénario prédictif peut être générique ou adaptatif. Un scénario *générique* sera toujours exécuté de manière identique, tandis qu'un scénario *adaptatif* prend en compte des profils d'apprenants permettant l'exécution conditionnelle de plusieurs scénarios personnalisés se distinguant par la nature des interactions proposées (rétroactions, parcours, etc.) ou par la nature des ressources de manipulation de connaissances mises à disposition.

Quelles que soient les propriétés du scénario, il y a une séparation entre les ressources manipulées par les élèves (exercices, cours, logiciel, etc.) et les activités qui leur sont proposées sur ces ressources. Un scénario décrit le déroulement d'une activité et non le contenu des ressources.

1.5.2. CRÉATION ET ÉCHANGE DE SCÉNARIOS

Pour prendre en compte la diversité des approches pédagogiques et permettre l'échange et l'interopérabilité des scénarios, plusieurs standards ont été proposés. Les modèles EML (Educational Modelling Language) [Koper 2001] et IMS-LD (IMS Learning Design) [IMS-LD] permettent ainsi de définir les relations entre les objectifs en termes de connaissances ou d'habiletés, les acteurs de l'apprentissage, les activités réalisées, ainsi que l'environnement et les contenus nécessaires à la mise en place de la situation d'apprentissage.

Utilisant une métaphore théâtrale, IMS-LD définit la structure d'une unité d'apprentissage comme un ensemble d'actes composés de partitions associant des activités à des rôles. Une activité est située dans un environnement incluant des services et des ressources de contenu décrites à l'aide du standard LOM [LOM 2002].

Le projet européen ICLASS [iClass, Türker et al. 2006] est un exemple d'utilisation du standard IMS-LD. Le but de ce projet est de fournir un environnement d'apprentissage en ligne adapté à chaque apprenant, ainsi que des services pédagogiques à destination de tous les acteurs concernés : étudiants, professeurs, administrations scolaires, parents d'élèves, mais aussi maisons d'édition, fournisseurs de contenus, etc. Le processus de personnalisation se fait en quatre étapes : la modélisation de l'apprenant, le choix d'un scénario adapté à l'apprenant, l'instanciation du scénario avec des ressources et la présentation du scénario adapté à l'apprenant. Concernant le choix du scénario, l'environnement contient des scénarios génériques définis selon le format IMS-LD. Le choix du scénario se fait en fonction du modèle de l'apprenant, du domaine étudié, des préférences de l'enseignant et du contexte. Une fois le scénario choisi, il est complété avec des ressources générées ou sélectionnées dans les bases d'objets pédagogiques [Brady et al. 2005].

Ces différents standards (EML, IMS-LD, LOM) partent du principe que les théories d'apprentissage reflétant des visions pédagogiques différentes peuvent être décrites grâce à des modèles neutres pédagogiquement. Pour cela, il est nécessaire de produire des objets d'apprentissage décontextualisés avant de les réutiliser, en y faisant référence, dans des scénarios d'apprentissage divers. La limite de ces standards est que certaines situations d'apprentissage, comme celles relevant de la théorie constructiviste de l'éducation, ne peuvent pas être précisément définies en utilisant ces standards [Nodenot 2006].

1.6. SYNTHÈSE DES APPROCHES RELATIVES À LA PERSONNALISATION DE L'APPRENTISSAGE

Dans la littérature, les approches proposées dans le cadre de la personnalisation de l'apprentissage traitent de questions de recherche précises mais différentes. Elles ont chacune des forces et des faiblesses. C'est pourquoi nous proposons de discuter des limites des approches existantes en considérant trois critères d'analyse liés aux trois points de notre problématique de recherche : la prise en compte des individualités des apprenants, la prise en compte des besoins et habitudes pédagogiques des enseignants et l'affectation d'une activité à un apprenant.

La prise en compte des individualités des apprenants peut se faire à travers l'utilisation de stéréotypes [Vincent et al. 2005, Girard et al. 2007] ou de profils d'apprenants [Mitrovic 1998, Sørmo et al. 2002].

Un stéréotype contient un ensemble de caractéristiques que l'on trouve chez plusieurs apprenants. Cette factorisation de la représentation ne permet pas une prise en compte fine des individualités des apprenants. À l'inverse, un profil d'apprenant est spécifique à un individu. Il encode plus finement les informations déduites sur les apprenants suites aux différentes activités pédagogiques qu'ils ont réalisées. Néanmoins, le processus de construction des profils d'apprenants n'est pas sans risque et des erreurs de diagnostic peuvent conduire à la prise en compte d'informations erronées.

Dans les deux cas, toute la difficulté consiste à capturer les données pertinentes concernant les apprenants et à donner les moyens à l'enseignant ou au système mettant en œuvre la personnalisation d'accéder facilement à ces données.

La prise en compte des besoins et habitudes pédagogiques des enseignants est possible lorsque que ceux-ci utilisent des outils auteurs [David et al. 1996, Van Joolingen et al. 2003], définissent des scénarios pédagogiques [iClass, Pernin et al. 2004] ou paramètrent eux-mêmes les logiciels pédagogiques [Murray 2003a, Duclosson et al. 2005a].

Les outils auteurs existants permettent de créer des ressources (exercices, feuilles d'exercices ou logiciels) adaptées à chaque enseignant. Pour cela, l'enseignant doit fournir entièrement le contenu pédagogique. L'utilisation d'outils auteurs est donc coûteuse en temps.

Les scénarios pédagogiques permettent de définir les activités proposées aux apprenants, en précisant le contexte dans lequel ils se trouvent, les rôles de tous les participants, les actions à accomplir sur les ressources. Les outils permettant de définir un scénario ne permettent cependant pas de créer des ressources. Ils permettent d'associer des ressources existantes qui sont soit contenues dans l'application sur laquelle le scénario sera mis en œuvre, soit créées manuellement par l'enseignant, soit créées en utilisant une application spécifique.

Certains logiciels pédagogiques possèdent une partie spécifique permettant à l'enseignant de paramétrer l'environnement proposé à l'apprenant (contenu pédagogique et/ou interface). Le problème de cette approche est que peu de logiciels possèdent une partie réservée à l'enseignant. De plus, ces parties sont différentes d'un logiciel à l'autre. Ainsi, un enseignant voulant utiliser plusieurs systèmes devra maîtriser les différents outils de paramétrage.

L'hétérogénéité des systèmes constitue en elle-même une limite du point de vue de l'enseignant. En effet, pour adapter des activités de diverses provenances (issues de générateurs, décrites dans des scénarios, contenues dans des logiciels, etc.) à ses buts pédagogiques, l'enseignant doit apprendre à utiliser l'interface de nombreux outils. De plus, il n'existe pas d'approche unifiée permettant d'assister l'enseignant dans cette tâche d'adaptation des activités.

L'affectation d'une activité à un apprenant peut être faite automatiquement par le système [Burton 1982, Mitrovic 1998], ou manuellement soit par l'apprenant [Melis et al. 2001, VanLehn et al. 2005], soit par l'enseignant [Leroux 2002, Duclosson et al. 2005a].

Dans le cas où le système s'adapte automatiquement à l'apprenant, la personnalisation est faite en fonction de la connaissance que le système a de l'apprenant, mais elle n'est pas forcément adaptée aux buts pédagogiques des enseignants. De même, dans le cas où l'apprenant régule lui-même son apprentissage, il est difficile pour un enseignant de le contraindre à respecter ses méthodes de travail.

Dans le cas où l'enseignant personnalise les logiciels, nous venons de voir qu'il n'a pas facilement accès aux informations relatives à l'apprenant. Cette personnalisation est donc souvent longue à mettre en place.

D'une manière générale, les choix pédagogiques intervenant dans la personnalisation, et donc dans l'affectation d'une activité à un apprenant, doivent être faits par l'enseignant si l'on souhaite que la personnalisation mise en œuvre corresponde à ses buts et besoins pédagogiques. Or, rares sont les systèmes qui permettent de prendre en compte ces besoins.

Notre analyse de l'existant ne nous a pas permis de trouver d'approche permettant de répondre complètement à notre problématique : quel processus générique peut permettre de personnaliser les activités pédagogiques proposées aux apprenants, à partir de leurs profils, tout en respectant les spécificités du contexte pédagogique ?

L'approche que nous proposons consiste donc à fournir un outil unique permettant :

- d'avoir accès facilement aux informations relatives aux apprenants à travers des profils d'apprenants ;
- de paramétrer les activités¹ proposées pour qu'elles correspondent aux besoins de chaque enseignant ;
- d'intégrer les choix pédagogiques de chaque enseignant en matière d'affectation d'activités aux apprenants.

Ces activités que nous proposons de personnaliser peuvent être des activités papier, mais aussi des activités contenues dans un logiciel pédagogique. Pour permettre à l'enseignant d'adapter les activités logicielles, il est nécessaire de pouvoir agir sur les logiciels de manière externalisée, mais également de connaître leurs possibilités de paramétrage. L'étude de l'existant relatif à cet aspect est traitée dans le chapitre suivant.

¹ Concernant le paramétrage d'activités, il faut noter que ce que nous appelons *activité* correspond à ce que les travaux sur les scénarios pédagogiques qualifient de ressources.

CHAPITRE 2. COMMENT PILOTER UN
SYSTÈME INFORMATIQUE DE FAÇON
EXTERNALISÉE ? ET PLUS
PARTICULIÈREMENT UN EIAH ?

PLAN DU CHAPITRE

2.1.	Introduction.....	61
2.2.	Piloter un système informatique de façon externalisée	61
2.2.1.	Approche des systèmes épiphytes.....	61
2.2.2.	Approche par composants	62
2.2.3.	Bilan.....	63
2.3.	Décrire un EIAH à l'aide de métadonnées	63
2.3.1.	Standards orientés description de contenus	63
a.	Dublin core	63
b.	LOM.....	64
c.	SCORM	66
d.	Synthèse sur les standards.....	67
2.3.2.	LSCM - Schéma de description des composants logiciels.....	67
2.3.3.	Synthèse des approches relatives à la description d'EIAH	70

2.1. INTRODUCTION

L'approche que nous souhaitons mettre en place pour permettre à un enseignant de personnaliser à la fois des activités papier et des logiciels pédagogiques nécessite, entre autres, de pouvoir agir sur les logiciels pédagogiques de façon externalisée et de connaître leurs possibilités de paramétrage pour les proposer à l'enseignant de manière uniformisée.

Dans la première partie de ce chapitre, nous nous intéressons aux différentes approches permettant un pilotage externalisé de logiciels : l'approche des systèmes épiphytes et l'approche par composants. Nous expliquons le principe de chacune avant de mettre en évidence ce qui ne convient pas dans ces méthodes pour permettre une personnalisation comme nous l'entendons.

Dans la seconde partie de ce chapitre, nous regardons les normes et standards proposés pour décrire du matériel pédagogique afin de savoir s'ils permettent de décrire un EIAH dans un but de personnalisation externalisée.

2.2. PILOTER UN SYSTÈME INFORMATIQUE DE FAÇON EXTERNALISÉE

Dans cette partie, nous présentons les deux approches permettant à un système d'observer et/ou d'agir sur un autre système.

2.2.1. APPROCHE DES SYSTÈMES ÉPIPHYTES

Le terme épiphyte, issu de la botanique, est utilisé en informatique pour décrire des systèmes capables de se greffer sur une application sans en troubler le fonctionnement normal [Paquette et al. 1994, Paquette et al. 1996b].

Un système épiphyte est donc un système qui raisonne et agit à partir des interactions observées entre les différentes composantes de l'application hôte. Les systèmes épiphytes et leurs hôtes sont indépendants du point de vue conceptuel (leurs architectures sont indépendantes) et du point de vue logiciel (les hôtes ne sont aucunement conçus en fonction du système épiphyte) [Giroux et al. 1995].

Cette approche nécessite de développer, pour chaque hôte, un système particulier. Pour des hôtes de même nature, les systèmes épiphytes pourront avoir la même architecture, mais leurs contenus différeront pour pouvoir observer chaque hôte et le piloter.

Cette observation a permis la création de la plateforme EPI-TALK [Paquette et al. 1996b] qui permet de décrire, de générer et de gérer des systèmes conseillers. Nous rappelons qu'un système conseiller est un système informatique qui produit des conseils à partir de l'observation de l'interaction entre un utilisateur et un environnement d'apprentissage [Paquette et al. 1994]. EPI-TALK a ainsi permis de créer les systèmes conseillers de plusieurs applications, comme ADISA [Paquette et al. 1993a], un atelier de génie didactique, COPERNIC-2 [Paquette 1992], un environnement d'apprentissage sur la démarche scientifique, et HYPERGUIDE [Paquette et al. 1993b], un environnement pour la formation à distance. Ces systèmes conseillers se greffent à un environnement d'apprentissage ou à un système d'aide à la tâche existant sans en perturber le fonctionnement. Ils respectent donc bien la définition d'un système épiphyte.

Le tuteur PACT est un autre exemple de système épiphyte [Ritter et al. 1995] (cité dans [Macrelle-Rosselle 2001]) pouvant être greffé sur un logiciel, éducatif ou non, afin d'utiliser l'ensemble comme outil éducatif. Ce tuteur a été ainsi utilisé avec MICROSOFT EXCEL et GEOMETER'S SKETCHPAD [Ritter et al. 1996].

Plus récemment, l'environnement SAAFIR a été proposé [Rasseneur et al. 2003, Rasseneur-Coffinet 2004] afin de se greffer aux dispositifs informatiques de formation existants pour aider l'apprenant dans son processus d'appropriation de la formation.

2.2.2. APPROCHE PAR COMPOSANTS

L'approche par composants considère les logiciels existants comme des composants pouvant être combinés dans un dispositif unique afin de créer un nouveau logiciel.

Dans le cadre des EIAH, cette approche permet d'utiliser dans un environnement donné des fonctionnalités complémentaires issues de systèmes différents pour construire de nouvelles activités pédagogiques [Rosselle 2003]. L'avantage d'une telle démarche est d'éviter de redévelopper toutes les fonctionnalités d'un EIAH pour chaque nouveau système, car elle permet d'intégrer et d'utiliser des composants existants [Futtersack et al. 2000].

Pour définir le concept de composant, nous donnons la définition de [Allen et al. 1998], traduite par [Oubahssi 2005] : « un composant est une unité exécutable ayant la forme d'une boîte noire encapsulant les services qu'elle fournit. Ces services ne sont accessibles que par les interfaces publiées correspondantes et ce à travers un standard d'interaction ».

Afin de faciliter cette approche dans le cadre des EIAH, [Rosselle et al. 2004, Rosselle et al. 2005] ont proposé des recommandations à suivre pour le développement de composants. Chaque composant doit :

- être *inspectable* pour permettre l'observation de certains de ses mécanismes, états, objets, etc. ;
- être *traçable* pour fournir les traces intelligibles de l'interaction de l'utilisateur avec le composant ;
- être *scriptable* pour permettre au dispositif pilotant le composant de l'activer, totalement ou partiellement et, par ailleurs, permettre au dispositif de récupérer seulement les informations dont il a besoin (fichiers de sortie, traces d'interaction, etc.) ;
- être *indexable* afin de permettre aux concepteurs de nouveaux dispositifs de retrouver le composant à travers les fonctionnalités qu'il possède ;
- être *intégrable au niveau plastique* pour pouvoir exporter son interface graphique, permettre son adaptation ou sa reprogrammation ;
- être *décomposable en fonctionnalités indépendantes ou élémentaires* pour permettre au dispositif le pilotant d'accéder indépendamment à ses différentes fonctionnalités ;
- utiliser des *formats normalisés*, des standards pour assurer la communication des données dans un format adéquat.

Le dispositif gérant ces composants doit posséder les mêmes propriétés qu'un composant, mais il doit également, entre autres, intégrer un langage de commande des composants, gérer les formats de données et une base d'informations sur les composants pour indexer les fonctionnalités intéressantes.

Ces recommandations se retrouvent dans les travaux d'autres chercheurs. Par exemple, [Guéraud et al. 2007] notent que pour qu'un logiciel pédagogique soit contrôlable par un

scénario pédagogique, il est nécessaire que le logiciel soit *inspectable* et *scriptable*, c'est-à-dire que les valeurs de ses variables soient consultables et modifiables depuis l'extérieur. De même, [Ritter et al. 1998] (cités dans [Murray 1999]) indiquaient déjà que pour mettre en place une approche par composants, il est nécessaire d'avoir des modules réutilisables et d'utiliser des protocoles de communication standards.

L'approche par composants nécessite donc d'indexer les EIAH avec des métadonnées qui doivent porter sur les aspects techniques et sur l'usage pédagogique de l'EIAH, elle demande aussi de permettre une communication des données et des modèles entre chaque composant.

2.2.3. BILAN

Dans le cadre de nos travaux de recherche, nous retenons de ces approches le besoin d'informations sur le système à piloter. Mais contrairement à l'approche des systèmes épiphytes qui propose un système de pilotage en temps réel, et à l'approche par composants qui propose un système combinant plusieurs logiciels, notre approche vise à fournir un système paramétrant les logiciels préalablement à leur utilisation autonome par un apprenant.

2.3. DÉCRIRE UN EIAH À L'AIDE DE MÉTADONNÉES

La question de la description des EIAH s'est historiquement très vite posée pour permettre le partage et la réutilisation des ressources qui les composent, comme le rapportent [Grandbastien et al. 2002]. Plusieurs propositions de normalisation ont été faites avec des objectifs variés. Dans cette section, nous présentons tout d'abord des normes et standards orientés description de contenus, puis un schéma permettant de décrire les composants logiciels avant de conclure sur leurs possibles utilisations afin de décrire un EIAH dans un but de personnalisation.

2.3.1. STANDARDS ORIENTÉS DESCRIPTION DE CONTENUS

Pour décrire un EIAH, une approche consiste à utiliser des normes ou standards orientés description de contenus. Parmi ceux-ci, le Dublin Core comporte un jeu de métadonnées généralistes permettant de décrire tous types de ressources [DublinCore 1995, Keenoy 2003], alors que d'autres standards sont plus spécifiques à des domaines ou métiers. Concernant les ressources éducatives, nous pouvons citer le LOM qui permet de décrire les outils éducatifs [LOM 2002], notamment les logiciels d'e-learning, ou SCORM qui permet de créer des objets pédagogiques structurés, interopérables et réutilisables [SCORM 2001]. Ces trois standards sont les mieux adaptés au domaine des EIAH. Nous les présentons ici avant d'indiquer leurs limites vis-à-vis de la description d'un logiciel pédagogique dans un but de personnalisation externalisée.

A. DUBLIN CORE

Le Dublin Core est une norme établie par un consensus international de professionnels issus de diverses disciplines, visant à proposer un modèle de document assurant une interopérabilité à l'échelle internationale et décrivant une grande variété de ressources documentaires. Il comprend 15 éléments, tous optionnels. La Figure 2-1 présente ces différents éléments en donnant pour chacun une définition.

	Élément	Définition
1	Title	Nom donné à la ressource.
2	Creator	Personne ou système principalement responsable de la création du contenu de la ressource.
3	Subject	Thème de la ressource.
4	Description	Description textuelle sur le contenu de la ressource.
5	Publisher	Personne ou société responsable de la diffusion de la ressource.
6	Contributor	Personne ou système ayant contribué à la création de la ressource.
7	Date	Date associée à un événement du cycle de vie de la ressource (création ou publication).
8	Type	Nature du contenu de la ressource.
9	Format	Matérialisation physique ou numérique de la ressource.
10	Identifier	Référence non ambiguë de la ressource dans un contexte donné.
11	Source	Référence à une ressource à partir de laquelle cette ressource est dérivée.
12	Language	Langage du contenu de la ressource.
13	Relation	Référence à une ressource en liaison avec cette ressource.
14	Coverage	Portée ou couverture spatio-temporelle de la ressource.
15	Rights	Information sur les droits sur et au sujet de la ressource.

Figure 2-1 : Métadonnées du Dublin Core [DublinCoreFr 2000].

Le Dublin Core sert de base à tous les autres standards permettant la description des ressources numériques. Il contient les métadonnées nécessaires à l'identification et la description générale d'une ressource, mais aucune de ces métadonnées ne permet de décrire ni l'usage qui peut être fait d'un logiciel pédagogique, ni les propriétés techniques de celui-ci permettant de le paramétrer.

B. LOM

Le LOM (Learning Object Metadata) est un standard issu du projet de l'IEEE-LTSC visant à proposer un modèle standard de métadonnées permettant de décrire et référencer tout document pédagogique numérique avec pour objectif la réutilisation des objets pédagogiques par les apprenants, les enseignants ou les processus logiciels automatisés [De la Passardière et al. 2003]. Ce standard se veut général et applicable à un très grand nombre de situations éducatives en répondant à la fois aux besoins des utilisateurs et aux contraintes des producteurs de ressources.

Le LOM reprend des éléments du standard Dublin Core et contient des extensions propres au domaine éducatif. Il se limite à un ensemble de caractéristiques jugées indispensables pour gérer les objets pédagogiques. Il comporte 80 métadonnées, toutes facultatives, classées selon neuf catégories présentées dans la Figure 2-2.

	Catégorie	Description
1	General	Contient les caractéristiques globales de la ressource : identifiant, titre, langue utilisée pour communiquer avec l'utilisateur, mots-clés, etc.
2	Life Cycle	Contient les caractéristiques relatives à l'historique et à l'état courant de la ressource (informations sur la version, statut, personnes qui l'ont modifié, à quelle date, etc.).
3	Meta-Metadata	Décrit le(s) schéma(s) ou la(les) spécification(s) utilisé(s).
4	Technical	Contient les caractéristiques et exigences techniques liées à l'exploitation de la ressource : format de l'objet pédagogique, taille, besoins logiciels pour l'utiliser (navigateur, système d'exploitation...), remarques sur son installation, etc.
5	Educational	Décrit les caractéristiques pédagogiques de la ressource à travers ses conditions d'utilisation : comment la ressource doit être utilisée, quel est son type (exercice, figure...), son niveau d'interactivité, à qui s'adresse la ressource (apprenant, enseignant, auteur...), quel est le contexte (université, formation professionnelle, école primaire...) ou la tranche d'âge à laquelle s'adresse la ressource, etc.
6	Rights	Spécifie les informations relatives à la propriété intellectuelle, aux droits d'usages de la ressource pédagogique (copyright) et ses conditions d'utilisation (coût).
7	Relation	Décrit les liens entre la ressource et d'autres objets pédagogiques, en spécifiant différents niveaux d'agrégation grâce aux relations d'appartenance ou de composition.
8	Annotation	Comporte un ensemble de commentaires relatifs à l'utilisation pédagogique de la ressource, avec pour chacun les détails relatifs à l'auteur et à la date de création du commentaire.
9	Classification	Indique l'appartenance de la ressource à une ou plusieurs instances de classification en précisant l'objet de la classification (discipline, pré-requis, objectif pédagogique, restriction d'accès, niveau d'enseignement, niveau d'habileté, niveau de sécurité,...), la taxonomie liée à la classification (nom du système de classification, chemin taxonomique), la description, ainsi que les mots-clés de la ressource vis-à-vis du critère de classification.

Figure 2-2 : Catégories du LOM [LOM 2002].

La Figure 2-3, extraite de [De la Passardièrre et al. 2004], donne un aperçu global des métadonnées du LOM, en les regroupant selon les neuf catégories détaillées dans la Figure 2-2.

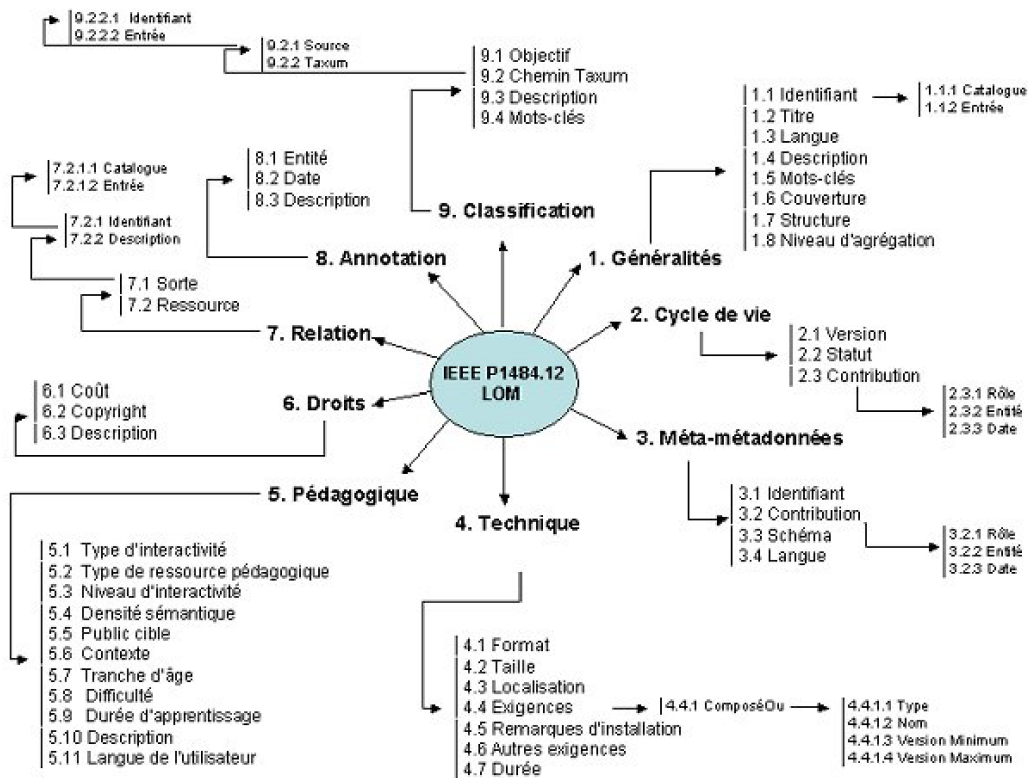


Figure 2-3 : Métadonnées du LOM [De la Passardièrre et al. 2004].

Plusieurs acteurs de la normalisation ont émis des critiques ou noté des insuffisances à propos du LOM [Bourda 2001, Pernin 2003, De la Passardièrre et al. 2004, Delestre et al. 2004], dont la principale porte sur l'ambiguïté du modèle. Cette ambiguïté concerne les termes choisis pour les métadonnées ainsi que leur définition.

Dans le cadre de notre recherche, nous souhaitons décrire des logiciels afin de les paramétrer. Le LOM a été proposé pour décrire des ressources pédagogiques non modifiables se trouvant dans un fichier. Les métadonnées de la catégorie « Technique » (cf. 4 sur les figures 2-2 et 2-3) ne sont donc logiquement pas adaptées à la description des paramètres d'un logiciel.

Le LOM permet donc de décrire plus précisément que le Dublin Core des ressources pédagogiques grâce à des métadonnées permettant de décrire l'usage pédagogique qui peut être fait d'une ressource. Mais, comme le Dublin Core, il ne contient aucune métadonnée permettant la description du paramétrage des logiciels.

C. SCORM

Le SCORM (Sharable Content Object Reference Model) [SCORM 2001] a pour objectif de promouvoir l'utilisation de l'apprentissage basé sur les technologies et le web en particulier, ainsi que de fournir un modèle de référence permettant de garantir la qualité des contenus en termes de réutilisabilité, d'accessibilité, de pérennité, d'interopérabilité [Pernin 2003].

Le SCORM propose deux types de recommandations pour définir comment construire un contenu de formation : le modèle d'agrégation de contenu et l'environnement d'exécution.

Le modèle d'agrégation de contenu (*Content Aggregation Model*) définit les différents types d'éléments nécessaires à la mise en place d'une formation. Il prend en compte les spécifications du LOM et distingue trois niveaux d'éléments : les ressources numériques élémentaires, les objets pédagogiques partageables et les agrégats de contenu.

Une ressource numérique élémentaire (*Asset*) peut représenter un document (image, son, page web), mais également tout ensemble d'informations pouvant être délivré vers un client Web (document Flash, code JavaScript, applet Java, etc.).

Un objet de contenu partageable (*Sharable Content Object*, noté SCO par la suite) est un ensemble cohérent de ressources numériques élémentaires. Il peut être contrôlé depuis une plateforme de e-learning respectant le protocole d'exécution SCORM. Il représente le plus bas niveau de granularité pouvant faire l'objet d'un suivi. Un SCO doit être le plus indépendant possible du contexte d'apprentissage afin de pouvoir être agrégé dans des unités d'apprentissage poursuivant des objectifs pédagogiques différents.

Un agrégat de contenu (*Content Aggregation*) est un ensemble de ressources pédagogiques structurées de façon cohérente au sein d'une entité de plus haut niveau, telle qu'un cours, un chapitre, un module, etc. Les ressources pédagogiques peuvent être aussi bien des ressources numériques élémentaires que des objets de contenu partageables.

L'environnement d'exécution (*Run-Time Environment*) fournit un ensemble de recommandations permettant aux développeurs d'intégrer et d'exploiter les objets pédagogiques au sein d'une plateforme de e-learning.

Le SCORM est donc un cadre spécifique pour les contenus de formations dispensées sur le web et les recommandations fournies dans l'environnement d'exécution ne sont donc pas adaptées au paramétrage de tout type de logiciel.

D. SYNTHÈSE SUR LES STANDARDS

Ces normes et standards ne se sont pas vraiment intéressés à la description du paramétrage des logiciels, et de ce fait ne comportent pas un spectre suffisant de métadonnées permettant la description des informations nécessaires aux pédagogues voulant les personnaliser. Ces informations manquantes portent sur l'aspect logiciel spécifique des EIAH, comme les besoins techniques (matériels et logiciels) pour les faire fonctionner, les caractéristiques de leur paramétrage (fichier de configuration, propriétés représentées dans ces fichiers, valeurs possibles pour ces propriétés, incidence de la modification d'une propriété sur l'aspect pédagogique de l'EIAH) et enfin les règles et les dépendances d'assemblage de ces propriétés.

2.3.2. LSCM - SCHÉMA DE DESCRIPTION DES COMPOSANTS LOGICIELS

Face à l'absence de schéma de métadonnées dédié aux composants logiciels et plus spécialement à ceux destinés à l'apprentissage et l'enseignement, [Rebaï 2006] a proposé le schéma LSCM (Learning Software Component Metadata) permettant de décrire les composants logiciels servant à construire des EIAH.

LSCM repose sur la catégorisation des composants logiciels spécifiques aux EIAH en quatre classes définies comme suit [Rebaï et al. 2004] :

- Les composants logiciels pédagogiques (CLP) sont des composants métiers apportant une plus-value pédagogique. Ils participent directement ou indirectement au processus d'apprentissage humain. Ce sont des composants réutilisables essentiellement dans les EIAH ;
- Les composants logiciels de service (CLS) sont des composants métiers apportant une plus-value fonctionnelle pouvant rendre service aux CLP ou aux utilisateurs de l'EIAH. Ce

sont des composants utiles, mais pas nécessaires pour garantir la vocation pédagogique de l'EIAH. Ils offrent des services annexes et assurent une meilleure ergonomie et maniabilité pour les utilisateurs ;

- Les composants logiciels techniques (CLT) sont des composants apportant des fonctionnalités non pédagogiques aux EIAH. Ils fournissent les mécanismes de base assurant le bon fonctionnement des CLP et CLS. Ce sont des composants susceptibles d'être utilisés dans tous les domaines ;
- Les composants logiciels de fabrication (CLF) sont des composants métiers servant à construire et à modifier des composants pédagogiques. Ce type de composant n'intervient pas lors de l'exploitation de l'EIAH, mais lors de la construction des cursus de formation.

Le schéma LSCM est divisé en deux sections : une section générique, appelée « SCM » (Software Component Metadata), décrivant les aspects génie logiciel de tout composant, et une section spécifique à chaque classe des composants logiciels, appelée « xM », où est x une chaîne de caractères présentant les initiales du domaine.

	Rubriques	Description
1	General	Regroupe les informations d'identification et de description du composant.
2	Meta_MetaData	Contient des renseignements sur les métadonnées décrivant le composant, comme la version du schéma de métadonnées utilisé.
3	Cycle_De_Vie	Décrit l'état courant du composant, ainsi que son historique.
4	Contact	Contient les coordonnées du fournisseur du composant et les services qu'il propose.
5	Droit	Contient les droits relatifs à la propriété intellectuelle et les conditions d'utilisation commerciales.
6	Information_Technique	Spécifie les contraintes matérielles et logicielles nécessaires pour utiliser les composants.
7	Relation	Énumère les composants devant être connectés avec le composant pour permettre son fonctionnement.
8	Caracteristique_Composant	Liste les propriétés et services disponibles sur ce composant.
9	Documentation	Contient des informations sur la documentation jointe au composant.
10	Annotation	Regroupe les critiques, recommandations, suggestion, etc. des utilisateurs du composant.
11	Evaluation	Permet de noter le composant selon un ensemble de critères (performance, ergonomie, etc.).
12	Extension	Permet d'étendre le schéma SCM avec de nouvelles métadonnées ou avec des entrées dans des systèmes de classification.

Figure 2-4 : Rubriques de la section générique « Software Component Metadata » du schéma LSCM [Rebaï 2006].

Les métadonnées de la section générique SCM permettent de décrire tous les composants logiciels indépendamment du domaine et de la discipline. Elles sont organisées en douze rubriques et sont, pour partie, issues du LOM. La Figure 2-4 décrit chacune de ces rubriques.

La section SCM comporte suffisamment d'informations pour pouvoir assembler un composant, le configurer et le déployer sur une architecture matérielle et logicielle. Mais pour pouvoir l'utiliser, il est nécessaire d'utiliser les informations de la section spécifique du composant. Pour les composants logiciels pédagogiques (CLP), la section spécifique s'appelle « LM » (Learning Metadata) et décrit les aspects pédagogiques, didactiques et exploratoires du composant logiciel. Les métadonnées de cette section sont organisées en six rubriques, décrites dans la Figure 2-5, dont certaines sont optionnelles.

	Rubriques	Description	Rubrique obligatoire
1	Catégorie	Précise la classe d'environnements informatiques dédiés à l'apprentissage humain (tuteur intelligent, micromonde, etc.).	X
2	Utilisation	Précise les conditions d'exploitation et les situations de formation prévues par les concepteurs et supportées par le composant.	X
3	Pédagogie	Décrit le composant selon un angle pédagogique en donnant des indications sur les méthodes et les démarches d'enseignement implémentées ou supportées par le composant et sur le contexte pédagogique dans lequel il est prévu de l'utiliser. Si le composant peut être configuré pour adopter des principes pédagogiques différents, plusieurs instances de « Pédagogie » doivent être créées.	X
4	Didactique	Décrit les disciplines, les savoirs et les ressources pédagogiques manipulés par le composant.	X
5	Exploitation	Précise les interactions que peut avoir un acteur avec le composant.	
6	Durée	Permet d'énumérer les actions ou les usages du composant ayant une durée d'exécution impactant sur le déroulement de l'apprentissage.	

Figure 2-5 : Rubriques de la section spécifique « Learning Metadata » du schéma LSCM [Rebaï 2006].

Dans notre recherche, nous cherchons à décrire un EIAH dans un but de personnalisation externalisée. Nous avons donc regardé en détail certaines rubriques du schéma LSCM pour savoir si les métadonnées qu'elles contiennent permettent cette description. La rubrique « Information Technique » (cf. 6 sur la Figure 2-4) décrit les propriétés logicielles ainsi que les besoins matériels et logiciels du composant pédagogique. La rubrique « Caractéristique Composant » (cf. 8 sur la Figure 2-4) décrit les caractéristiques de ses interfaces de communication. Les informations de ces deux rubriques sont importantes pour construire des EIAH car elles permettent de connaître les conditions nécessaires à la réutilisation des composants et les connexions à établir pour leur assemblage. Concernant les composants logiciels pédagogiques (CLP), la rubrique « Pédagogie » (cf. 3 sur la Figure 2-5) décrit les méthodes et les démarches d'enseignement supportées par le composant et ainsi que le contexte pédagogique dans lequel il est prévu de l'utiliser. En se basant sur ces informations, les enseignants peuvent décider s'ils peuvent utiliser le composant dans leur activité d'enseignement et obtenir des indications sur les choix pédagogiques permettant de mieux tirer profit des services fournis par le composant. La rubrique « Didactique » décrit les disciplines et les savoirs manipulés par le composant. Les informations de ces deux rubriques sont utiles pour

savoir quel composant utiliser pour une situation pédagogique donnée mais ne permettent pas d'adapter le composant à une situation pédagogique précise.

Le schéma LSCM permet donc de décrire des composants à des fins de réutilisation, mais non à des fins de personnalisation comme nous le souhaitons. Ses métadonnées ne permettent pas de décrire précisément les paramètres de configuration d'un composant, pour agir par exemple sur le choix des activités ou sur l'ordonnancement de celles-ci, de manière à l'adapter au souhait précis d'un enseignant.

2.3.3. SYNTHÈSE DES APPROCHES RELATIVES À LA DESCRIPTION D'EIAH

Même s'ils ont de nombreux points communs, les normes et standards que nous venons de présenter n'ont pas les mêmes objectifs et leur métadonnées ne servent donc pas à décrire les mêmes choses.

Le Dublin Core permet de décrire les métadonnées nécessaires à l'identification et la description générale d'une ressource numérique. Étant généraliste, il ne contient aucune métadonnée relative aux spécificités des ressources pédagogiques.

Le LOM sert à décrire des ressources pédagogiques. Il s'intéresse surtout au contenu de ressources stockées dans un fichier. Les métadonnées sur les caractéristiques techniques d'un objet pédagogique sont donc très limitées. Elles sont suffisantes pour des entités simples comme la majorité des objets pédagogiques, mais ne permettent pas de décrire des éléments aussi complexes que les composants logiciels ou des EIAH dans leur globalité.

Le SCORM est destiné aux ressources pédagogiques intégrables dans un dispositif d'apprentissage en ligne. Les recommandations techniques qu'il fournit ne sont donc pas adaptées au paramétrage des logiciels.

Quant au schéma LSCM, il sert à décrire des composants logiciels utilisés dans les EIAH en adoptant le point de vue des constituants fonctionnels des plates-formes de formation. Il contient pour cela plusieurs métadonnées permettant de définir les propriétés techniques pour configurer un composant comme un tout. Ces propriétés ne permettent pas d'agir par exemple sur les activités proposées dans le composant.

Les différents normes et standards proposés actuellement ne fournissent donc pas de solution pour décrire les EIAH à des fins de personnalisation. Ils ne permettent pas une description de leur contenu pédagogique suffisamment fine pour en permettre la personnalisation. Ils ne permettent pas non plus de décrire les connaissances techniques permettant d'agir sur le système afin de mettre en place cette personnalisation externalisée.

CHAPITRE 3. ANALYSE DE L'EXISTANT, SYNTHÈSE ET PROPOSITION

PLAN DU CHAPITRE

3.1. Retour sur les scénarios d'usage.....	73
3.2. Verrous	74
3.3. Notre approche.....	75

3.1. RETOUR SUR LES SCÉNARIOS D'USAGE

En introduction, nous avons présenté six scénarios montrant les enjeux de la personnalisation des activités pédagogiques dans les pratiques des enseignants. Nous revenons à présent sur les deux premiers scénarios afin de montrer les limites de leur mise en œuvre par les différentes approches présentées dans notre état de l'art.

SCÉNARIO 1 : AIDE PERSONNALISÉE À L'ÉCOLE PRIMAIRE

Alphonse, enseignant de CE2, encadre des séances de soutien pour les élèves en difficulté de son établissement. Durant chaque séance, il encadre six élèves. Il souhaite s'appuyer sur toutes les informations dont il dispose sur les connaissances des élèves (données issues des évaluations nationales et profils d'élèves créés par le logiciel AMBRE-ADD) pour proposer des séances de remédiation soit sous forme papier-crayon, soit avec des logiciels d'apprentissage adaptés à chacun des élèves.

La première approche conseillant l'enseignant sur le contenu de l'apprentissage (cf. section 1.2 page 35) lui permet de savoir quel thème étudier ou quel logiciel utiliser, mais pour cela, Alphonse doit parcourir toutes les informations qu'il possède pour chacun des six élèves. Cette tâche est d'autant plus longue que les informations se trouvent dans des endroits différents (dans le logiciel J'ADE pour les évaluations nationales et dans le dossier du logiciel AMBRE-ADD pour les profils que celui-ci crée) et sous des formes différentes (à l'interface ou imprimées pour les premières, dans un fichier XML pour les secondes). Il doit ensuite soit créer les exercices papier correspondant aux six thèmes à étudier, soit définir les six paramétrages du logiciel.

La seconde approche aidant à la création de ressources pédagogiques (cf. section 1.3 page 37) peut l'aider lors de la création des exercices papier. Selon les thèmes à étudier, Alphonse a le choix entre des générateurs créant rapidement des exercices (via les générateurs automatiques ou semi-automatiques) ou des outils auteur le guidant dans cette tâche de création. Dans le premier cas, il ne pourra obtenir des exercices que s'il existe des générateurs pour les thèmes qu'il doit faire travailler. Dans le second cas, la création des exercices n'est pas vraiment plus rapide que s'il les créait manuellement.

La troisième approche concerne l'adaptation des EIAH à chaque apprenant (cf. section 1.4 page 47). Trois cas de figure se posent pour les logiciels à paramétrer. Si l'apprenant peut choisir l'activité à effectuer, Alphonse devra fournir la liste des activités à faire, en s'assurant que ses élèves respectent ses recommandations. Si ces logiciels possèdent une interface enseignant, Alphonse pourra les personnaliser mais il devra apprendre à se servir de l'interface de paramétrage de chacun des logiciels. Enfin, si le logiciel s'adapte automatiquement à l'apprenant, Alphonse ne pourra pas adapter ce logiciel à ses choix pédagogiques.

Enfin, la dernière approche, celle des scénarios pédagogiques (cf. section 1.5 page 54), permet de scénariser les séances de soutien en précisant quel apprenant doit effectuer quelle tâche et sur quel support (papier ou logiciel). Comme la première approche, cette approche ne permet pas d'obtenir les ressources sur lesquelles les élèves doivent travailler.

Ce scénario montre que pour pouvoir créer des séances adaptées aux informations qu'un enseignant a sur ses apprenants, celui-ci doit d'une part traiter lui-même ces informations et d'autre part, apprendre à manipuler différents outils (générateurs et/ou interfaces de paramétrage) pour obtenir des ressources adaptées à ses élèves et à ses buts pédagogiques.

SCÉNARIO 2 : COMBINER REMÉDIATION ET APPROFONDISSEMENT

Fred, professeur d'anglais au collège, souhaite que les logiciels que ses élèves utilisent en salle informatique s'adaptent automatiquement à leurs lacunes.

Fred ayant déjà choisi les logiciels que ses élèves utilisent, il n'a besoin d'aucun conseil pour les choisir, ni d'aucune aide pour les créer. De plus, ces logiciels ne contenant pas de module pédagogique adaptant les sessions à l'apprenant les utilisant, ils ne peuvent s'adapter automatiquement au besoin de ses élèves. Quand les logiciels possèdent une interface enseignant, Fred pourra, pour chaque élève, définir le paramétrage du logiciel. Enfin, il a précisément défini le déroulement des séances en salle informatique. Des outils lui permettant de les scénariser ne lui seraient donc d'aucune utilité.

Ce scénario montre qu'actuellement la personnalisation automatique de logiciels existants est compliquée. Si les logiciels ne possèdent pas de module pédagogique la mettant en œuvre, aucune personnalisation automatique n'est possible. S'ils possèdent une interface de paramétrage, il faut, pour chaque apprenant, définir ce paramétrage. Cette personnalisation n'est donc pas automatique et demande beaucoup de temps à l'enseignant.

Ces scénarios d'usage étant les plus représentatifs, nous ne reviendrons pas sur les quatre autres. En effet, à eux deux ils couvrent les limites de mises en œuvre des six scénarios.

3.2. VERROUS

Ce retour sur les scénarios d'usage a permis de souligner les limites des approches actuelles permettant une personnalisation de l'apprentissage. Tout d'abord, pour se servir d'informations sur les apprenants issues de sources diverses (évaluations papier, observations de l'enseignant, profils issus de logiciels pédagogiques) afin de personnaliser les activités proposées aux apprenants, un enseignant doit actuellement traiter ces informations lui-même, ce pour chaque apprenant. Ensuite, pour générer des exercices papier ou paramétrer plusieurs logiciels, les enseignants doivent apprendre à manipuler différents outils (générateurs ou interfaces de paramétrage). Enfin, il est actuellement impossible de personnaliser automatiquement les logiciels pédagogiques s'ils ne possèdent pas de module pédagogique gérant cette personnalisation.

Pour dépasser ces limites, et permettre la réalisation des différents scénarios présentés en introduction de ce manuscrit, il est nécessaire de s'intéresser aux trois verrous suivants :

- Comment diminuer la charge de travail d'un enseignant quand il traite les informations relatives aux apprenants ?
- Comment homogénéiser les outils de personnalisation des activités pédagogiques (générateurs d'exercices papier et paramétreurs d'EIAH) ?
- Comment permettre une personnalisation automatique de logiciels existants n'ayant pas prévu cette adaptation ?

3.3. NOTRE APPROCHE

Actuellement, si un enseignant souhaite adapter à ses habitudes de travail les activités pédagogiques proposées aux apprenants, il doit maîtriser une multitude d'outils. Il a le choix entre créer ses exercices sur papier, utiliser différents générateurs d'exercices, modifier les fichiers de configuration des EIAH ou se servir des interfaces de paramétrage des EIAH lorsqu'elles existent. Pour les outils informatisés, il doit à chaque fois prendre en main un nouvel environnement, puisque d'un outil à l'autre, les interfaces et les principes d'utilisation diffèrent. De plus, s'il souhaite adapter les activités aux informations dont il dispose sur chaque apprenant (ses observations, divers profils papiers ou informatisés), il doit traiter manuellement l'ensemble des informations pour chaque apprenant. Cette complexité est illustrée sur la Figure 3-1.

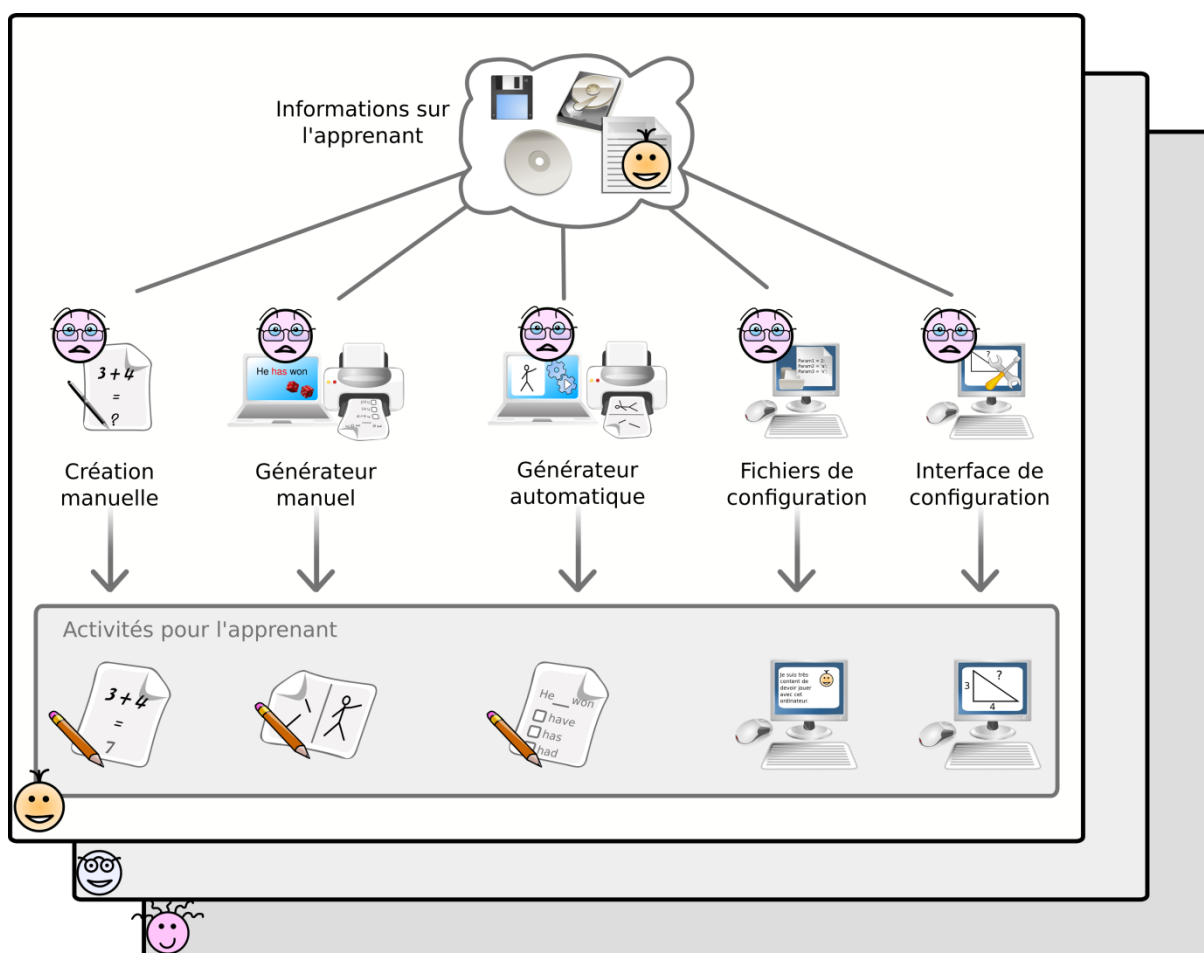


Figure 3-1 : Principe de la personnalisation par un enseignant des activités pédagogiques pour un apprenant.

Nous proposons une personnalisation unifiée de l'apprentissage, ayant pour support un outil unique utilisé par l'enseignant. Cet outil intègre les diverses informations sur les apprenants et permet d'adapter les activités aux besoins pédagogiques de l'enseignant, à travers une interface unique (cf. Figure 3-2). Pour cela, cet outil s'appuie sur des modèles de personnalisation définis en fonction des buts pédagogiques de chaque enseignant.

Nous défendons l'idée qu'il est possible de formaliser les différents éléments nécessaires à la mise en œuvre de cette approche. Tout d'abord, les informations relatives aux apprenants issues de sources diverses peuvent être décrites dans un même formalisme, par exemple le langage de

description de profils PMDL [Eyssautier-Bavay 2008, Jean-Daubias et al. 2009a]. Nous avons augmenté ce langage en proposant le modèle cPMDL, détaillé dans le Chapitre 5, permettant de contraindre les éléments d'un profil pour en sélectionner une partie. De même, la création d'activités, papier ou logicielles, repose sur des modèles implicites ou explicites, qui peuvent être formalisés. Nous proposons pour cela l'approche GEPPETO, détaillée dans le Chapitre 6. Enfin, il est possible de définir des modèles de personnalisation permettant d'acquérir les pratiques éducatives des enseignants. Nous proposons pour cela le modèle PERSUA2, détaillé dans le Chapitre 4.

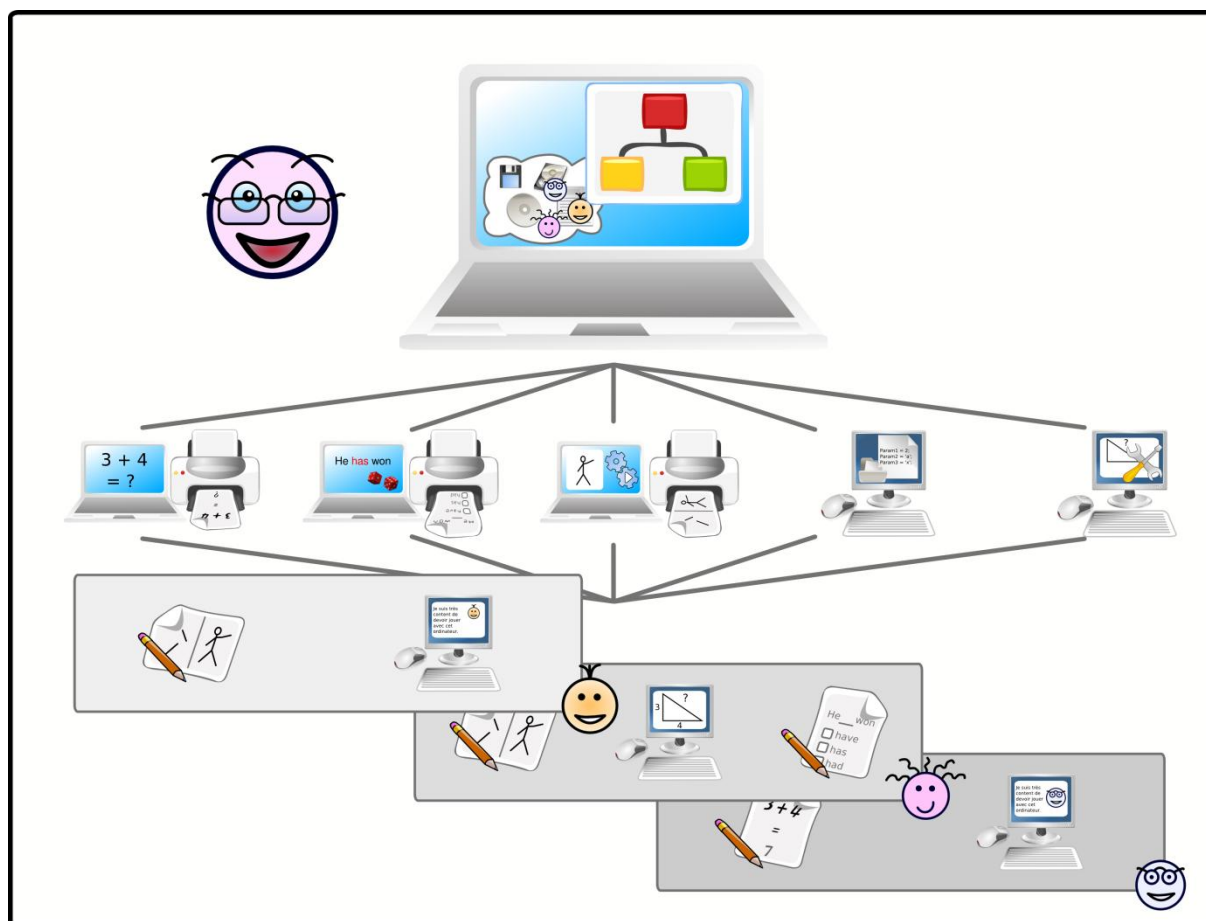


Figure 3-2 : Principe d'une personnalisation unifiée des activités pédagogiques.

Comme nous le verrons par la suite, l'approche que nous proposons repose sur différents niveaux de modèles qui doivent être définis par des experts ou en collaboration avec des enseignants. En cela, notre travail s'inscrit dans une démarche d'Ingénierie Dirigée par les Modèles (IDM) [Nodenot 2005, Choquet 2007].

L'un des avantages de la formalisation de la personnalisation est qu'elle permet à chaque enseignant de matérialiser son savoir sous forme de modèles explicites. Ces modèles peuvent alors être partagés, échangés et réutilisés par les enseignants et peuvent également conduire à une augmentation de leur productivité.

Notre revue de l'existant montre qu'il n'y a pas eu de tentative d'unification des approches permettant une personnalisation de l'apprentissage. En fonction des problématiques qu'elles abordent, les différentes propositions ne traitent qu'une sous-partie des aspects de la question de la personnalisation de l'apprentissage. Il n'existe donc aucun système permettant une personnalisation externalisée des activités papier et de plusieurs EIAH en adaptant les

séquences d'activités proposées aux apprenants à partir de leur profil tout en laissant intervenir l'enseignant dans les choix de personnalisation.

Les différents aspects de l'approche que nous proposons, ainsi que sa mise en œuvre dans le logiciel Adapte, sont détaillés dans la suite de ce manuscrit.

PARTIE 2. CONTRIBUTIONS THÉORIQUES

PLAN DE LA PARTIE

La personnalisation de l'apprentissage est une tâche complexe à mettre en œuvre, tout particulièrement si l'on souhaite adopter une démarche générique, puisqu'il existe d'une part des situations pédagogiques très diverses, et d'autre part une grande variété d'activités pédagogiques.

L'hétérogénéité des situations pédagogiques s'explique notamment par la diversité des types d'acteurs et des rôles que ceux-ci peuvent avoir sur l'activité d'apprentissage. Ainsi, l'acteur *enseignant* peut avoir un rôle de prescripteur de l'activité et/ou de tuteur, tandis que l'acteur *apprenant* intervient dans des situations d'apprentissage individuelles, collectives ou encore collaboratives.

L'hétérogénéité des activités pédagogiques résulte tout d'abord du support sur lesquelles elles sont proposées : activité à faire sur une feuille papier ou au sein d'un EIAH. De plus, les logiciels pédagogiques sont divers de part l'environnement proposé, mais également de part le contenu de cet environnement [Wenger 1987, Bruillard 1997]. Ainsi les logiciels pédagogiques peuvent prendre la forme d'un tuteur intelligent, d'un micromonde, d'un simulateur, d'un hypertexte, etc. À chacune de ces formes sont associés un mode d'utilisation (parcours libre ou guidé), un contenu (séquence d'activités prédéfinies, ensemble d'objets pouvant être manipulés...), mais également des buts pédagogiques variés (acquisition d'une méthode, d'un ensemble de connaissances, d'une pratique...).

Cette double hétérogénéité exige que l'enseignant puisse d'une part construire le type de profils qui lui permettra de gérer la situation pédagogique des apprenants, et d'autre part qu'il puisse adapter les activités à ses propres objectifs. Pour cela, il est nécessaire de lui fournir un outil permettant non seulement de créer des profils d'apprenants contenant les informations adéquates, et mais aussi d'agir sur les activités pédagogiques proposées. Cet outil doit de plus lui permettre de faire le lien entre le contenu des profils et les modifications à faire sur les activités.

Dans l'optique de construire un tel outil, nous proposons un modèle de personnalisation unifiée que nous présentons dans cette partie. Ainsi, le Chapitre 4 présente les principes, puis la définition formelle de ce modèle de personnalisation unifiée. Le Chapitre 5 détaille la partie du modèle permettant de prendre en compte les spécificités des apprenants à travers l'utilisation de profils d'apprenants. Cette partie du modèle, appelée cPMDL est une extension du langage de modélisation de profil existant PMDL. Le Chapitre 6 détaille la partie du modèle permettant d'adapter les activités à des besoins pédagogiques précis à travers l'approche GEPPETO. Enfin le Chapitre 7 présente le processus permettant d'exploiter le modèle PERSUA2 afin créer des séquences de travail adaptées aux profils des apprenants et aux besoins pédagogiques des enseignants. L'illustration de la mise en œuvre de ce modèle de personnalisation unifiée est faite par le développement d'un nouveau module de l'environnement EPROFILEA, le module Adapte [Lefevre et al. 2008b, Lefevre et al. 2009a], présenté dans le Chapitre 8.

Notons que la personnalisation des activités proposées aux apprenants est faite par l'enseignant soit directement, soit par l'intermédiaire d'un EIAH. Dans tous les cas, ce sont les choix pédagogiques de l'enseignant qui sont mis en œuvre. Pour refléter cette dualité dans les acteurs effectifs de la personnalisation, nous utilisons le terme *agent* et non *enseignant* lorsque nous présentons nos contributions théoriques.

CHAPITRE 4. PERSUA2 : UN MODÈLE POUR UNE PERSONNALISATION UNIFIÉE DES ACTIVITÉS D'APRENTISSAGE

*Comment attribuer
une activité à un apprenant ?*

PLAN DU CHAPITRE

4.1. Introduction.....	83
4.2. Principe du modèle PERSUA2	83
4.3. Définition formelle du modèle PERSUA2	84
4.4. Principe du processus d'exploitation du modèle PERSUA2.....	87
4.5. Comparaison de PERSUA2 avec d'autres approches utilisant des règles pédagogiques d'adaptation à l'apprenant.....	89
4.6. Ce qu'il faut retenir	90

PUBLICATION RELATIVE À CE CHAPITRE

[Lefevre et al. 2009c]

Lefevre, M., Jean-Daubias, S., Guin, N. « Personnaliser des séquences de travail à partir de profils d'apprenants ». Poster, 4ème Conférence en Environnements Informatiques pour l'Apprentissage Humain (EIAH'2009), Le Mans, France, 23-26 juin 2009.

4.1. INTRODUCTION

Pour permettre une personnalisation d'une séance de travail, il est nécessaire d'avoir des informations sur l'apprenant (connaissances, compétences, etc.) auquel la séance de travail est destinée, mais également sur la situation dans laquelle cette séance va se dérouler (lieu, durée, but pédagogique, etc.). À partir de ces deux types d'informations, il est possible de créer une séance de travail adaptée d'une part à l'apprenant et d'autre part aux buts pédagogiques de l'agent de la personnalisation (enseignant ou EIAH).

Dans ce chapitre, nous présentons tout d'abord les principes du modèle PERSUA2 que nous proposons pour définir une personnalisation des activités. Nous donnons ensuite la définition formelle de notre modèle. Enfin, nous expliquons brièvement comment ce modèle est appliqué pour créer des séquences de travail personnalisées, l'architecture du processus d'exploitation du modèle étant détaillée dans le Chapitre 7. Nous terminons par un résumé des propositions présentées dans ce chapitre.

4.2. PRINCIPE DU MODÈLE PERSUA2

Pour permettre à un agent de personnaliser une séance de travail selon ses besoins pédagogiques, il est nécessaire de connaître d'une part la manière dont il souhaite affecter une activité à un apprenant, et selon quels critères, et d'autre part dans quel contexte les apprenants vont utiliser la séquence de travail qui leur est destinée.

Le modèle PERSUA2 (PERsonnalisation Unifiée des Activités d'Apprentissage) que nous proposons comprend deux éléments (cf. Figure 4-1) : la stratégie pédagogique à mettre en place, c'est-à-dire les règles permettant d'affecter une activité à un apprenant, et le contexte d'utilisation de la stratégie pédagogique.

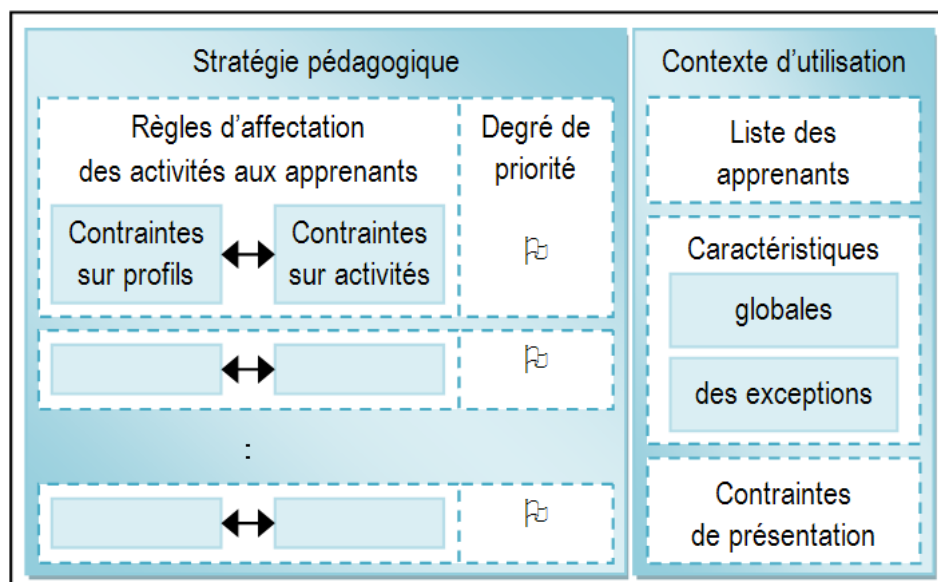


Figure 4-1 : Principe du modèle PERSUA2.

Une **stratégie pédagogique** définit comment affecter des activités aux apprenants. Cette affectation est décrite à l'aide d'un ensemble de liens hiérarchisés appelés règles d'affectation. Une règle d'affectation lie des contraintes sur le profil des apprenants à des contraintes sur une

ou plusieurs activités pédagogiques. Les contraintes sur le profil permettent de sélectionner certaines valeurs du profil des apprenants. Les contraintes sur les activités permettent de sélectionner une activité correspondant à la stratégie pédagogique à mettre en place. Par activité, nous entendons d'une part des exercices papier à imprimer, d'autre part des activités sur un logiciel pédagogique associées à la configuration de l'environnement du logiciel.

Un **contexte d'utilisation** définit, quant à lui, un ensemble d'informations permettant de caractériser la situation dans laquelle se trouve un apprenant lorsque qu'il effectue la séance de travail. Ces informations portent, entre autres, sur la liste des élèves pour lesquels une personnalisation des séances est souhaitée, sur leur profil, mais aussi sur des caractéristiques plus générales sur la séquence d'activités (nombre d'exercices, durée, etc.). Le contexte contiendra en plus de ces informations globales à tous les apprenants, des exceptions pour certains d'entre eux.

L'indépendance des deux parties du modèle de personnalisation permet d'associer une même stratégie pédagogique à plusieurs contextes d'utilisation et inversement. De plus, l'agent de la personnalisation peut définir autant de stratégies qu'il le souhaite, chacune pouvant refléter un but pédagogique différent.

Un exemple de modèle de personnalisation sera donné dans le Chapitre 7 (section 7.2, page 181) lorsque nous expliquons le principe d'exploitation de ce modèle.

4.3. DÉFINITION FORMELLE DU MODÈLE PERSUA2

Nous venons de voir les principes de notre modèle de personnalisation PERSUA2. Nous en donnons maintenant une définition formelle.

Le **modèle de la personnalisation** d'une séquence de travail voulue par un agent a est noté P_a .

$$P_a = (SP, CU)$$

Équation 4-1 : Modèle de la personnalisation souhaité par un agent donné.

L'agent a est soit un enseignant, soit un EIAH. Le modèle de personnalisation P_a peut être appliqué à un apprenant ou à un ensemble d'apprenants. Il se compose d'une stratégie pédagogique SP et d'un contexte d'utilisation CU .

Une **stratégie pédagogique** SP est un ensemble de paires constituées d'une règle d'affectation RA et de son degré de priorité DP .

$$SP = \{(RA_1, DP_1), \dots, (RA_n, DP_n)\}, n \geq 1$$

Équation 4-2 : Modèle d'une stratégie pédagogique.

Une stratégie pédagogique SP peut contenir autant de paires (RA_i, DP_i) que nécessaire. Le degré de priorité DP_i d'une règle d'affectation RA_i est une valeur appartenant à une liste graduée, définie par une borne inférieure, une borne supérieure et un pas.

Une **règle d'affectation** RA_i est un triplet composé d'une expression booléenne portant sur des contraintes sur les profils CP_j et de deux listes de contraintes sur les activités CA_k .

$$RA_i = \{\text{Exp}(CP_1, \dots, CP_m), (CA_x, \dots, CA_y), (CA_{x'}, \dots, CA_{y'})\}, m \geq 1$$

Équation 4-3 : Modèle d'une règle d'affectation.

Ce triplet représente les conditions et les conclusions d'une règle de type « si-alors ». L'Équation 4-3 est donc équivalente à l'Équation 4-4.

$$RA_i = \text{SI Exp}(CP_1, \dots, CP_m) \text{ ALORS } (CA_x, \dots, CA_y) \text{ SINON } (CA_{x'}, \dots, CA_{y'})$$

Équation 4-4 : Équivalence du modèle d'une règle d'affectation.

Les contraintes sur les profils CP_j ont été formalisées à partir du langage de description de profils PMDL [Eyssautier-Bavay 2008, Jean-Daubias et al. 2009a]. Le modèle cPMDL des contraintes CP_j et les différentes façons de les combiner pour construire une expression booléenne sont détaillés dans la section 5.4.

La première liste de contraintes sur activités (CA_x, \dots, CA_y) correspond aux contraintes permettant de créer les activités qui seront fournies à un apprenant dont les valeurs du profil respecteront l'expression booléenne. La seconde liste $(CA_{x'}, \dots, CA_{y'})$ correspond aux contraintes permettant de créer les activités qui seront fournies si les valeurs du profil ne respectent pas l'expression. La seconde liste est optionnelle. L'intersection entre les listes (CA_x, \dots, CA_y) et $(CA_{x'}, \dots, CA_{y'})$ peut être non vide, c'est-à-dire qu'un même ensemble de contraintes pourra être utilisé pour créer une activité qui sera fournie à des apprenants dont les valeurs du profil respectent ou non l'expression booléenne. La formalisation des contraintes sur activités CA_k respecte les principes de l'approche GEPPETO détaillée dans le Chapitre 6.

Un **contexte d'utilisation** CU contient la liste des profils d'apprenants LP , un ensemble de contraintes de création des séquences d'activités CC et un ensemble de contraintes de présentation CP .

$$CU = \{LP, CC, CP\}$$

Équation 4-5 : Modèle d'un contexte d'utilisation.

La liste des profils d'apprenants LP permet de connaître les apprenants concernés par la personnalisation. Les contraintes de création CC permettent de fixer des limites générales pour la création des séquences d'activités, et les contraintes de présentation CP permettent de fixer des conditions de présentation des activités au sein des séquences. Les contraintes CC et CP du contexte d'utilisation sont appliquées pour les séquences de tous les apprenants contenus dans LP .

Les **contraintes de création** CC portent sur le nombre d'activités NA , la variation des énoncés VE , la durée de la séance D , le support de la séquence S et les exceptions Ex . Ces contraintes vont intervenir lors de la création des séquences de travail personnalisées.

$$CC = \{NA, VE, D, S, Ex(NA, D, S)\}$$

Équation 4-6 : Modèle des contraintes de création d'une séquence d'activités.

Le nombre d'activités NA est un intervalle $[NA_{min}, NA_{max}]$ permettant de spécifier un nombre d'activités minimum et un nombre d'activités maximum que doit contenir la séquence de travail.

La variation des énoncés VE permet d'indiquer, dans le cas où plusieurs apprenants doivent faire la même activité, si l'instanciation de l'activité doit être la même pour tous les apprenants ou, différente pour chaque apprenant. Cette contrainte peut être utilisée pour fournir, par exemple, une évaluation où des énoncés d'exercices sont différents d'un élève à l'autre.

La durée de la séance D est un intervalle $[D_{min}, D_{max}]$ permettant de spécifier une durée minimum et une durée maximum pour la séquence de travail. Cette contrainte sert par exemple à adapter la durée de la séquence au temps dont dispose un enseignant pour faire travailler ses élèves en soutien ou pour fournir un devoir maison personnalisé d'une certaine durée.

Le **support d'une séquence d'activités** S précise les contraintes techniques pour créer la séquence travail.

$$S = \{M, L(MP), FS\}$$

Équation 4-7 : Modèle du support d'une séquence d'activités.

La première contrainte concerne le matériel M dont disposent les apprenants pour effectuer la séquence de travail (papier, ordinateur ou les deux). Si cette contrainte indique que les apprenants travaillent sur papier, d'autres contraintes sont alors disponibles : la longueur de la séquence L , la mise en page MP et le format de sortie FS .

La longueur de la séquence L est un intervalle $[L_{min}, L_{max}]$ permettant de définir une longueur physique pour la page à imprimer à l'aide d'un minimum et d'un maximum. Cette longueur physique dépend des contraintes de mise en page MP .

Les contraintes de mise en page MP permettent de spécifier la mise en forme des caractères : police, taille, style, etc.

Le format de sortie FS permet de préciser le format de la feuille d'exercices à imprimer (RTF, HTML...).

Les exceptions Ex permettent de définir des contraintes propres à certains apprenants dont les profils sont fournis dans LP . Ces exceptions peuvent porter sur le nombre d'activités NA , la durée de la séance D mais aussi le support de la séquence d'activité S . Elles permettent de prendre en compte des difficultés ou les facilités de certains apprenants, leurs contraintes d'emploi du temps, etc. Un enseignant pourra par exemple faire travailler tous ses élèves durant la dernière heure de cours de la journée, sauf ceux qui doivent partir en soutien la dernière demi-heure. Il pourra aussi adapter la police d'écriture des feuilles à imprimer pour les élèves ayant des problèmes de vue (taille de caractère plus élevée), de dyslexie (police sans sérif), etc.

Les contraintes de création sont toutes optionnelles et peuvent être combinées. Il est par exemple possible de demander une séquence d'exercices sur papier (M) contenant entre trois et cinq exercices (NA) pour une durée totale de trente minutes (D). Dans cet exemple, les contraintes VE , L , MP , FS ne sont pas définies et aucune exception n'est précisée. Pour ces contraintes non définies par l'agent de la personnalisation, les séquences d'activités seront créées à partir des valeurs par défaut propres au système mettant en œuvre cette personnalisation. Par exemple, un premier système pourra proposer des fichiers PDF à imprimer tandis qu'un autre fournira des feuilles au format HTML.

Les **contraintes de présentation** CP portent sur l'ordonnancement des activités dans la séquence de travail.

$$CP = \{(CT_1, \dots, CT_p)\}, p \geq 1$$

Équation 4-8 : Modèle des contraintes de présentation d'une personnalisation.

Elles ne modifient pas le contenu d'une séquence, mais permettent de classer les activités en fonction de certains critères de tri CT_k . Les critères portent sur les métadonnées des activités contenues dans la séance. Ces métadonnées sont présentées Figure 8-18 page 219.

La Figure 4-2 reprend la Figure 4-1 (cf. page 83) en situant les différents modèles formels présentés dans cette section.

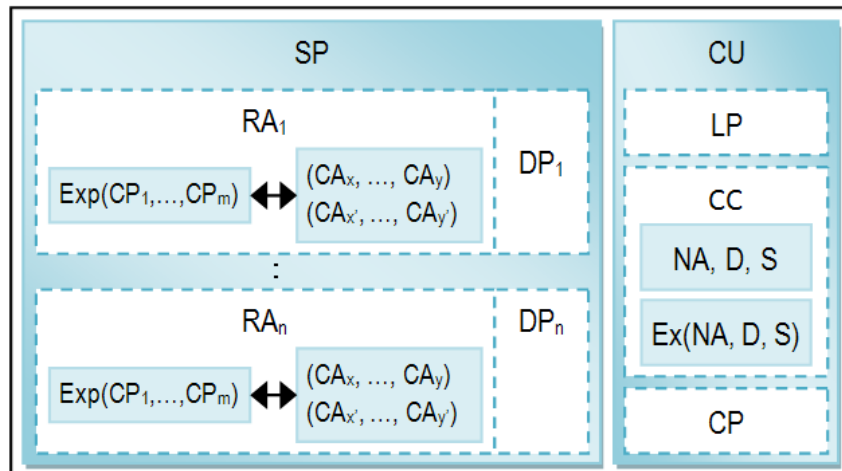


Figure 4-2 : Représentation graphique du modèle PERSUA2.

4.4. PRINCIPE DU PROCESSUS D'EXPLOITATION DU MODÈLE PERSUA2

Le modèle de personnalisation présenté dans la section précédente permet à chaque agent de la personnalisation de choisir comment affecter des activités aux apprenants en fonction de ses propres buts pédagogiques.

Pour exploiter ce modèle, nous proposons un processus permettant de l'appliquer pour créer des séquences d'activités personnalisées en fonction du contenu des profils des apprenants. Dans cette section nous présentons le principe de ce processus (cf. Figure 4-3), son architecture étant détaillée dans la section 7.3 page 187, quand les contraintes sur les profils et les contraintes sur les activités auront été présentées.

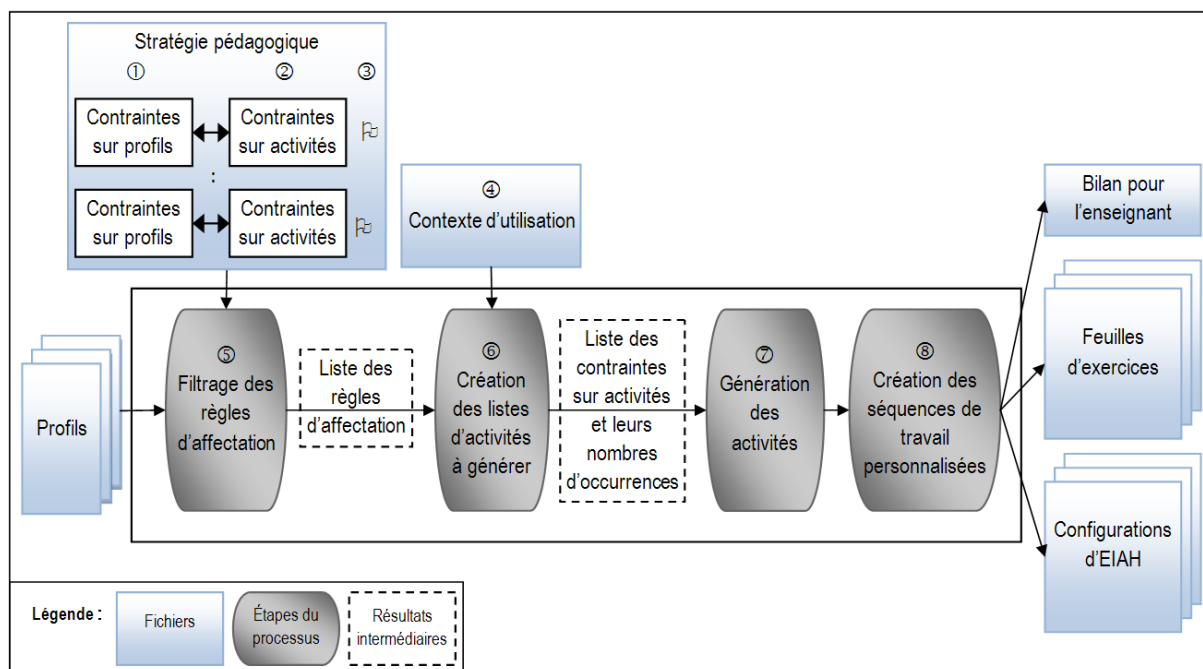


Figure 4-3 : Principe du processus d'exploitation du modèle PERSUA2.

L'utilisation du modèle de personnalisation se fait en deux temps.

Dans un premier temps, l'agent de la personnalisation crée son modèle de personnalisation avec ses objectifs pédagogiques et ses contraintes d'utilisation. Pour cela, il définit des règles d'affectation des activités aux apprenants en créant des filtres permettant de ne garder que les valeurs du profil qui lui semblent pertinentes (cf. ① sur la Figure 4-3) et en contraignant les activités à fournir (cf. ② sur la Figure 4-3). Ensuite il hiérarchise ces règles d'affectation en fonction de leur niveau de priorité pour créer sa stratégie pédagogique (cf. ③ sur la Figure 4-3). Enfin, il définit un contexte d'utilisation contenant entre autres les profils des apprenants pour lesquels il souhaite une séquence personnalisée (cf. ④ sur la Figure 4-3).

Dans un second temps, ce modèle de personnalisation est appliqué, pour chaque apprenant, en quatre étapes. Tout d'abord, les règles d'affectation sont filtrées en fonction des valeurs du profil de l'apprenant (cf. ⑤ sur la Figure 4-3). Ensuite, les activités à générer, contenues dans les règles activées lors de la première étape sont filtrées en fonction du contexte d'utilisation (cf. ⑥ sur la Figure 4-3). À partir de la liste des activités à générer contenant les contraintes de génération, et pour chaque ensemble de contraintes, le nombre d'activités à générer, les activités personnalisées sont créées (cf. ⑦ sur la Figure 4-3).

Au final, l'exploitation du modèle de personnalisation permet de créer autant de séquences personnalisées qu'il y a de profils d'apprenants, ainsi qu'un bilan pour l'enseignant (cf. ⑧ sur la Figure 4-3). Ce bilan contient la liste et la description des activités, papier ou logicielles, fournies à chaque apprenant, et pour les activités papier, leur correction.

4.5. COMPARAISON DE PERSUA2 AVEC D'AUTRES APPROCHES UTILISANT DES RÈGLES PÉDAGOGIQUES D'ADAPTATION À L'APPRENANT

Plusieurs outils auteurs tels que REDEEM [Ainsworth 2000], GTE [Van Marcke 1998], SMART TRAINER [Chen et al. 1998] et EON [Murray 2003a] permettent aux enseignants créant un logiciel pédagogique de paramétrer des modèles de stratégies pédagogiques. Comme dans le modèle PERSUA2, ces stratégies pédagogiques sont des règles « si-alors-sinon ». Toutefois, dans ces systèmes, une stratégie pédagogique est associée, et même incluse, dans un EIAH donné. Un enseignant ne peut donc pas, comme avec le modèle PERSUA2, définir une stratégie menant à la création d'une séquence de travail sur plusieurs supports, c'est-à-dire contenant à la fois des activités papier et des activités sur plusieurs EIAH.

Dans le projet PÉPITE, ce sont des stratégies d'apprentissage qui sont proposées à l'enseignant pour chacun de ses apprenants. Pour cela, le système PEPISTÉRÉO [Vincent et al. 2005] attribue un stéréotype à chaque apprenant, chaque stéréotype ayant préalablement été associé à une stratégie d'apprentissage. Contrairement aux stratégies pédagogiques du modèle PERSUA2 qui vont permettre de créer pour chaque apprenant une séquence de travail unique, les stratégies d'apprentissage de PÉPITE contiennent une suite d'étapes permettant de faire travailler les apprenants associés à un stéréotype. Ces étapes contiennent des exemples d'exercices ainsi qu'une liste d'EIAH pouvant être utilisés pour faire ce type d'exercices. Dans PÉPITE, plusieurs apprenants pourront donc être associés à la même stratégie d'apprentissage et c'est l'enseignant qui devra construire la séquence de travail à partir des indications de la stratégie d'apprentissage. De plus, il existe un nombre limité de stratégies d'apprentissage et celles-ci sont spécifiques au domaine de l'algèbre élémentaire.

Le modèle PERSUA2 que nous proposons est, à notre connaissance, le premier modèle permettant à un enseignant, quelque soit le domaine dans lequel il enseigne, de créer des séquences de travail propres à chaque apprenant, séquences pouvant être effectuées sur plusieurs supports, papier et/ou logiciels.

4.6. CE QU'IL FAUT RETENIR

Nous défendons l'idée que pour personnaliser des activités d'apprentissage, il est possible de définir les règles de cette personnalisation dans un modèle unique. Dans ce chapitre, nous avons proposé une définition de ce modèle de personnalisation, le modèle PERSUA2, qui peut être instancié par un enseignant ou par un EIAH.

Le modèle PERSUA2 que nous proposons contient un modèle de la stratégie pédagogique à mettre en place, c'est-à-dire une description des règles permettant d'affecter une activité à un apprenant, ainsi qu'un modèle du contexte d'utilisation d'une stratégie pédagogique.

Notre modèle PERSUA2, qui n'a à notre connaissance pas d'équivalent à l'heure actuelle, permet de modéliser aussi bien des situations de personnalisation simples que complexes. S'il n'est pas exhaustif, il possède l'avantage d'être ouvert. Il a en effet été conçu afin d'être extensible et peut ainsi évoluer pour prendre en compte de nouvelles pratiques ou de nouveaux besoins éducatifs.

CHAPITRE 5. cPMDL : UN MODÈLE DE CONTRAINTES SUR PROFILS

*Comment prendre en compte
les individualités des apprenants ?*

PLAN DU CHAPITRE

5.1.	Introduction.....	93
5.2.	Pourquoi harmoniser la structure des profils ?.....	93
5.3.	Choix d'un formalisme commun pour décrire un profil : PMDL.....	95
5.3.1.	Structuration générale de PMDL.....	95
5.3.2.	Types de données dans PMDL.....	96
a.	Type <i>liste_composantes</i>	96
b.	Type <i>liste_repartition</i>	98
c.	Type <i>graphe</i>	99
d.	Type <i>texte</i>	101
5.3.3.	Échelles dans PMDL.....	101
5.4.	Définition formelle du modèle cPMDL.....	102
5.4.1.	Contrainte sur profils portant sur une valeur.....	102
a.	Contrainte sur profils portant sur une valeur textuelle.....	103
b.	Contrainte sur profils portant sur une valeur numérique.....	103
c.	Contrainte sur profils portant sur un entier positif.....	103
d.	Exemples de contraintes sur profils portant sur une valeur.....	104
5.4.2.	Contrainte sur profils portant sur un élément.....	105
5.4.3.	Contrainte sur profils portant sur un nombre d'occurrences.....	109
5.4.4.	Combinaison de contraintes sur profils.....	110
5.5.	Ce qu'il faut retenir.....	112

5.1. INTRODUCTION

Pour prendre en compte les individualités des apprenants, le modèle de personnalisation présenté dans la section 4.3 permet de définir des contraintes sur profils. Ces contraintes permettent de sélectionner les profils d'apprenants qui les respectent.

Pour définir des contraintes sur un profil, il est nécessaire de connaître la structure de ce profil. Or, nous souhaitons prendre en compte divers profils ayant des structures différentes. Afin de définir un modèle uniforme des contraintes sur profils, il est donc nécessaire d'avoir un modèle uniforme de la structure de tous ces profils. Plusieurs formalismes et processus permettant de décrire des profils d'apprenants de manière uniforme ont déjà été proposés. Pour définir notre modèle de contraintes sur profils, nous ne nous intéressons donc pas au processus d'harmonisation des structures de profils, nous considérons qu'il se trouve en amont du processus de personnalisation, mais nous nous intéressons en revanche au formalisme utilisé lors de l'harmonisation, puisque nous nous appuyons sur ce formalisme pour définir notre modèle de contraintes sur profils.

Dans la première section de ce chapitre, nous détaillons pourquoi il est nécessaire de décrire les profils d'apprenants grâce à un unique formalisme avant de pouvoir établir des contraintes sur ces profils. Dans la seconde section, nous présentons le formalisme PMDL que nous avons retenu pour décrire les profils d'apprenants. Dans la troisième section, nous présentons l'extension du langage PMDL que nous proposons : le modèle cPMDL, un modèle de contraintes sur profils définissant les différentes contraintes qui peuvent être associées à un profil d'apprenant défini selon le formalisme PMDL.

5.2. POURQUOI HARMONISER LA STRUCTURE DES PROFILS ?

Nous définissons un profil comme un ensemble d'informations interprétées, concernant un apprenant ou un groupe d'apprenants. Celles-ci sont collectées ou déduites à l'issue d'une ou plusieurs activités pédagogiques, informatisées ou non. Les informations contenues dans le profil de l'apprenant peuvent concerner ses connaissances, compétences, conceptions, ou encore son comportement [Jean-Daubias 2003, Jean-Daubias et al. 2005].

Les profils d'apprenants créés par des systèmes informatiques ou par les enseignants dans le cadre de leurs pratiques papier-crayon peuvent être établis à la demande de différentes personnes, enseignants, institutions ou apprenants, et contenir des informations très variées. Pour un même apprenant, il peut exister plusieurs profils parfois complémentaires, parfois incompatibles, dans une ou plusieurs disciplines. Chaque profil apporte un éclairage particulier sur l'apprentissage de l'apprenant, sur l'état de ses connaissances et sa façon d'apprendre. Ces profils sont très divers, tant par leur contenu que par leur représentation externe et leur structure [Leroux et al. 2007, Eyssautier-Bavay 2008].

Cette hétérogénéité des profils rend difficile leur réutilisation. En effet, pour pouvoir proposer, comme nous le souhaitons, un modèle permettant de contraindre un profil pour n'en sélectionner que certaines valeurs, nous avons besoin de connaître le formalisme dans lequel il est défini. La solution adoptée dans le cadre du projet PERLEA est d'harmoniser la structure et le contenu des profils grâce à un formalisme commun.

Pour cela, il est nécessaire de disposer de modèles génériques permettant d'harmoniser des profils [Eyssautier-Bavay et al. 2009]. Cette harmonisation va consister en la réécriture par l'enseignant, assisté ou non d'un système informatique, de la structure des profils selon un formalisme commun, prédéfini, de modélisation de profils. L'harmonisation de profils prend en entrée un ou plusieurs profils que l'on souhaite réutiliser, sous forme papier-crayon ou numérique, et fournit en sortie autant de profils décrits dans un même formalisme.

Des travaux menés précédemment dans notre équipe de recherche (équipe SILEX du LIRIS) ont montré les limites des formalismes communs existants pour décrire un profil [Eyssautier-Bavay 2008, Jean-Daubias et al. 2009a]. La description d'un profil d'apprenant dans un formalisme commun peut se faire selon deux approches.

La première approche consiste à définir de façon consensuelle *a priori* ce qu'est un ensemble d'informations sur l'apprentissage d'un individu. C'est l'approche adoptée par les travaux sur la normalisation des données personnelles des apprenants. Cependant, les standards proposés n'ont pas pour principal but la personnalisation de l'apprentissage, ce qui explique que les informations les plus pertinentes pour la personnalisation ne soient pas présentes ou décrites suffisamment précisément dans ces standards. En effet, un profil d'apprenant représente des informations sur les connaissances de l'apprenant à un niveau de granularité fin, alors que les standards cherchent plutôt à faciliter le stockage et l'échange des données pour fournir une aide à la gestion des institutions éducatives, ce qui explique qu'ils s'intéressent à des informations de niveau de granularité plus élevé [Keenoy et al. 2004]. Ainsi, le standard PAPI ne fait pas mention des compétences, connaissances ou conceptions de l'apprenant, ni des structures que peuvent avoir ces informations [PAPI 2002]. Le standard IMS-RDCEO permet quant à lui de décrire précisément des compétences [IMS_RDCEO 2001], mais ne contient pas de données individuelles. Le standard IMS-LIP permet de représenter les différentes compétences et connaissances acquises par un apprenant particulier, mais ne décrit pas les relations des compétences entre elles, ni leur association à un ou plusieurs résultats d'évaluation [IMS-LIP 2001].

La seconde approche vise à réutiliser des profils externes dans un unique environnement informatique en les réécrivant *a posteriori* selon un formalisme interne. Ainsi les systèmes VISMOD [Zapata-Rivera et al. 2004] et DYNMAP [Rueda et al. 2006] réécrivent d'une part les données du domaine étudié, et d'autre part les données de l'apprenant considéré. Ces systèmes manquent toutefois de généralité puisqu'ils ne permettent pas de représenter les informations sous forme de texte en langage naturel, de graphes avec valeurs portées par les arcs, de composantes à valeurs issues de répartition. De plus, dans ces systèmes, l'enseignant intervient peu et ses pratiques ne sont pas prises en compte, seuls les profils issus d'EIAH étant considérés. D'autres travaux [Ramandalahy et al. 2009] proposent quant à eux un modèle du profil de l'apprenant pouvant être étendu, intégrant IMS-LIP ainsi qu'un certain nombre d'éléments d'ordre cognitif et métacognitif. Ce modèle est mis en œuvre de manière à enrichir des profils d'apprenants à partir d'applications hétérogènes. La limite principale de ces travaux est leur manque de souplesse : la prise en compte d'un élément de profils non prévu demande la modification du modèle sous-jacent et de l'application.

En réponse à ces limites, le langage PMDL a été proposé [Eyssautier-Bavay 2008] afin de permettre à l'enseignant ou au concepteur pédagogique de modéliser les profils qu'il souhaite réutiliser. Ce langage ne vise pas à décrire le modèle de connaissances du domaine, mais seulement les données de l'apprenant. Les deux avantages de ce langage sont qu'il permet d'une part de traiter des profils hétérogènes (en associant des profils issus des pratiques en classe des enseignants à des profils externes, issus d'EIAH), et d'autre part de décrire des données de types plus variés que les formalismes utilisés dans les systèmes VISMOD et DYNMAP. Ce langage, proposé

dans le cadre du projet PERLEA, a été implémenté dans l'environnement EPROFILEA. Notre thèse s'inscrivant dans ce même projet, c'est sur ce langage que nous nous appuyons pour décrire des contraintes sur le profil des apprenants dans le cadre de notre modèle de personnalisation.

5.3. CHOIX D'UN FORMALISME COMMUN POUR DÉCRIRE UN PROFIL : PMDL

Pour pouvoir spécifier le format des contraintes sur les profils décrits selon le langage PMDL, nous avons besoin de connaître les différents éléments qui le constituent. Nous présentons donc ici la structure générale du langage PMDL, avant de revenir sur les différents types de données qu'il permet de décrire et les échelles pouvant être créées pour décrire ces données. Un exemple de profil d'apprenant exprimé dans le langage PMDL est fourni en Annexe B page 285.

5.3.1. STRUCTURATION GÉNÉRALE DE PMDL

Le langage de modélisation de profils PMDL (Profiles MoDeling Language) [Eyssautier-Bavay 2008, Jean-Daubias et al. 2009a] permet l'harmonisation des profils d'apprenants. Pour cela, PMDL distingue dans un profil d'apprenant sa partie structure et sa partie donnée. La partie structure peut ainsi être commune à plusieurs profils d'apprenants alors que la partie donnée est propre à chaque apprenant.

La Figure 5-1 présente la structuration générale du langage PMDL, qui décrit un profil comme étant constitué d'un *nom*, d'un *identifiant*, d'une *date-creation*, d'une partie *structure*, d'une partie *donnees* et éventuellement d'un *commentaire*.

La **partie structure** décrit uniquement l'organisation des données. Elle est composée d'une ou plusieurs informations générales sur l'élève, *info_eleve*, et d'un ensemble non vide d'*element*.

Les *info_eleve* représentent des informations concernant l'état civil de l'apprenant, ainsi que son contexte d'apprentissage. Une *info_eleve* est constituée d'un *identifiant*, d'un *intitule* de type chaîne de caractères (par exemple « ville de résidence ») et soit d'un *type* (« chaîne de caractère » dans notre exemple), soit d'une liste énumérée, *enum* (par exemple « Lyon, Villeurbanne, Bron »). La description de la structure est indépendante d'un apprenant particulier. Les informations sur l'élève sont donc la description du type de données qui seront représentées pour chacun des apprenants et non de données individuelles.

Un *element* est constitué d'un *nom* (par exemple « algèbre »), d'un *identifiant*, éventuellement d'un *commentaire*, et d'un *contenu*. Le *contenu* permet au concepteur pédagogique de décrire les informations relatives à l'analyse de l'activité de l'apprenant. Ce *contenu* peut être de quatre types : *liste_composantes*, *liste_repartition*, *graphe* et *texte*. Nous reviendrons par la suite sur ces quatre types pour les décrire plus précisément (cf. section 5.3.2).

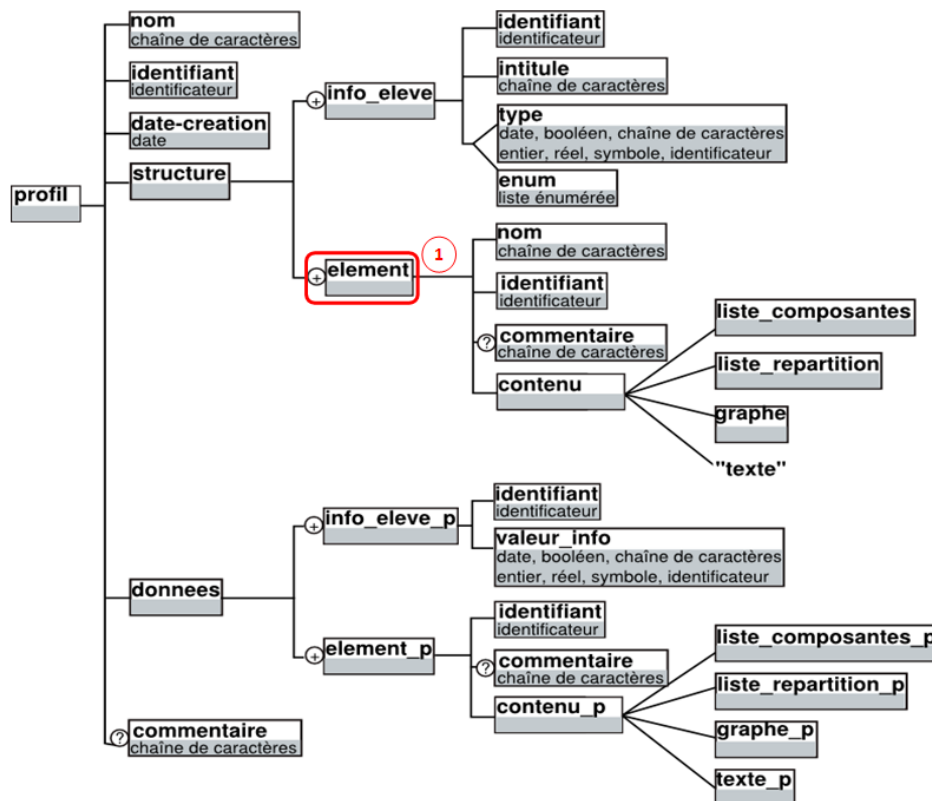


Figure 5-1 : Représentation graphique de la structuration générale de PMDL [Jean-Daubias et al. 2009a].

La **partie donnees** contient les valeurs spécifiques à un apprenant pour les éléments définis dans la partie *structure*. La partie *donnee* est donc composée, comme la partie *structure*, d'un ensemble d'informations sur l'élève, *info_eleve_p*, et d'un ensemble d'*element_p*. Il existera autant d'*info_eleve_p* (respectivement *element_p*) que d'*info_eleve* (respectivement *element*) déclarés dans la partie *structure*.

Une *info_eleve_p* est constituée d'un *identifiant* et d'une *valeur_info*. L'*identifiant* correspond à celui qui a été défini dans la partie *structure* et la *valeur_info* respecte le *type* qui a été défini dans la partie *structure*.

Un *element_p* est constitué d'un *identifiant*, éventuellement d'un *commentaire*, et d'un *contenu_p*. Pour chaque *identifiant* d'*element*, il existera un *element_p* reprenant cet *identifiant* et lui associant un *contenu_p* de même type que le *contenu* précédemment déclaré.

5.3.2. TYPES DE DONNÉES DANS PMDL

Détaillons maintenant les quatre types permettant de décrire les données de l'apprenant dans le langage PMDL.

A. TYPE LISTE_COMPOSANTES

Certaines informations contenues dans les profils sont structurées sous forme de listes de composantes à valeurs d'évaluation, hiérarchiques ou non, auxquelles sont associées une ou plusieurs valeurs. Une liste de composantes hiérarchique est une liste dans laquelle les composantes peuvent contenir des sous-composantes qui elles-mêmes peuvent être détaillées en sous-composantes, etc.

La Figure 5-2 montre deux profils d'élèves proposés par le logiciel J'ADE [J'ADE 2007b] contenant des informations de ce type. Chaque profil contient une composante « Mathématiques » à laquelle est associé un taux de réussite. Cette composante est décomposée en cinq sous-composantes, auxquelles sont également associés des taux de réussite. La valeur de la composante est calculée en faisant la moyenne des valeurs des cinq sous-composantes.

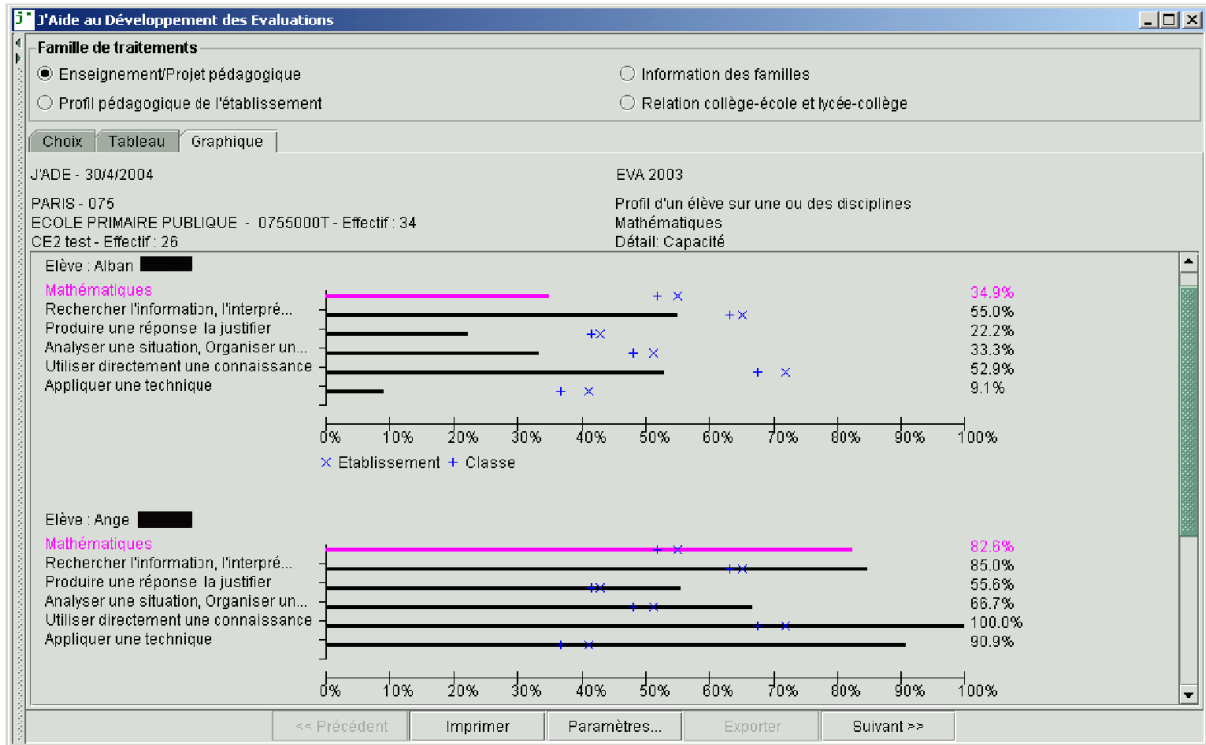


Figure 5-2 : Profils d'élèves dans J'ADE [J'ADE 2007b].

L'élément *liste_composantes* permet de représenter ce type d'information. Il est scindé en deux éléments, *liste_comp_ponderee* et *liste_comp_non_ponderee* qui possèdent une construction quasi-identique, à la présence de l'élément *poids* près (cf. Figure 5-3). L'élément *liste_composantes* comporte également une entité *echelles* représentant les types de valeurs utilisées pour l'évaluation. L'élément *echelles* étant commun à plusieurs types de données, nous reviendrons dessus après avoir présenté les quatre types de données.

L'élément *liste_comp_ponderee* est constitué d'un ensemble non-vide de *composante_ponderee*. Une *composante_ponderee* est constituée du *niveau* de la composante dans l'arbre des composantes, d'une *composante* contenant un *identifiant* et un *intitulé*, d'un *poids* et éventuellement d'un *commentaire*. Dans le cas où la *composante_ponderee* correspond à une feuille dans l'arbre des composantes, elle est précédée du marqueur *feuille*. Dans le cas inverse, elle possède un ensemble non-vide de *sous_composantes_ponderees*. Une *sous_composante_ponderee* possède la même structure qu'une *composante_ponderee*.

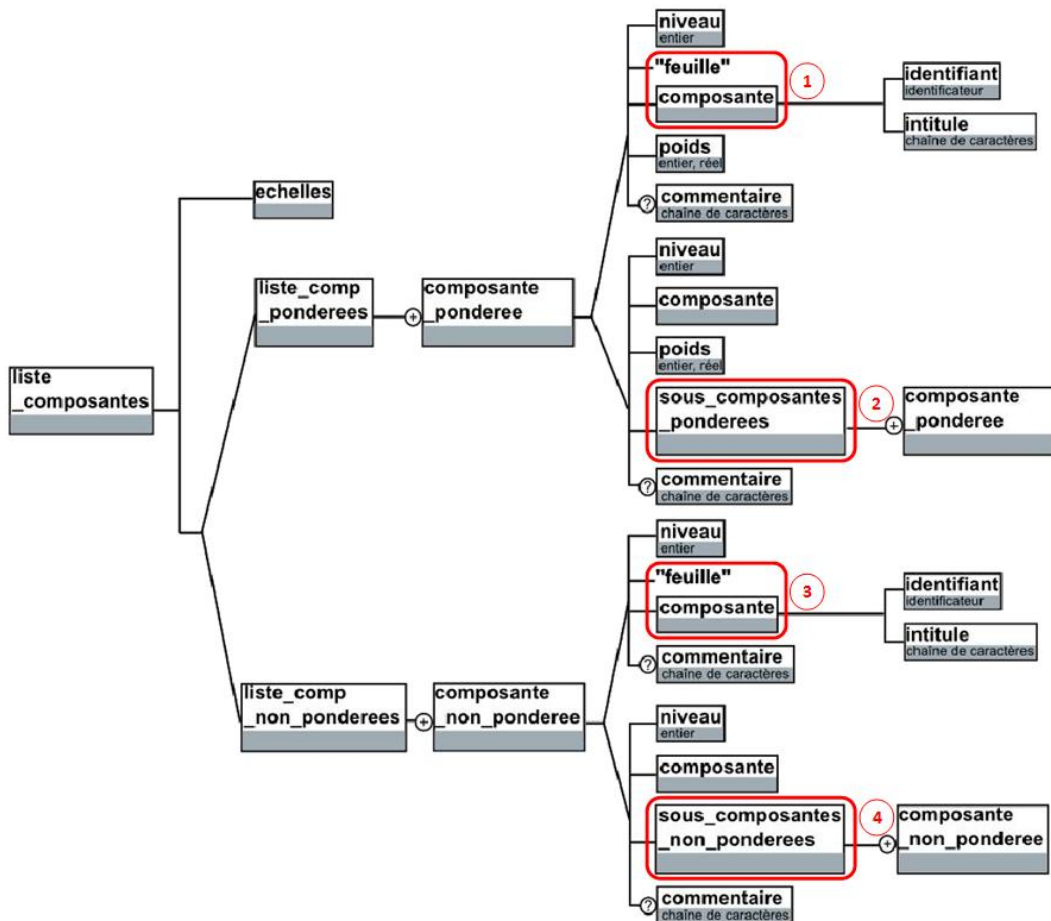


Figure 5-3 : Représentation graphique de l'élément *liste_composante* de PMDL [Eyssautier-Bavay 2008].

B. TYPE LISTE_REPARTITION

D'autres informations contenues dans les profils sont structurées sous forme de liste de composantes à valeurs de répartition. Comme dans le cas précédent, les informations sont représentées par des listes, hiérarchisées ou non, de composantes. De même, à chaque composante de niveau hiérarchique le plus bas sont associées une ou plusieurs valeurs. Mais contrairement au type de données précédent, la valeur associée à une composante n'est pas une valeur d'évaluation de la composante. Elle représente le nombre de fois où l'apprenant a mis en œuvre la composante. La différence entre ces deux catégories de liste de composantes n'est pas seulement sémantique. En effet, les valeurs d'évaluation sont des valeurs qui peuvent être très variées, telles que des entiers, des chaînes de caractères, etc., alors que les valeurs de répartition sont uniquement numériques, puisque ce sont des valeurs de comptage.

La Figure 5-4 montre une partie du profil de l'apprenant proposé par le logiciel MOREMATHS [Bull et al. 2003]. Ce profil contient trois composantes « Introduction », « Polynomial division » et « Factorising polynomials ». Pour chaque composante, les sous-composantes donnent la répartition des réponses de l'apprenant entre réponses correctes, réponses fausses et non réponses.

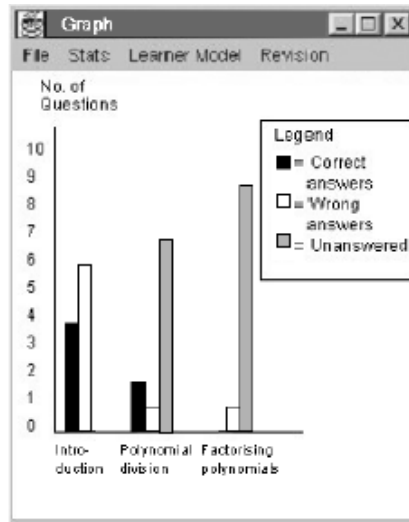


Figure 5-4 : Profil de l'apprenant dans la version mobile de MOREMATHS [Bull et al. 2003].

L'élément *liste_repartition* permet de représenter ce type d'information. Il se compose (cf. Figure 5-5) d'une *liste_comp_non_ponderees*, définie précédemment, d'un *nbre_valeurs* indiquant le nombre de valeurs de répartition que cet élément contiendra, et éventuellement d'*etiquettes* pour certaines de ces valeurs afin de leur donner du sens.

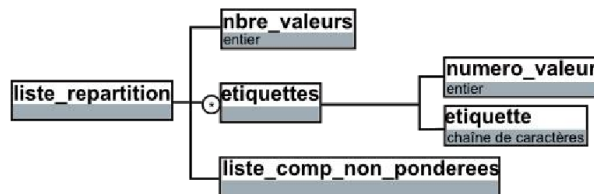


Figure 5-5 : Représentation graphique de l'élément *liste_repartition* de PMDL [Eyssautier-Bavay 2008].

C. TYPE GRAPHE

Certaines informations contenues dans les profils sont structurées sous forme d'un graphe de composantes. Un graphe est constitué de nœuds et d'arcs. Ici, les nœuds représentent les composantes, et les arcs, les liens qui peuvent exister entre ces composantes. Il n'y a pas de sous-composantes dans une représentation de type graphe. Les valeurs d'évaluation peuvent être associées aux composantes, c'est-à-dire aux nœuds du graphe, mais également aux arcs pour représenter la maîtrise du passage d'une composante à une autre. Un arc est monodirectionnel, mais il peut y avoir deux arcs entre deux mêmes éléments, de sorte qu'il y ait un arc dans chaque sens entre ces deux éléments.

La Figure 5-6 montre un profil de l'apprenant représenté sous forme de graphe dans le logiciel PÉPITE [Jean 2000]. Dans ce profil, les nœuds représentent les composantes, ici « cadre algébrique », « cadre graphique », « cadre numérique », « cadre géométrique » et « langage naturel » et les valeurs sont portées par les arcs. L'objectif est de représenter si le passage d'une composante à un autre est maîtrisé ou non. À chaque arc sont associées deux valeurs : le nombre total d'exercices associés et le nombre d'exercices réussis. Ainsi, dans cet exemple, le lien entre « cadre algébrique » et « cadre numérique » a été réussi 3 fois sur 13 exercices par l'apprenant. Les valeurs des arcs sont de plus dupliquées grâce à une représentation graphique du lien : la couleur et la forme du trait marquant ce lien sont fonction du rapport entre la première et la deuxième valeur portées par les arcs.

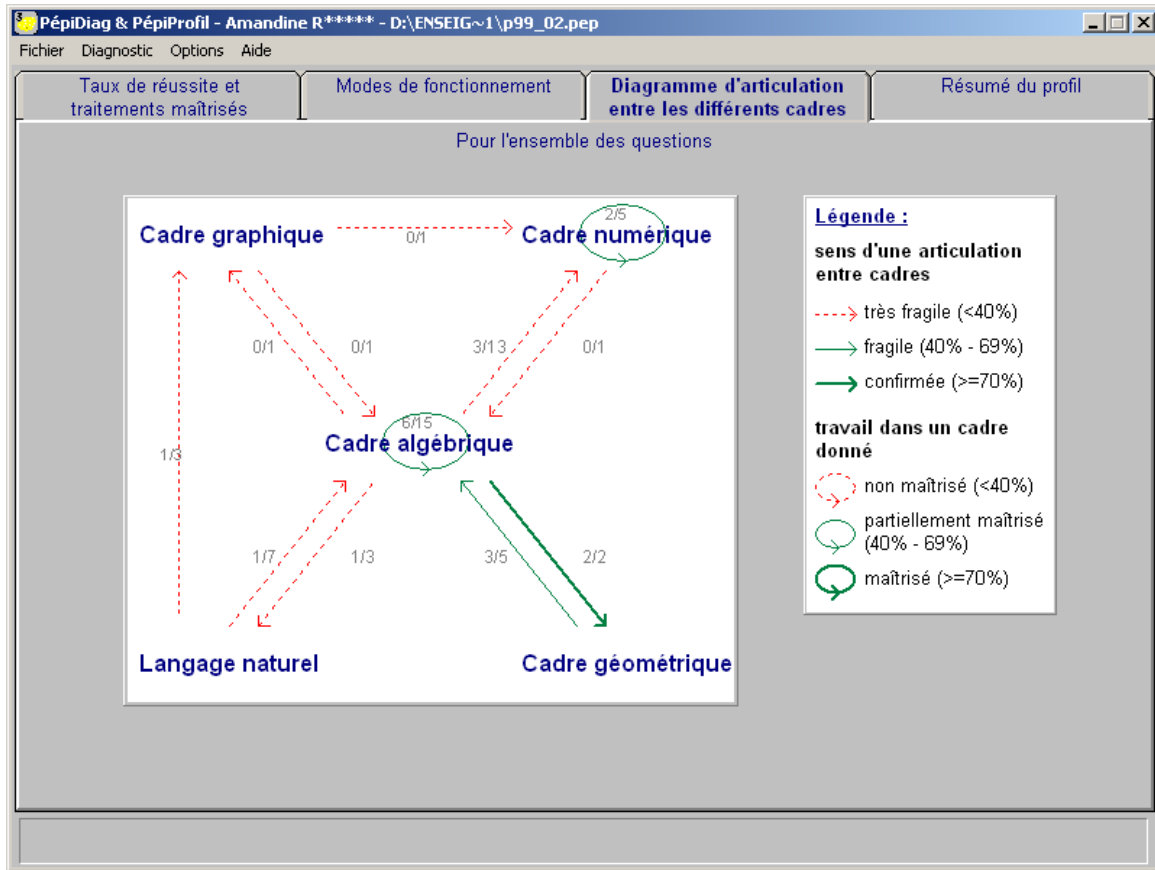


Figure 5-6 : Profil de l'apprenant dans PÉPITE [Jean 2000].

L'élément *graphe* permet de représenter ce type d'information. Un *graphe* est un ensemble de *composantes* et de *lien* (cf. Figure 5-7). Il est possible d'associer des *echelles* aux *composantes* et/ou aux *lien*. Chaque *composante* peut être accompagnée d'un *commentaire*, valable pour tous les profils partageant cette structure. Un *lien* est constitué d'un *identifiant*, d'un couple de deux *id_composante*, le premier étant l'identifiant de la *composante* origine et le second celui de la *composante* destination. Ces *composante* doivent appartenir à l'ensemble des *composante* décrites précédemment dans *graphe*. Un *lien* peut également avoir un *type_lien* permettant de donner une sémantique au lien si tous n'ont pas le même (par exemple l'un sera « héritage », l'autre « précedence »). Enfin, un *commentaire* peut être attaché à un *lien*.

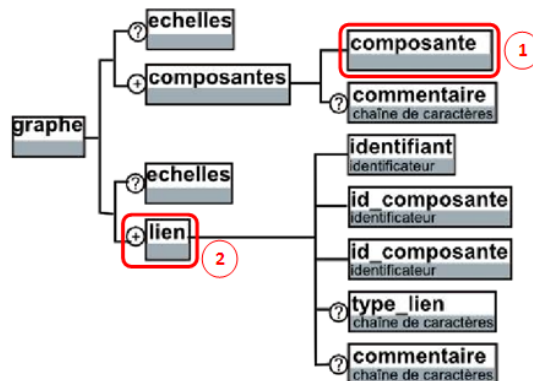


Figure 5-7 : Représentation graphique de l'élément *graphe* de PMDL [Eyssautier-Bavay 2008].

D. TYPE TEXTE

Enfin, dans certains profils, on trouve des informations sous forme de texte portant sur l'analyse de l'activité de l'apprenant (appréciations sur le travail de l'apprenant, conceptions erronées, etc.).

La Figure 5-8 montre une partie du profil d'élève issu de pratiques d'enseignants. Ce profil fournit un texte contenant l'appréciation de fin de trimestre faite par l'enseignant pour une élève de CE1.

Périodes	Commentaires de l'enseignant
septembre à décembre	Mariélys est très bien entrée dans les apprentissages du CE1 et ce premier trimestre est très satisfaisant surtout en mathématiques. Sa participation à l'oral reste timide. De plus, les difficultés à se concentrer posent des problèmes pour tous les exercices de copies qui prennent beaucoup trop de temps.

Figure 5-8 : Profil de l'apprenant issu de pratiques d'enseignants.

Dans le cas d'un élément *texte*, aucun trait de structure ne peut être dégagé, le contenu pouvant varier d'un apprenant à l'autre. Ainsi, les éléments de type *texte* sont indiqués dans la structure simplement à l'aide du marqueur *texte* (cf. Figure 5-1).

5.3.3. ÉCHELLES DANS PMDL

Les éléments *liste_composantes* et *graphe* contiennent des *echelles*, contenant à leur tour un ensemble d'*echelle*. Il y a autant d'élément *echelle* utilisés qu'il y aura de valeurs différentes pour une *composante* ou un *lien* (cf. Figure 5-9). Une *echelle* comporte un *nom*, un *identifiant*, éventuellement une *unite* et une *etiquette*. Ces deux derniers champs permettent de donner une signification pédagogique à l'*echelle*. Une *echelle* peut prendre la forme d'un *intervalle_valeurs* ou d'une *liste_valeurs*.

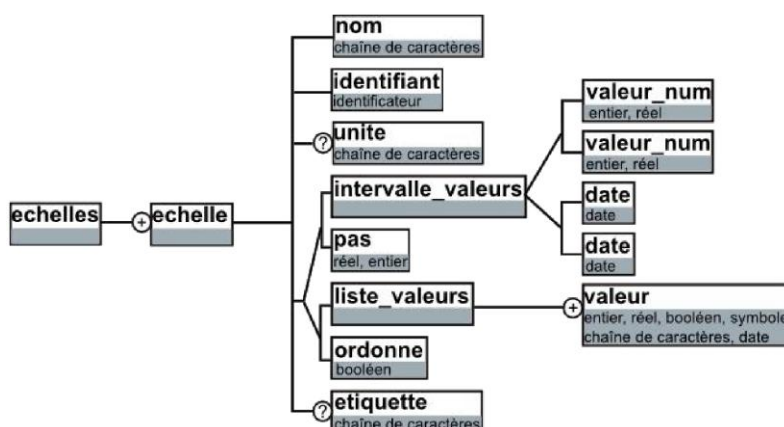


Figure 5-9 : Représentation graphique de l'élément *echelles* de PMDL [Eyssautier-Bavay 2008].

Un *intervalle_valeurs* est caractérisé par deux bornes, *valeur_num* ou *date*, et par un *pas* déterminant l'écart entre deux valeurs au sein de l'intervalle. Des exemples d'*echelle* représentant des intervalles peuvent être « 0..20 », indiquant que les notes sont entre 0 et 20, et « 13/09/07..30/09/07 » permettant de sélectionner la date à laquelle l'activité a été réalisée.

Une *liste_valeurs* peut être ordonnée ou non, ce qui est spécifié grâce au booléen *ordonne*, et contient au moins une *valeur*. Une *valeur* peut être un entier, un réel, une chaîne de caractères, un symbole, un booléen ou une date. Un exemple d'*echelle* ordonnée peut être « non maîtrisé, partiellement maîtrisé, maîtrisé », et un exemple d'*echelle* non ordonnée peut être « visuel, auditif, tactile ».

5.4. DÉFINITION FORMELLE DU MODÈLE cPMDL

Nous avons vu dans le chapitre précédent que pour décrire un modèle de personnalisation, il est nécessaire de définir des règles d'affectation. Ces règles d'affectation correspondent à des liens entre des contraintes sur profils CP_i permettant de sélectionner certaines valeurs du profil des apprenants, et des contraintes sur activités CA_i permettant d'agir sur la sélection des activités à fournir aux apprenants respectant les contraintes sur le profils (cf. Équation 4-3 page 85).

Pour définir le modèle cPMDL des contraintes sur le profil des apprenants, nous nous appuyons sur le langage de description de profils PMDL, présenté dans la section précédente. Ce langage permet de décrire les éléments d'un profil d'apprenant (*element*), chaque élément pouvant contenir une ou plusieurs valeurs (*valeur*) organisées soit dans une liste hiérarchique de composantes (*liste_composantes*), soit dans une liste à valeurs de répartition (*liste_repartition*), soit un graphe (*graphe*), soit dans du texte libre (*texte*).

Les valeurs contenues dans les types *liste_composantes* et *graphe* sont associées à des *echelles*. Les valeurs contenues dans le type *liste_repartition* sont des valeurs de comptage. Les valeurs contenues dans le type *texte* sont du texte libre, ne respectant aucune structure précise.

Pour les données décrites sous forme de *texte*, il n'est pas possible de définir de contraintes, puisqu'il est impossible de connaître l'organisation de ces données. Pour les autres types de données, il est possible de définir plusieurs types de contraintes sur profils : des contraintes portant sur une valeur CP_v , des contraintes portant sur un élément CP_e et des contraintes portant sur un nombre d'occurrences CP_o . Nous définissons à présent formellement ces trois types de contraintes avant d'en donner des exemples.

5.4.1. CONTRAINTES SUR PROFILS PORTANT SUR UNE VALEUR

Le but d'une contrainte sur profils portant sur une valeur est de contraindre une valeur précise présente dans le profil des apprenants. Ce type de contrainte permet de sélectionner les apprenants maîtrisant un élément non décomposable du profil.

Une contrainte sur profils portant sur une valeur CP_v est donc définie pour une valeur précise d'une composante terminale donnée (une *feuille*) ou d'un lien du profil. Ainsi, pour les types de données *liste_composantes* et *liste_repartition*, elle est définie au niveau des *composante* possédant le marqueur *feuille* (cf. ① et ③ sur la Figure 5-3). Pour le type *graphe*, elle est définie soit pour un *lien*, soit pour une *composante* (cf. ① et ② sur la Figure 5-7). Dans tous les cas, elle est définie pour une valeur précise, c'est-à-dire pour une seule des *echelles*. La forme de la contrainte va dépendre du type d'*echelle* à laquelle la valeur est associée. Ces *echelles* peuvent être (cf. Figure 5-9):

- des échelles que nous nommerons **textuelles**, contenant une liste énumérée de valeurs, regroupant les échelles ordonnées et non ordonnées du langage PMDL ;

- des échelles que nous nommerons **numériques**, contenant un intervalle de valeurs, portant soit sur des nombres, soit sur des dates ;
- un **entier positif**, pour les valeurs de comptage des éléments de type *liste_repartition*.

Les contraintes sur profils portant sur une valeur peuvent donc être de trois types, $CP_{V(T)}$, $CP_{V(N)}$, $CP_{V(E)}$, selon le type d'échelles associé à la valeur.

A. CONTRAINTE SUR PROFILS PORTANT SUR UNE VALEUR TEXTUELLE

Une contrainte sur profils portant sur une valeur textuelle $CP_{V(T)}$ a le même format, que la liste énumérée des valeurs soit ordonnée ou non. Ce format est le suivant :

$$CP_{V(T)} = \{Elt, (valeur_1, \dots, valeur_n)\}, 1 \leq n \leq \text{card}(liste_valeurs)$$

Équation 5-1 : Modèle d'une contrainte sur profils portant sur une valeur textuelle.

Définir une contrainte sur profils portant sur une valeur textuelle consiste à spécifier, pour un élément *Elt* du profil, un sous-ensemble de *valeur* appartenant à l'échelle définie avec PMDL. Ces valeurs peuvent être disjointes, c'est-à-dire qu'elles peuvent être choisies de façon non consécutive dans la *liste_valeurs*. Ce sous-ensemble de valeurs doit être non vide et peut couvrir tout le spectre de l'échelle précisée dans la structure de profil. Ainsi, *n* doit être supérieur ou égal à 1 et inférieur ou égal au cardinal de la *liste_valeurs*.

B. CONTRAINTE SUR PROFILS PORTANT SUR UNE VALEUR NUMÉRIQUE

Une contrainte sur profils portant sur une valeur numérique $CP_{V(N)}$ a le même format, que l'intervalle de valeurs porte sur des nombres, *valeur_num*, ou sur des dates, *date*. Ce format est le suivant :

$$CP_{V(N)} = \{Elt, ((i, b_i), (s, b_s))\}, i \leq s$$

Équation 5-2 : Modèle d'une contrainte sur profils portant sur une valeur numérique.

Définir une contrainte sur profils portant sur une valeur numérique consiste à spécifier, pour un élément *Elt* du profil, un intervalle de valeurs inclus dans l'intervalle initial. La contrainte peut porter sur l'une ou les deux bornes de l'intervalle. La borne *i* est la nouvelle borne inférieure de l'intervalle de valeurs, et la borne *s* la nouvelle borne supérieure. Les bornes *i* et *s* sont comprises entre les deux *valeur_num* ou *date* de l'échelle d'origine. Dans tous les cas, *s* doit être supérieure ou égale à *i*. *b_i* et *b_s* sont les booléens indiquant si les bornes sont incluses ou non dans la contrainte.

C. CONTRAINTE SUR PROFILS PORTANT SUR UN ENTIER POSITIF

Une contrainte sur profils portant sur un entier positif $CP_{V(E)}$ est un cas particulier des contraintes sur profils portant sur une valeur numérique $CP_{V(N)}$. Leurs formats sont donc presque identiques :

$$CP_{V(E)} = \{Elt, ((i, b_i), (s, b_s))\}, i \in \mathbb{N}, i \leq s$$

Équation 5-3 : Modèle d'une contrainte sur profils portant sur un entier positif.

Les différences portent d'une part sur les bornes i et s qui doivent être des entiers positifs, i pouvant être égale à zéro, et d'autre part sur s qui doit toujours être supérieure à i , mais peut cette fois aller jusqu'à l'infini. b_i et b_s sont toujours des booléens indiquant si les bornes sont incluses ou non dans la contrainte. Dans le cas où la borne s est égale à l'infini, le booléen b_s prend la valeur *false*.

Il faut noter que certains intervalles portant sur des nombres entiers peuvent être équivalents mais avoir des définitions différentes. Par exemple les intervalles $[1, 3[$ et $]0, 2]$ sont équivalents lorsque les valeurs à choisir sont des entiers, mais leurs définitions sont bien différentes.

D. EXEMPLES DE CONTRAINTES SUR PROFILS PORTANT SUR UNE VALEUR

Lors des définitions de contraintes sur profils portant sur une valeur, nous avons fourni des exemples de contraintes sans lien avec un profil d'apprenant. Nous illustrons maintenant ces contraintes en définissant des contraintes sur l'exemple du profil exprimé avec PMDL fourni en Annexe B page 285.

Dans ce profil, l'élément E1 « Algèbre » contient une liste hiérarchique à deux niveaux. Pour chacune des feuilles de cette liste, il y a deux valeurs, l'une exprimée selon l'échelle « notes de 0 à 10 » et l'autre selon l'échelle « pourcentage ». Nous pouvons définir sur la première de ces feuilles la contrainte $CP_{V(N)}$ de la Figure 5-10 qui permet de sélectionner les apprenants n'ayant pas la moyenne à la compétence « Maîtrise de l'addition des nombres entiers ».

$$CP_{V(N)} = \{E1_C2_valeur1, ((0, true), (5, false))\}$$

Figure 5-10 : Contrainte sur profils portant sur une valeur numérique.

$$CP_{V(E)} = \{E3_C6_valeur2, ((5, false), (\infty, false))\}$$

Figure 5-11 : Contrainte sur profils portant sur un entier positif.

L'élément E3 « Résolution de divisions » contient une liste de composantes à un seul niveau. Chaque élément de cette liste contient deux nombres positifs, le premier indique la valeur de répartition et le second le nombre de réussites. Nous pouvons définir sur la deuxième de ces feuilles la contrainte $CP_{V(E)}$ de la Figure 5-11 qui permet de sélectionner les apprenants faisant plus de cinq erreurs de type « Erreur de calcul ».

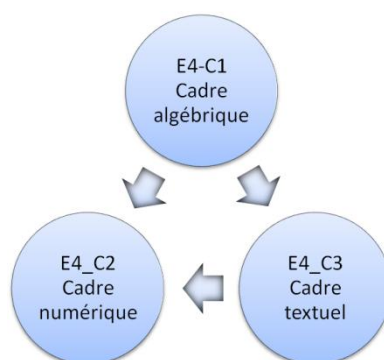


Figure 5-12 : Représentation graphique de l'élément E4 « Mathématiques » du profil exprimé avec PMDL de l'Annexe B.

L'élément E4 « Mathématiques » contient un graphe ayant trois sommets et trois liens. Cet élément peut être représenté par la Figure 5-12. Les sommets de ce graphe sont évalués par une seule valeur d'échelle « pourcentage » et les liens sont évalués par une seule valeur respectant l'échelle « Textuelle 3 niveaux ». Nous pouvons définir sur le sommet « Cadre algébrique » la contrainte $CP_{V(N)}$ de la Figure 5-13 qui permet de sélectionner les apprenants ayant un taux de réussite inférieur ou égal à 50% pour cette compétence. Nous pouvons de plus définir sur l'arc

entre les sommets E4_C1 et E4_C3, la contrainte $CP_{V(T)}$ de la Figure 5-14 qui permet de sélectionner les apprenants maîtrisant bien ou très bien le passage du cadre algébrique au cadre textuel.

$$CP_{V(N)} = \{E4_C1_valeur1, ((0, true), (0.5, true))\}$$

Figure 5-13 : Contrainte sur profils portant sur une valeur numérique.

$$CP_{V(T)} = \{E4_L2, (Bien, Très Bien)\}$$

Figure 5-14 : Contrainte sur profils portant sur une valeur textuelle.

5.4.2. CONTRAINTE SUR PROFILS PORTANT SUR UN ÉLÉMENT

Le but d'une contrainte sur profils portant sur un élément est de contraindre la valeur d'un élément composé du profil des apprenants. Ce type de contraintes permet de sélectionner les apprenants maîtrisant un élément décomposé en plusieurs sous-éléments.

Une contrainte sur profils portant sur un élément CP_E est définie pour un élément donné du profil contenant plusieurs valeurs. Cet élément peut être :

- une *feuille* des types de données *liste_composantes* et *liste_repartition* contenant plusieurs valeurs (cf. ① et ③ sur la Figure 5-3). Cet élément correspond au « cas 1 » sur la Figure 5-15;
- un *lien* ou une *composante* du type *graphe* contenant plusieurs valeurs (cf. ① et ② sur la Figure 5-7). Ces éléments correspondent aux deux « cas 2 » sur la Figure 5-15;
- une *composante* des types *liste_composantes* et *liste_repartition* possédant des *sous_composantes*, avec une valeur (cf. ② et ④ sur la Figure 5-3). Cet élément correspond au « cas 3 » sur la Figure 5-15;
- une *composante* des types *liste_composantes* et *liste_repartition* possédant des *sous_composantes*, avec plusieurs valeurs (cf. ② et ④ sur la Figure 5-3). Cet élément correspond au « cas 4 » sur la Figure 5-15;
- un *element* possédant des données de type *liste_composantes*, *liste_repartition* ou *graphe*, chaque donnée étant décrite avec une valeur (cf. ① sur la Figure 5-1). Cet élément correspond au « cas 5 » sur la Figure 5-15;
- un *element* possédant des données de type *liste_composantes*, *liste_repartition* ou *graphe*, chaque donnée étant décrite avec plusieurs valeurs (cf. ① sur la Figure 5-1). Cet élément correspond aux deux « cas 6 » sur la Figure 5-15.

Une contrainte sur profils portant sur un élément contraint donc la valeur d'un élément du profil. Or cet élément n'a pas de valeur propre. Celle-ci va donc être calculée en fonction de certaines ou de toutes les valeurs de ses sous-éléments. Par exemple, dans le « cas 4 » de la Figure 5-15, il est possible de définir la valeur de la composante soit à partir des deux valeurs de chacune de ses sous-composantes, soit à partir d'une seule des deux valeurs de ses sous-composantes. Quand plusieurs valeurs sont utilisées, et qu'elles ne sont pas toutes définies à partir de la même échelle, pour pouvoir faire des calculs dessus, il faut les convertir dans une échelle unique.

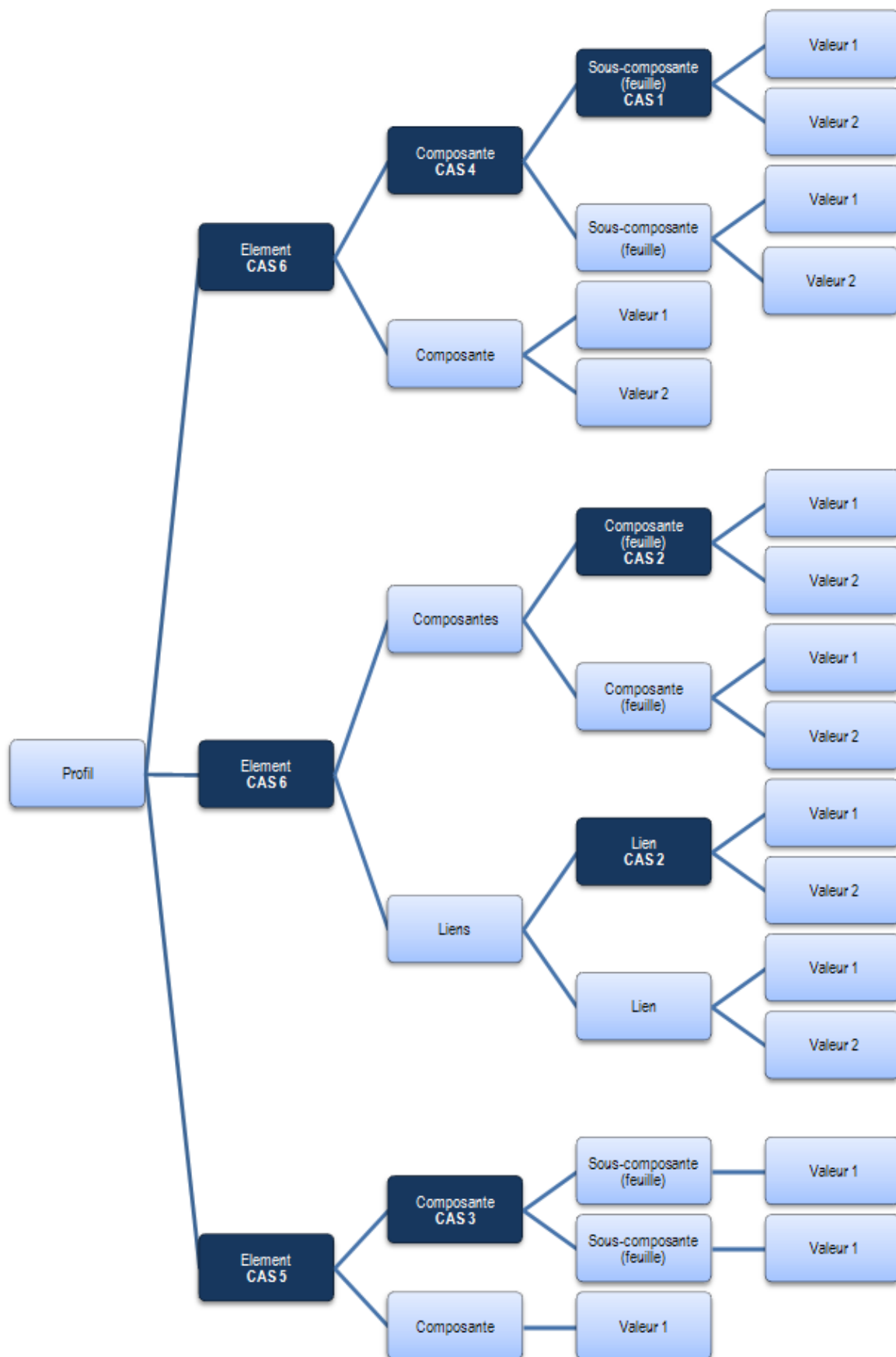


Figure 5-15 : Exemple d'un profil fictif illustrant les différents éléments de PMDL sur lesquels il est possible de définir une contrainte sur profils portant sur un élément.

Définir une contrainte sur profils portant sur un élément consiste donc à spécifier, pour un élément *Elt* du profil, les valeurs à partir desquelles la valeur de l'élément sera calculée et donc leurs *echelle*, l'échelle résultante *Ech_R* dans laquelle doivent être converties toutes les valeurs, l'opération *Op* permettant de combiner ces valeurs ainsi qu'une contrainte sur la valeur résultant de l'opération du type contrainte sur profils portant sur une valeur CP_V .

Quel que soit l'élément du profil sur lequel est définie la contrainte sur profils portant sur un élément, le format de la contrainte est toujours le suivant :

$$CP_E = \{Elt, (echelle_1, \dots, echelle_n), Ech_R, Op, CP_V\}$$

Équation 5-4 : Modèle d'une contrainte sur profils portant sur un élément.

Le type de l'échelle résultante *Ech_R* dépend du nombre de valeurs choisies et du type de l'*echelle* de chacune. La conversion permettant de passer des échelles d'origine à l'échelle résultante est faite automatiquement grâce à un ensemble de formules fournies en Annexe C page 289. Nous ne les détaillons pas ici puisqu'elles ne sont pas nécessaires pour définir une contrainte sur profils portant sur un élément. Il est seulement utile de connaître les types de conversions permis pour définir la contrainte. La Figure 5-16 résume les différentes possibilités que nous expliquons par la suite.

	Nombre de valeurs	Type(s) des échelles d'origine	Type de l'échelle résultante
1	1	Entier positif	Entier positif
2	1	Textuelle non ordonnée	Textuelle non ordonnée
3	1	Textuelle ordonnée	Textuelle ordonnée ou numérique
4	1	Numérique	Textuelle ordonnée ou numérique
5	n	Entier(s) positif(s) et autre(s) type(s)	-
6	n	Textuelle(s) non ordonnée(s) et autre(s) type(s)	-
7	n	Textuelles ordonnées	Textuelle ordonnée ou numérique
8	n	Numériques	Textuelle ordonnée ou numérique
9	n	Textuelle(s) ordonnée(s) et numérique(s)	Textuelle ordonnée ou numérique

Figure 5-16 : Types possibles de l'échelle résultante dans une contrainte sur profils portant sur un élément, en fonction des échelles des valeurs contenues dans les sous-éléments.

Si la valeur de l'élément *Elt* est calculée à partir d'une seule valeur de chacun des sous-éléments, l'échelle des sous-éléments est toujours du même type. Tout d'abord, la valeur de chaque sous-élément peut être un entier positif (cf. 1 sur la Figure 5-16), dans ce cas, l'échelle résultante sera forcément un entier positif. En effet, il est impossible de convertir automatiquement une valeur non bornée, comme un nombre entier, dans une échelle numérique possédant des bornes, ou dans une échelle textuelle possédant un nombre fini de valeurs. De la même façon, quand la valeur des sous-éléments appartient à une échelle textuelle non ordonnée (cf. 2 sur la

Figure 5-16), l'échelle résultante sera la même échelle textuelle, les conversions d'une échelle textuelle non ordonnée vers un autre type d'échelles n'étant pas possible. Par exemple, il n'est pas possible de convertir automatiquement l'échelle textuelle non ordonnée « Bavard, Passif, Attentif » en échelle numérique « 0..10 » ou en l'échelle textuelle ordonnée « ABCDEF ». Enfin, si la valeur des sous-éléments appartient à une échelle textuelle ordonnée ou à une échelle numérique (cf. 3 et 4 sur la Figure 5-16), l'échelle résultante pourra être indifféremment une échelle textuelle ordonnée ou une échelle numérique.

Si la valeur de l'élément E_{lt} est calculée à partir de plusieurs ou de toutes les valeurs de chacun des sous-éléments, les échelles peuvent être de différents types. Si une des valeurs est un entier positif (cf. 5 sur la Figure 5-16) ou si elle est définie selon une échelle textuelle non ordonnée (cf. 6 sur la Figure 5-16), il n'est pas possible de définir de contrainte puisqu'il n'est pas possible de combiner ces valeurs avec des valeurs définies dans d'autres échelles, pour les raisons présentées dans le paragraphe précédent. Sinon, que les échelles soient toutes des échelles textuelles ordonnées, toutes des échelles numériques ou pour une partie des échelles textuelles ordonnées et pour une autre partie des échelles numériques, l'échelle résultante peut être indifféremment une échelle textuelle ordonnée ou une échelle numérique.

L'opération Op permettant de combiner les différentes valeurs des sous-éléments dépend du type de l'échelle résultante. Si cette échelle est une échelle textuelle, ordonnée ou non, l'opération sera forcément une moyenne. Si c'est une échelle numérique ou de type entier positif, l'opération pourra être une somme ou une moyenne.

La contrainte sur profils portant sur une valeur CP_v respectera le format associé au type de l'échelle résultante. Pour une échelle textuelle, ordonnée ou non, le format sera celui d'une contrainte sur profils portant sur une valeur textuelle $CP_{V(T)}$ (cf. Équation 5-1). Pour une échelle numérique, le format sera celui d'une contrainte sur profils portant sur une valeur numérique $CP_{V(N)}$ (cf. Équation 5-2). Pour un entier positif, le format sera celui d'une contrainte sur profils portant sur un entier positif $CP_{V(E)}$ (cf. Équation 5-3).

Pour illustrer cette définition d'une contrainte sur profils portant sur un élément, reprenons l'exemple du profil exprimé avec PMDL fourni en Annexe B page 285.

Sur la composante $E1_C1$ « Maîtrise de l'addition » de l'élément $E1$ « Algèbre », nous pouvons définir la contrainte CP_{E1} permettant de sélectionner les apprenants ne maîtrisant pas l'addition en général. Pour cela, il faut regarder la première valeur de toutes les sous-composantes de la composante $E1_C1$, faire la moyenne de toutes ces valeurs et regarder si le résultat est inférieur à 5/10 (c'est-à-dire compris dans l'intervalle $[0, 5[$). La contrainte CP_{E1} est présentée sur la Figure 5-17. Dans cette contrainte, X sera remplacé par le résultat de la moyenne des premières valeurs des sous-éléments contenus dans la composante $E1_C1$.

$$CP_{E1} = \{E1_C1, (\text{« notes de 0 à 10 »}, \text{« notes de 0 à 10 »}, \text{moyenne}, \{X, ((0, \text{true}), (5, \text{false}))\})\}$$

Figure 5-17 : Contrainte sur profils portant sur un élément.

Sur l'élément $E4$ « Mathématiques » (cf. Figure 5-12), nous pouvons définir la contrainte CP_{E2} permettant de sélectionner les apprenants ayant un taux de maîtrise des mathématiques supérieur à 50%. Pour cela, il faut faire la moyenne de tous les sommets du graphe, ainsi que la moyenne de tous les liens. La contrainte CP_{E2} est présentée sur la Figure 5-18. Dans cette contrainte, X sera remplacé par le résultat de la moyenne de la valeur de chaque composante et de chaque lien contenus dans l'élément $E4$.

$$CP_{E2} = \{E4, (\text{« pourcentage »}, \text{« Textuelle 3 niveaux »}), \text{« pourcentage »}, \text{moyenne}, \{X, ((0, \text{true}), (0.5, \text{false}))\}\}$$

Figure 5-18 : Contrainte sur profils portant sur un élément.

5.4.3. CONTRAINTE SUR PROFILS PORTANT SUR UN NOMBRE D'OCCURRENCES

Le but d'une contrainte sur profils portant sur un nombre d'occurrences est de contraindre le nombre de sous-éléments respectant chacun une contrainte sur profil. Une contrainte sur profils portant sur un nombre d'occurrences peut par exemple permettre de sélectionner les apprenants maîtrisant 3 des 4 sous-éléments d'un élément donné de leur profil.

Une contrainte sur profils portant sur un nombre d'occurrences CP_o est définie pour un élément donné du profil contenant plusieurs sous-éléments. Cet élément peut être du même type que les éléments d'une contrainte sur profils portant sur un élément (cf. section 5.4.2).

Définir une contrainte sur profils portant sur un nombre d'occurrences consiste donc à spécifier d'une part, pour un élément Elt du profil, les sous-éléments $SsElt_i$ à prendre en compte, et pour chaque sous-élément une contrainte sur profils CP_i , et d'autre part, l'intervalle de valeurs indiquant combien de sous-éléments doivent respecter leur contrainte sur profil.

Quel que soit l'élément du profil sur lequel est définie la contrainte sur profils portant sur un nombre d'occurrences, le format de la contrainte est toujours le suivant :

$$CP_o = \{Elt, ((SsElt_1, CP_1), \dots, (SsElt_n, CP_n)), ((i, b_i), (s, b_s))\},$$

$$n \geq 2, i, s \in \mathbb{N}^*, i \leq s, s \leq n$$

Équation 5-5 : Modèle d'une contrainte sur profils portant sur un nombre d'occurrences.

Une contrainte doit porter sur au moins deux sous-éléments $SsElt_i$. Ces sous-éléments peuvent être soit des valeurs, comme pour les cas 1 et 2 de la Figure 5-15, la contrainte sur profils CP_i respectera alors le format d'une contrainte sur profils portant sur une valeur (cf. section 5.4.1), soit des éléments composés d'autres éléments, comme pour les cas 3, 4, 5 et 6 de la Figure 5-15, la contrainte sur profils CP_i respectera alors le format d'une contrainte sur profils portant sur un élément (cf. section 5.4.2).

L'intervalle $((i, b_i), (s, b_s))$ permet de définir le nombre minimum et le nombre maximum de sous-éléments qui doivent respecter les contraintes sur profils. La borne inférieure i doit être un entier compris entre zéro et le nombre de sous-éléments. La borne supérieure s doit être un entier compris entre i et le nombre de sous-éléments. Les booléens b_i et b_s indiquent si les bornes sont incluses ou non dans l'intervalle.

Pour illustrer cette définition d'une contrainte sur profils portant sur un nombre d'occurrences, reprenons l'exemple du profil exprimé avec PMDL fourni en Annexe B page 285.

Sur la composante $E1_C1$ « Maîtrise de l'addition » de l'élément $E1$ « Algèbre », nous pouvons définir la contrainte CP_{R1} permettant de sélectionner les apprenants ne maîtrisant pas deux des trois sous-composantes de la composante. Pour cela, il faut regarder la première valeur de toutes les sous-composantes de la composante $E1_C1$. Cette valeur est toujours exprimée selon l'échelle numérique « notes de 0 à 10 ». La contrainte sur profils définie pour les sous-éléments est donc une contrainte sur profils portant sur une valeur numérique $CP_{V(N)}$ permettant de

sélectionner les apprenants ne maîtrisant pas le sous-élément, c'est-à-dire ayant une note comprise dans l'intervalle $[0, 5[$. La contrainte CP_{R1} est présentée sur la Figure 5-19. Dans cette contrainte, X sera remplacé par toutes les premières valeurs des sous-éléments contenues dans la composante E1_C1.

$$CP_{R1} = \{E1_C1, (\text{« notes de 0 à 10 »}, \{X, ((0, \text{true}), (5, \text{false}))\}), ((2, \text{true}), (2, \text{true}))\}$$

Figure 5-19 : Contrainte sur profils portant sur un nombre d'occurrences.

Sur le graphe de l'élément E4 « Mathématiques » (cf. Figure 5-12), nous pouvons définir la contrainte CP_{R2} permettant de sélectionner les apprenants ne maîtrisant pas un des passages d'une composante à une autre. Pour cela, il faut regarder les valeurs portées par les liens du graphe. Ces valeurs sont exprimées selon l'échelle textuelle « Textuelle 3 niveaux », la contrainte portant sur la valeur des sous-éléments est donc une contrainte sur profils portant sur une valeur textuelle $CP_{V(T)}$ permettant de sélectionner les apprenants ne maîtrisant pas un lien, c'est-à-dire ayant la valeur « Assez bien » ou la valeur « Bien » mais pas la valeur « Très bien ». Il faut ensuite définir l'intervalle permettant de sélectionner les apprenants qui ont au moins un lien respectant la contrainte $CP_{V(T)}$, c'est-à-dire l'intervalle $[1, +\infty[$. La contrainte CP_{R2} est présentée sur la Figure 5-20. Dans cette contrainte, X sera remplacé par tous les liens du graphe de l'élément E4.

$$CP_{R2} = \{E4_Liens, (\text{« Textuelle 3 niveaux »}, \{X, (\text{Assez Bien; Bien})\}), ((1, \text{true}), (\infty, \text{false}))\}$$

Figure 5-20 : Contrainte sur profils portant sur un nombre d'occurrences.

5.4.4. COMBINAISON DE CONTRAINTES SUR PROFILS

Nous venons de voir les différentes contraintes permettant de sélectionner une partie du profil des apprenants. Il est parfois nécessaire de combiner ces contraintes pour sélectionner plusieurs parties du profil.

Ainsi, le modèle de personnalisation permet de définir des expressions booléennes combinant les contraintes sur profils CP (cf. section 4.3). Quel que soit le type des contraintes sur profils (portant sur une valeur, un élément ou un nombre d'occurrences), elles peuvent être combinées grâce aux opérateurs de conjonction « ET », de disjonction « OU », de négation « NON » et aux parenthèses « () ».

Pour illustrer ces combinaisons de contraintes sur profils, reprenons l'exemple du profil exprimé avec PMDL fourni en Annexe B page 285.

L'élément E4 « Mathématiques » contient un graphe ayant trois sommets et trois liens (cf. Figure 5-12). Les sommets de ce graphe sont évalués par une seule valeur respectant l'échelle « pourcentage » et les liens sont évalués par une seule valeur respectant l'échelle « Textuelle 3 niveaux ». Nous pouvons définir une contrainte permettant de sélectionner les apprenants maîtrisant les trois sommets du graphe, mais ne maîtrisant pas au moins l'un des trois arcs. Pour cela, nous définissons l'expression :

$$CP = (CP_{V(N)1} \text{ ET } CP_{V(N)2} \text{ ET } CP_{V(N)3}) \text{ ET } (CP_{V(T)1} \text{ OU } CP_{V(T)2} \text{ OU NON}(CP_{V(T)3}))$$

Figure 5-21 : Exemple de combinaison de contraintes sur profils.

Les contraintes $CP_{V(N)1}$, $CP_{V(N)2}$, $CP_{V(N)3}$ sont des contraintes sur profils portant sur une valeur numérique permettant de sélectionner les apprenants maîtrisant les trois sommets du graphe.

Pour cet exemple, nous considérons que la maîtrise d'un sommet équivaut à un taux de réussite d'au moins 70%. La définition de ces contraintes est donc la suivante :

$$\begin{array}{|l} \text{CP}_{V(N)1} = \{ \text{E4_C1_valeur1}, \\ ((0.7, \text{true}), (1, \text{true})) \} \end{array} \quad \begin{array}{|l} \text{CP}_{V(N)2} = \{ \text{E4_C2_valeur1}, \\ ((0.7, \text{true}), (1, \text{true})) \} \end{array} \quad \begin{array}{|l} \text{CP}_{V(N)3} = \{ \text{E4_C3_valeur1}, \\ ((0.7, \text{true}), (1, \text{true})) \} \end{array}$$

Figure 5-22 : Contraintes sur profils portant sur une valeur numérique utilisées dans la contrainte de la Figure 5-21.

Les contraintes $\text{CP}_{V(T)1}$, $\text{CP}_{V(T)2}$, $\text{CP}_{V(T)3}$ sont des contraintes sur profils portant sur une valeur textuelle permettant de sélectionner les apprenants ne maîtrisant pas les arcs du graphe. Nous considérons que la non maîtrise d'un arc équivaut à la valeur « Assez Bien » de l'échelle « Textuelle 3 niveaux ». Pour mémoire, cette échelle correspond à la liste énumérée « Assez bien, Bien, Très bien ». La définition de ces contraintes est donc la suivante :

$$\begin{array}{|l} \text{CP}_{V(T)1} = \{ \text{E4_L1}, (\text{Assez} \\ \text{bien}) \} \end{array} \quad \begin{array}{|l} \text{CP}_{V(T)2} = \{ \text{E4_L2}, (\text{Assez} \\ \text{bien}) \} \end{array} \quad \begin{array}{|l} \text{CP}_{V(T)3} = \{ \text{E4_L3}, (\text{Bien}, \\ \text{Très bien}) \} \end{array}$$

Figure 5-23 : Contraintes sur profils portant sur une valeur textuelle utilisées dans la contrainte de la Figure 5-21.

5.5. CE QU'IL FAUT RETENIR

Le modèle de personnalisation, présenté dans le chapitre précédent, permet de définir des règles d'affectation contenant des contraintes sur profils (cf. section 4.3). Ces contraintes sur profils permettent de contraindre certaines parties du profil des apprenants pour décider à quel apprenant les activités de la règle d'affectation doivent être données.

Dans ce chapitre, nous avons présenté le modèle cPMDL, une extension du langage PMDL contenant les différents formats que peuvent avoir ces contraintes sur profils. Elles peuvent porter sur une valeur, un élément ou un nombre d'occurrences et peuvent être combinées à partir des opérateurs booléens « ET, OU, NON, () ». Pour définir ces formats, nous nous appuyons sur le langage de modélisation des profils PMDL [Eyssautier-Bavay 2008].

Les contraintes sur profils portant sur une valeur CP_V permettent de contraindre la valeur d'un élément contenant une valeur dans le profil des apprenants. Par exemple, sélectionner les apprenants ayant un taux de maîtrise d'une compétence élémentaire donnée compris entre 30 et 50 %. Ces contraintes peuvent être de trois types : portant sur une valeur textuelle $CP_{V(T)}$, portant sur une valeur numérique $CP_{V(N)}$ ou portant sur un entier positif $CP_{V(E)}$.

Les contraintes sur profils portant sur un élément CP_E permettent de contraindre la valeur d'un élément ne contenant pas de valeur dans le profil des apprenants. Par exemple, sélectionner les apprenants ayant un taux de maîtrise d'une compétence donnée, décrite à l'aide de plusieurs sous-compétences, compris entre 30 et 50 %. Pour cela, il est nécessaire de spécifier comment calculer la valeur de l'élément puis de la contraindre avec une contrainte sur profils portant sur une valeur.

Les contraintes sur profils portant sur un nombre d'occurrences permettent de contraindre le nombre de sous-éléments d'un élément donné devant respecter chacun une contrainte sur profils. Par exemple, sélectionner les apprenants maîtrisant deux ou trois des quatre méthodes permettant d'effectuer une tâche donnée. Pour cela, il est nécessaire de choisir les sous-éléments concernés, de spécifier pour chacun une contrainte sur profils et de borner le nombre minimum et le nombre maximum de contraintes sur les sous-éléments à valider.

CHAPITRE 6. GEPPETO : DES MODÈLES ET PROCESSUS POUR ADAPTER DES ACTIVITÉS PÉDAGOGIQUES

*Comment prendre en compte
les besoins et habitudes pédagogiques
des enseignants ?*

PLAN DU CHAPITRE

6.1.	Introduction.....	117
6.2.	Principe de l'approche GEPPETO.....	117
6.3.	GEPPETO _p : Adapter des activités papier	118
6.3.1.	Personnaliser quoi et comment ?.....	119
6.3.2.	Classification des activités selon la façon de les générer	120
a.	Travail sur texte.....	121
b.	Travail sur illustration.....	123
c.	Organisation d'éléments	124
d.	Tableau à double entrée.....	124
e.	Expression algébrique	125
f.	Problème scientifique	126
g.	Démonstration	126
h.	Questions / réponses.....	127
6.3.3.	Principe de l'approche GEPPETO _p	127
a.	Exemple d'utilisation.....	129
b.	Structure de patrons.....	129
c.	Patrons d'exercices	131
d.	générateurs semi-automatiques.....	141
6.3.4.	Bilan sur l'adaptation des activités papier	148
6.4.	GEPPETO _s : adapter des EIAH	149
6.4.1.	Personnaliser quoi et comment ?.....	149
6.4.2.	Classification des EIAH selon leur personnalisation	152
6.4.3.	Principe de l'approche GEPPETO _s	153
a.	Exemples d'utilisation.....	154
b.	Définition formelle du méta-modèle AKEPI	160
b.1.	Le modèle de description pédagogique	161
b.2.	Le modèle de description technique	165
c.	Processus AKASI.....	167
d.	Exemples de modèles OKEP/x.....	168
d.1.	Le modèle OKEP/ ABALECT	168
d.2.	Le modèle OKEP/ AMBRE-ADD	171
e.	Processus OPIKSI	174
6.4.4.	Bilan sur l'adaptation d'EIAH.....	175
6.5.	Ce qu'il faut retenir	177

PUBLICATIONS RELATIVES À GEPPETO_P

- [Lefevre et al. 2007a] Lefevre, M., Jean-Daubias, S., Guin, N. « Génération de feuilles d'exercices adaptées aux profils d'apprenants dans le cadre du projet PERLEA ». Poster, 3ème Conférence en Environnements Informatiques pour l'Apprentissage Humain (EIAH'2007), Lausanne, Suisse, 27-29 juin 2007.
- [Lefevre et al. 2007c] Lefevre, M., Jean-Daubias, S., Guin, N. « Génération d'exercices au sein du projet PERLEA », Rapport de recherche RR-LIRIS-2007-030, LIRIS, Villeurbanne, France, 2007.
- [Lefevre et al. 2008c] Lefevre, M., Jean-Daubias, S., Guin, N. « Generation of exercises within the PERLEA project », Rapport de recherche RR-LIRIS-2008-011, LIRIS, Villeurbanne, France, 2008.
- [Jean-Daubias et al. 2009c] Jean-Daubias, S., Lefevre, M., Guin, N. « Generation of exercises within the PERLEA project ». 2nd Workshop on Question Generation, International Conférence on Artificial Intelligence in Education (AIED 2009), Brighton, Grande-Bretagne, AIED 2009 Workshops Proceedings 1, pp. 38-42, 6-7 juillet 2009.
- [Lefevre et al. 2009d] Lefevre, M., Jean-Daubias, S., Guin, N. « Generation of pencil and paper exercises to personalize learners' work sequences: typology of exercises and meta-architecture for generators ». E-Learn 2009 (World Conference on E-Learning in Corporate, Government, Healthcare & Higher Education), Vancouver, Canada, 26-30 octobre 2009.
- [Jean-Daubias et al. 2009e] Jean-Daubias, S., Lefevre, M., Guin, N. « Generation of paper and pencil activities within the PERLEA project », Rapport de recherche RR-LIRIS-2009-016 (version longue de l'article du Workshop on Question Generation, AIED 2009), LIRIS, Villeurbanne, France, 2009.

PUBLICATIONS RELATIVES À GEPPETO_S

- [Lefevre et al. 2009b] Lefevre, M., Jean-Daubias, S., Guin, N. « Supporting Acquisition of Knowledge to Personalize Interactive Learning Environments through a Meta-Model ». 17th International Conference on Computers in Education (ICCE 2009), Hong Kong, 30 novembre - 4 décembre 2009.
- [Lefevre et al. 2009e] Lefevre, M., Mille, A., Jean-Daubias, S., Guin, N. « A Meta-Model to Acquire Relevant Knowledge for Interactive Learning Environments Personalization ». Adaptive 2009, Athènes, Grèce, 15-20 novembre 2009.

6.1. INTRODUCTION

Les contraintes sur les activités permettent de sélectionner ou de créer une activité correspondant aux habitudes pédagogiques d'un enseignant. Par activité, nous entendons d'une part des exercices papier à imprimer, d'autre part des activités sur un logiciel pédagogique, mais aussi la configuration de l'environnement logiciel. De la même façon que pour les profils, pour contraindre la sélection ou la création d'une activité, il est nécessaire de connaître le modèle de cette activité. Nous proposons pour cela l'approche GEPPETO permettant d'obtenir les modèles des activités à personnaliser, de contraindre ces modèles et de générer les activités en respectant ces contraintes. L'approche GEPPETO est générique, nous l'avons déclinée selon deux points de vue : papier et EIAH.

Dans ce chapitre, nous présentons l'approche GEPPETO que nous proposons pour personnaliser les activités. Nous la détaillons ensuite pour les activités papier, puis pour les activités effectuées sur des logiciels pédagogiques. Nous terminons par une synthèse soulignant les points communs entre les deux déclinaisons de l'approche GEPPETO.

6.2. PRINCIPE DE L'APPROCHE GEPPETO

Pour pouvoir adapter des activités pédagogiques, il est nécessaire de contraindre le choix ou la génération de ces activités. Pour pouvoir exprimer ces contraintes sur l'activité, il est nécessaire de disposer d'un modèle de l'activité² à personnaliser. Comme il n'est pas possible d'avoir un modèle générique pour n'importe quelle activité à personnaliser, il s'agit plutôt de proposer un méta-modèle qui guidera la modélisation de l'activité à personnaliser.

Nous proposons l'approche GEPPETO (GENerics models and processes to Personalize learners' PEDagogical activities according to Teaching Objectives). Cette approche propose des modèles et des processus permettant de personnaliser les activités pédagogiques proposées aux apprenants en respectant les objectifs pédagogiques de chaque agent de la personnalisation.

Dans cette approche, un **méta-modèle d'activités** (cf. ① sur la Figure 6-1) contraint l'expression des connaissances qu'un expert doit fournir pour pouvoir décrire des activités pédagogiques. Ces connaissances, une fois fournies par l'expert, forment soit le modèle d'une activité papier, soit le modèle d'un EIAH permettant sa personnalisation. À partir **d'un modèle d'activités** (cf. ② sur la Figure 6-1), il est possible de définir des contraintes permettant de contraindre le choix des activités. Ces **contraintes sur activités** (cf. ③ sur la Figure 6-1) sont définies par chaque agent en fonction de ses besoins de personnalisation. Ces contraintes sont ensuite interprétées par un système pour personnaliser les **activités pédagogiques** fournies aux apprenants (cf. ④ sur la Figure 6-1).

² Lorsque nous parlons de modèle de l'activité, il s'agit d'un modèle informatique formel, non exhaustif, contenant ce qu'il est nécessaire de savoir de l'activité pour la personnaliser. Il faut noter que cette acception diffère de celle utilisée en Sciences Humaines qui correspond à des modèles de l'activité complets mais non formels [Engeström 1987, Kaptelinin 1996, Béguin et al. 2000] cités dans [Bourguin 2000].

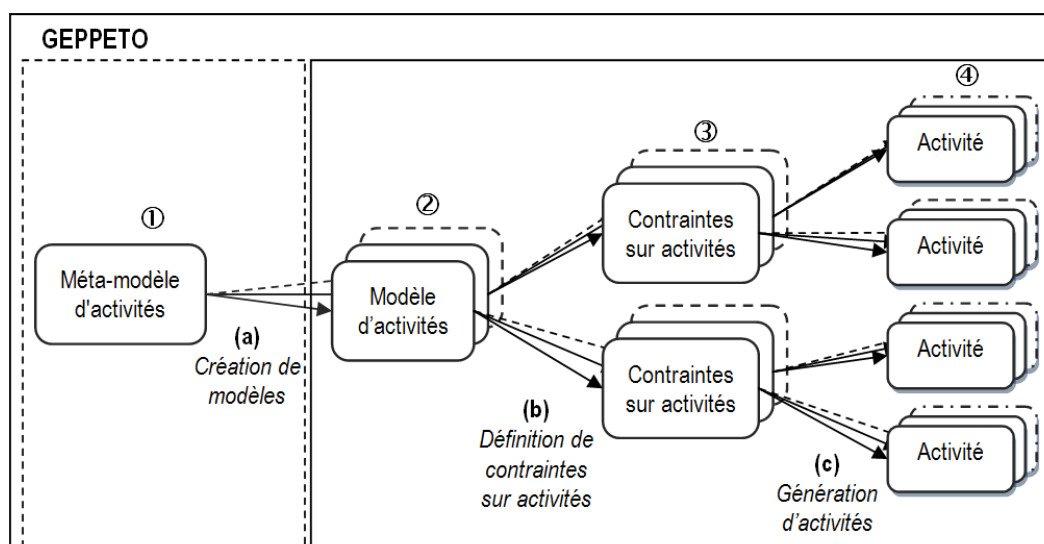


Figure 6-1 : Représentation graphique de l'approche GEPPETO.

Les processus GEPPETO se décomposent selon deux phases. La première, représentée par un cadre en pointillés sur la Figure 6-1, correspond à une phase d'initialisation. Elle n'est effectuée qu'une seule fois. Dans cette phase, un expert utilise le méta-modèle d'activités pour définir, pour chaque activité papier ou de l'EIAH que les agents, enseignants ou systèmes, souhaitent personnaliser, un modèle d'activités. Ce **processus de création de modèles d'activités** est représenté par les flèches (a) sur la Figure 6-1.

La seconde phase, représentée par un cadre plein sur la Figure 6-1, est celle de l'utilisation courante de GEPPETO. Dans cette phase, un agent se sert d'un modèle d'activités créé par l'expert pour contraindre la personnalisation des activités proposées à ses apprenants. Ce **processus de définition de contraintes sur activités** (cf. b sur la Figure 6-1) se décompose en deux étapes : la proposition d'une interface adaptée au modèle d'activités permettant de limiter les possibilités lors de la définition des contraintes sur activités, et l'enregistrement de ces contraintes. Ensuite, les contraintes sur activités sont utilisées pour personnaliser les activités pédagogiques proposées aux apprenants. Ce **processus de génération d'activités** (cf. c sur la Figure 6-1) se décompose lui aussi en deux étapes : l'utilisation des contraintes pour créer une activité et la mise en forme de cette activité pour être soit imprimée, soit intégrée à un logiciel.

L'approche GEPPETO peut être utilisée pour personnaliser des activités papier ou des EIAH. Nous détaillons à présent chacune de ces deux utilisations de l'approche en indiquant à chaque fois ce qui peut être personnalisé et comment. Nous justifions ces choix en présentant l'étude des activités que nous avons menée et la classification des activités que nous avons définie. À partir de chaque classification, nousinstancions les modèles et processus de l'approche. Nous terminons par une analyse critique de nos propositions.

6.3. GEPPETO_P : ADAPTER DES ACTIVITÉS PAPIER

Dans le cadre de nos recherches, nous nous intéressons à la personnalisation d'activités papier que nous définissons comme un ensemble d'exercices rassemblés dans une feuille à imprimer. Le nombre d'exercices contenus dans une feuille, ainsi que leurs types, vont être différents selon le profil de l'apprenant pour lequel la feuille est créée. Nous nous sommes donc intéressée aux différents exercices pouvant être générés et à la façon de les générer.

6.3.1. PERSONNALISER QUOI ET COMMENT ?

Rappelons que nos recherches concernent la personnalisation des activités quel que soit le domaine enseigné ou le niveau concerné (scolaire ou universitaire). Nous avons donc construit un corpus d'activités papier à partir de diverses ressources issues :

- de l'étude du programme scolaire 2005-2006, du CP à la Terminale, tiré des textes officiels du Ministère de l'Éducation Nationale [B.O. 2005, C.N.D.P. 2009] ;
- de différents exercices proposés par les générateurs d'exercices existants quel que soit le type de ces générateurs (cf. section 1.3.1) ;
- de sites Internet faits par des enseignants proposant des feuilles d'exercices à imprimer ou à faire en ligne, dont les sites [ConnaisTuLAnatomie 2002, Ortholud 2004, Camus 2007, Madoré 2007, Dromer 2009, Momes.Net 2009] ;
- des feuilles d'exercices fournies à une élève de maternelle ;
- des pratiques de deux enseignants de l'enseignement primaire partenaires du projet PERLEA (cf. le paragraphe concernant le partenariat avec les enseignants dans l'introduction du manuscrit) ;
- des manuels scolaires de collège et de lycée dans diverses disciplines (biologie, espagnol...) ;
- des feuilles de Travaux Dirigés fournies aux étudiants de licence Mathématiques et Informatique des universités de Lyon 1 et de Savoie ;
- des livres destinés au travail personnel et aux révisions de l'école maternelle à l'université [Arnaud 1998, Claudal 2007] ;
- de l'étude de travaux de recherche sur la génération des exercices, notamment la génération de QCM [Rus et al. 2009].

Le corpus ne contient naturellement pas toutes les activités existantes, mais il contient des exercices pour des niveaux allant de la maternelle à l'université, et pour des disciplines aussi variées que l'apprentissage de la lecture, l'histoire-géographie, les langues (français, espagnol et anglais), la biologie, les mathématiques, l'informatique, etc.

Ce corpus a été utilisé afin d'identifier des invariants dans la structure des activités. Nous avons ainsi remarqué qu'une activité papier est constituée d'un énoncé et éventuellement de documents annexes (textes et/ou figures). Un énoncé peut contenir des questions ouvertes ou fermées. Une question ouverte nécessite une rédaction de la part de l'apprenant (dissertation, études de document...), alors qu'une question fermée a une liste finie de réponses possibles (QCM, texte à trous...). Il est de plus possible de regrouper les activités papier selon leur type, et pour chaque type d'identifier d'autres invariants. Dans la section suivante, nous présentons la typologie des exercices papier que nous avons obtenue en nous appuyant sur les invariants identifiés pour chaque groupe d'exercices de notre corpus.

Nous nous sommes ensuite intéressée à la façon de générer ces activités papier. L'étude que nous avons faite des générateurs d'exercices existants (cf. section 1.3.1) nous a permis de les classer en trois catégories et d'identifier les forces et les faiblesses de chaque catégorie de générateurs.

Tout d'abord, les **générateurs automatiques** n'offrent aucune interaction avec leur utilisateur et ne permettent donc pas aux enseignants d'adapter les exercices à leurs méthodes de travail. Mais leur point fort est qu'ils permettent de créer rapidement un nombre important d'exercices. Pour cette raison, ce sont les plus souvent intégrés au sein des EIAH. Les **générateurs manuels**, quant à eux, laissent une totale liberté à l'utilisateur, mais, en contrepartie, l'enseignant doit créer l'exercice entièrement la plupart du temps sans aucune aide. À part le

côté interactif, les générateurs manuels n'apportent donc pas réellement d'avantages par rapport à des exercices créés à la main. Enfin, les **générateurs semi-automatiques** sont ceux qui nous semblent les plus intéressants. Ils permettent de créer un ensemble d'exercices diversifiés tout en permettant à un enseignant de les adapter à ses méthodes de travail.

La solution la plus adéquate pour permettre à un enseignant d'agir sur la génération des exercices est donc de lui proposer des générateurs semi-automatiques.

6.3.2. CLASSIFICATION DES ACTIVITÉS SELON LA FAÇON DE LES GÉNÉRER

L'étude de notre corpus d'activités papier nous a permis d'identifier quinze sortes d'activités papier et ainsi de construire une typologie de ces activités (cf. Figure 6-2). Cette typologie a été définie de façon générique et nous l'avons implémentée de façon spécifique dans logiciel Adapte (cf. Chapitre 8). Elle a été obtenue en prenant le point de vue de l'enseignant lors de la création d'une activité papier. Elle se base sur les invariants de structure présents dans les activités du corpus.

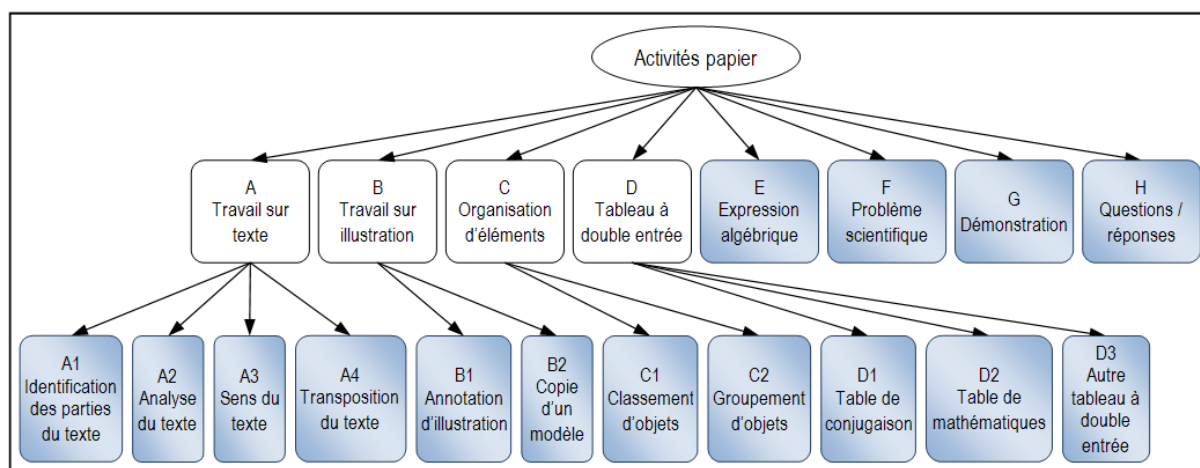


Figure 6-2 : Typologie des activités papier selon les éléments nécessaires à leur génération.

Cette typologie comporte huit catégories d'activités papier (cf. A à H dans la Figure 6-2) auxquelles correspondront huit générateurs. Ces générateurs sont détaillés par la suite (cf. section 6.3.3.d). Les huit catégories sont les suivantes :

- **Travail sur texte** (cf. A sur la Figure 6-2) : activité portant principalement sur un texte comme des textes à trous, des analyses de textes, des traductions ;
- **Travail sur illustration** (cf. B sur la Figure 6-2) : activité centrée sur une illustration comme compléter ou recopier un schéma ;
- **Organisation d'éléments** (cf. C sur la Figure 6-2) : activité demandant de classer des objets (textes, mots, images, nombres, etc.) selon un ou plusieurs critères ;
- **Tableau à double entrée** (cf. D sur la Figure 6-2) : activités portant sur la complétion de tableaux à doubles entrées comme des tables de multiplication, de conjugaison ;
- **Expression algébrique** (cf. E sur la Figure 6-2) : activité portant sur des expressions algébriques comme des résolutions d'équation, des réductions d'expression ;
- **Problème scientifique** (cf. F sur la Figure 6-2) : activité demandant de résoudre un problème portant sur un domaine scientifique (physique, chimie, etc.) ;
- **Démonstration** (cf. G sur la Figure 6-2) : activité demandant de faire une démonstration ;

- **Questions / réponses** (cf. H sur la Figure 6-2) : activité regroupant un ensemble de questions, avec ou sans propositions de réponses.

Certaines de ces catégories peuvent être spécifiées. Au total, notre typologie regroupe quinze types d'activités papier.

Nous présentons maintenant les caractéristiques de chaque catégorie en indiquant leur structure ainsi que les étapes nécessaires pour générer les activités qu'elles contiennent. Nous indiquons également des exemples d'utilisation par les enseignants et ce que les apprenants doivent faire dans une activité. Pour chaque catégorie, des exemples d'activités sont fournis en Annexe D page 291.

A. TRAVAIL SUR TEXTE

Les activités de la catégorie A sur la Figure 6-2 portent sur un ou plusieurs textes. Elles respectent toutes la même structure :

- Un énoncé indiquant le type de l'activité (identifier des mots, analyser le texte, donner un sens, transposer le texte) et les actions à faire par l'apprenant sur le ou les textes (entourer les verbes, compléter les trous, traduire le texte dans telle langue, etc.) ;
- Un ensemble non vide de textes.

Cette catégorie regroupe quatre types d'exercices, respectant la structure ci-dessus et se générant tous avec un même générateur dont les étapes sont :

- Définition de la consigne ;
- Choix des textes ;
- Étiquetage des textes ;
- Sélection des parties du texte ;
- Définition du sens des parties du texte ;
- Mise en page.

La différence entre les quatre types d'activités porte sur les actions que l'on demande de faire à l'apprenant sur les textes et donc sur la nécessité ou non de passer par toutes les étapes que nous venons de citer pour générer une activité.

Les activités de type « Identification des parties du texte » (cf. A1 sur la Figure 6-2) regroupent les textes à trous et les activités demandant à l'apprenant d'identifier des parties du texte.

Les exercices de textes à trous peuvent ou non comporter la liste des éléments permettant de remplir les trous. Dans les deux cas, ces exercices permettent de faire travailler les apprenants en langues sur la lecture, la compréhension de texte, la conjugaison, l'orthographe, le vocabulaire et tous types de travaux lexicaux. Ils permettent, dans de nombreuses matières, de faire travailler les notions de cours, et en mathématiques de faire travailler les apprenants sur les conversions, les expressions algébriques, les équivalences... (cf. Exemples A-1 à A-4 de l'Annexe D, page 291). Pour créer des textes à trous, l'enseignant doit choisir un texte ou le créer. Une fois le texte choisi, l'enseignant détermine les mots à retirer puis décide si l'exercice contiendra ou non la liste des mots à placer afin de faire varier le niveau de difficulté de l'activité.

Les exercices d'identification des parties du texte consistent à demander à l'apprenant de trouver un ou plusieurs mots dans un texte. Cela permet de le faire travailler, par exemple en

langues, sur le vocabulaire ou les synonymes. (cf. Exemples A-5 et A-6 de l'Annexe D, page 293). La création de l'activité ressemble à la précédente pour le choix du texte et des mots à trouver.

Les activités de type « Analyse du texte » (cf. A2 sur la Figure 6-2) regroupent les activités demandant aux apprenants de fournir les caractéristiques des mots ou des parties du texte. Elles sont particulièrement utilisées en langue pour demander de découper les phrases selon leur fonction (sujet, verbe, complément...), selon leur nature (verbe, nom, pronom, adjectif, ...), etc. (cf. Exemples A-7 et A-8 de l'Annexe D, page 293).

Les activités de type « Sens du texte » (cf. A3 sur la Figure 6-2) regroupent les activités de traduction et de définition.

La traduction est destinée aux langues étrangères. Elle consiste à fournir une liste de phrases ou de mots à l'apprenant et à lui demander la traduction dans une langue donnée (cf. Exemples A-9 et A-10 de l'Annexe D, page 294).

Les définitions consistent à demander à l'apprenant de fournir des notions vues ou non en cours. Elles peuvent être utilisées en langues, mais aussi en mathématiques, en biologie... (cf. Exemples A-11 et A-12 de l'Annexe D, page 294).

Les activités de type « Transposition du texte » (cf. A4 sur la Figure 6-2) regroupent les activités demandant à l'apprenant de modifier le ou les textes. Ces modifications peuvent porter sur le temps du texte, le genre des mots, la correction des fautes, etc. (cf. Exemples A-13 à A-16 de l'Annexe D, page 295).

	Identification des parties du texte	Analyse du texte	Sens du texte	Transposition du texte
Définition de la consigne	X	X	X	X
Choix des textes	X	X	X	X
Étiquetage des textes	Optionnelle	X		X
Sélection des parties du texte	X			X
Définition du sens des parties du texte			X	
Mise en page	X	X	X	X

Figure 6-3 : Étapes de génération des activités de la catégorie « Travail sur texte ».

Nous avons vu que la génération des activités de la catégorie « Travail sur texte » se fait en six étapes, et que ce qui différencie les quatre types contenus dans cette catégorie est la nécessité ou non de passer par toutes les étapes pour générer une activité d'un type donné. La Figure 6-3 reprend ces six étapes et indique les étapes nécessaires à la génération de chacun des types A1 à A4.

Ainsi, pour générer des activités de type « Identification des parties du texte », il faut définir la consigne, choisir le ou les textes, étiqueter le texte si besoin pour indiquer par exemple qu'un mot est un nom, un verbe, etc., déterminer les parties du texte à identifier et choisir la présentation de l'exercice comme la suppression ou non des parties à identifier du texte dans le cas des textes à trous.

Pour générer des activités de type « Analyse du texte », il faut définir la consigne, choisir le ou les textes, étiqueter le texte pour indiquer la nature des groupes de mots, des mots, leur fonction, etc. et choisir la présentation de l'exercice comme le fait d'écrire une ligne sur deux pour permettre à l'apprenant d'écrire sous les mots.

Pour générer des activités de type « Sens du texte », il faut définir la consigne, choisir le ou les textes (un texte peut être une phrase ou un mot), définir le sens (définition ou traduction) des textes et choisir la présentation de l'exercice.

Enfin, pour générer des activités de type « Transposition de texte », il faut définir la consigne, choisir le ou les textes (un texte peut être un mot), étiqueter les textes pour indiquer par exemple les mots au singulier, les verbes, etc., définir la transposition des mots, par exemple changement de temps ou de nombre et choisir la présentation de l'exercice, par exemple fournir le texte d'origine ou le texte transformé par l'ajout de fautes.

B. TRAVAIL SUR ILLUSTRATION

Les activités de la catégorie B sur la Figure 6-2 portent toutes sur une ou plusieurs illustrations. Elles ont comme structure générale :

- Un énoncé indiquant les actions à faire par l'apprenant sur la ou les illustrations (annoter une illustration ou recopier un modèle) ;
- Un ensemble non vide d'illustrations.

Cette catégorie regroupe deux types d'exercices respectant cette structure et se générant avec un même générateur dont les étapes sont :

- Définition de la consigne ;
- Choix des illustrations ;
- Modification des illustrations ;
- Mise en page.

La différence entre les deux types d'activités porte sur les actions que l'on demande de faire à l'apprenant sur les illustrations et donc sur les modifications possibles sur les illustrations lors de la génération.

Les activités de type « Annotation d'illustration » (cf. B1 sur la Figure 6-2) regroupent les différentes activités demandant à l'apprenant de compléter une illustration. Ces activités peuvent être utilisées en géographie, en biologie, etc. (cf. Exemples A-17 à A-20 de l'Annexe D, page 295). Dans ce cas, les modifications sur l'image concernent des ajouts de flèches ou de champs à remplir, des suppressions de parties de l'illustration comme des mots.

Les activités de type « Copie d'un modèle » (cf. B2 sur la Figure 6-2) consistent à demander à l'apprenant de recopier une illustration. Ces activités peuvent concerner des lettres de l'alphabet, des mots, mais également des figures (cf. Exemples A-21 et A-22 de l'Annexe D, page 297). Dans ce cas, les modifications sur l'image concernent la police d'écriture des lettres ou des mots à recopier, mais aussi les lignes fournies pour recopier le modèle.

C. ORGANISATION D'ÉLÉMENTS

Les activités de la catégorie C sur la Figure 6-2 portent toutes sur un ensemble d'objets. Un objet peut être un texte, un mot, une lettre, un nombre ou une illustration. Ces activités ont comme structure générale :

- Un énoncé indiquant les actions à faire par l'apprenant sur les objets (classer ou grouper les objets) et selon quels critères ;
- Un ensemble non vide d'objets.

Cette catégorie regroupe deux types d'exercices respectant cette structure et se générant avec un même générateur dont les étapes sont :

- Définition de la consigne ;
- Choix des objets ;
- Définition de l'organisation des objets ;
- Mise en page.

La différence entre les deux types d'activités porte sur les actions que l'on demande de faire à l'apprenant sur les objets et donc sur les critères permettant d'organiser les objets.

Les activités de type « Classement d'objets » (cf. C1 sur la Figure 6-2) demandent à l'apprenant de classer des objets selon des consignes. Elles peuvent être utilisées en mathématiques pour classer les nombres dans un ordre précis, mais également pour de jeunes apprenants à qui on va par exemple demander de classer des mots dans l'ordre alphabétique ou des objets selon leur taille. Elles peuvent également contenir un texte à remettre dans l'ordre. Cela permet de faire travailler un élève en langues sur la compréhension d'un texte, de reconstruire correctement des démonstrations géométriques ou de remettre dans l'ordre des événements, en histoire, en biologie, etc. (cf. Exemples A-23 à A-26 de l'Annexe D, page 298).

Les activités de type « Groupement d'objets » (cf. C2 sur la Figure 6-2) demandent à l'apprenant de grouper des objets selon certaines contraintes. Il peut s'agir de regrouper les fractions qui sont égales, de regrouper des mots qui ont des points communs, au niveau syntaxique ou linguistique, de rassembler des figures ayant des caractéristiques communes (la couleur pour les petits, les propriétés géométriques pour les plus âgés), etc. (cf. Exemples A-27 à A-29 de l'Annexe D, page 298).

D. TABLEAU À DOUBLE ENTRÉE

Les activités de la catégorie D sur la Figure 6-2 portent toutes sur la complétion de tableaux à double entrée. Ces tableaux peuvent être visible ou non pour l'apprenant. Ces activités ont comme structure générale :

- Un énoncé indiquant les actions à faire par l'apprenant (conjuguer un verbe, donner une table de multiplication, remplir le tableau fourni, etc.) ;
- L'intitulé des colonnes du tableau et leur contenu ;
- L'intitulé des lignes du tableau et leur contenu.

Cette catégorie regroupe trois types d'activités respectant cette structure et se générant avec un même générateur dont les étapes sont :

- Définition de la consigne ;
- Définition du contenu du tableau ;

- Définition de l'intitulé et du contenu des colonnes ;
- Définition de l'intitulé et du contenu des lignes ;
- Mise en page.

La différence entre les trois types d'activités porte sur les connaissances que l'on demande de fournir à l'apprenant et donc sur les connaissances nécessaires pour créer et remplir les tableaux.

Les activités de type « Table de conjugaison » (cf. D1 sur la Figure 6-2) peuvent être vues comme un ensemble de tableaux à double entrée. Ensuite, chaque tableau peut porter sur un verbe dans une langue donnée et dans ce cas, l'intitulé des colonnes peut être « le temps » et l'intitulé des lignes « la personne ». Mais un tableau peut aussi porter sur une personne donnée, par exemple la première, et l'intitulé des colonnes sera « le temps » et celle des lignes « le verbe ». On peut donc dire que ce type d'activités correspond à un tableau à double entrée dont le titre, l'intitulé des colonnes et des lignes peuvent être, dans n'importe quel ordre, le temps de conjugaison, le verbe, et la personne. La langue est un paramètre global à l'ensemble des tableaux d'une activité. Ces tableaux peuvent être ou non visibles pour l'apprenant (cf. Exemples A-30 à A-32 de l'Annexe D, page 299). Pour les générer il faut donc définir la consigne, choisir la langue et la listes des verbes, choisir la liste des temps, choisir la liste des personnes et définir la mise en page.

Les activités de type « Table de mathématiques » (cf. D2 sur la Figure 6-2) peuvent être vues comme un ensemble de tableaux à double entrée, de la même manière que le type « Table de conjugaison ». Ainsi ce type d'activités correspond à un tableau à double entrée dont le titre, l'intitulé des colonnes et des lignes peuvent être, dans n'importe quel ordre, l'opération demandée (addition ou multiplication) et les deux membres de l'opération (deux nombres) (cf. Exemple A-33 de l'Annexe D, page 300). Pour les générer, il faut donc définir la consigne, choisir l'opération demandée (addition ou multiplication), choisir la liste des tables, c'est-à-dire les nombres et définir la mise en page.

Les activités de type « Autre tableau à double entrée » (cf. D3 sur la Figure 6-2) sont des tableaux permettant demander à l'apprenant le résultat de la combinaison de deux paramètres (cf. Exemples A-34 à A-38 de l'Annexe D, page 300).

E. EXPRESSION ALGÈBRIQUE

Les activités de type « Expression algébrique » (cf. E sur la Figure 6-2) consistent à fournir à l'apprenant une expression algébrique en lui demandant de faire certaines opérations dessus. Ces exercices permettent de faire travailler les apprenants en algèbre et plus précisément sur les opérations élémentaires, les priorités de calculs, la factorisation et le développement, les identités remarquables, la résolution d'équations et d'inéquations. Ces exercices d'algèbre peuvent être utilisés tout au long de la scolarité (cf. Exemples A-39 à A-42 de l'Annexe D, page 302).

Les activités de ce type ont comme structure générale :

- Un énoncé indiquant le type d'opération à effectuer (calcul, développement, résolution, etc.) ;
- La forme de l'équation ;
- Le domaine de valeur de chacun des paramètres de l'équation.

Pour générer ce type d'activités, il existe un générateur automatique d'exercices d'algèbre dans le micromonde APLUSIX [Bouhineau et al. 2005b]. Il utilise des patrons d'exercices pour connaître

la structure des exercices à fournir et choisit ensuite aléatoirement la valeur des nombres contenus dans ces patrons, en respectant des intervalles associés au niveau demandé.

Les étapes pour générer des activités de ce type sont donc :

- Définition de la consigne ;
- Choix de la forme de l'expression algébrique ;
- Instanciation des paramètres de l'expression algébrique ;
- Mise en page.

F. PROBLÈME SCIENTIFIQUE

Les activités de type « Problème scientifique » (cf. F sur la Figure 6-2) sont communes à toutes les matières dites scientifiques (mathématiques, physique, chimie...). Elles consistent à fournir à l'apprenant un énoncé de problème à résoudre (cf. Exemples A-43 à A-46 de l'Annexe D, page 302). Toute la difficulté de ce type d'exercices est de fournir également une solution. Pour cela, un résolveur de problèmes est nécessaire, comme le résolveur SYRCLAD [Guin 1997].

Les activités de ce type ont comme structure générale :

- Une description de la situation ;
- Une question sur un des éléments de cette situation.

Elles sont créées grâce à un générateur dont les étapes sont :

- Définition du domaine sur lequel porte le problème ;
- Définition des paramètres du problème et des paramètres de complication de l'énoncé (par exemple ajout d'indications inutiles à l'apprenant pour résoudre l'énoncé) ;
- Création de l'énoncé ;
- Mise en page.

G. DÉMONSTRATION

Les activités de type « Démonstration » (cf. G sur la Figure 6-2) sont utilisées en géométrie et en algèbre dans le secondaire et le supérieur. Elles consistent à fournir un énoncé et à demander à l'apprenant de prouver certaines propriétés (cf. Exemple A-47 de l'Annexe D, page 303). Comme pour les problèmes scientifiques, la difficulté pour ce type d'activités est de fournir non seulement un énoncé, mais aussi une solution. Il est donc nécessaire de s'appuyer sur des résolveurs, comme ceux proposés par [Vivet 1984, Pastre et al. 1989, Spagnol 2001].

Les activités de ce type ont comme structure générale :

- Une description de la situation ;
- Une demande de preuve sur une propriété de la situation décrite.

Elles sont créées grâce à un générateur dont les étapes sont :

- Définition du domaine sur lequel porte le problème ;
- Définition des paramètres de la situation et des paramètres de complication de l'énoncé (par exemple ajout d'indications inutiles à l'apprenant pour résoudre l'énoncé) ;
- Création de l'énoncé ;
- Mise en page.

H. QUESTIONS / RÉPONSES

Les activités de type « Questions / réponses » (cf. H sur la Figure 6-2) regroupent plusieurs types d'exercices : les questionnaires à choix multiples (QCM), les questionnaires à réponses ouvertes et courtes (QROC), l'étude de documents, la lecture, la récitation. Ces types d'exercices ont tous la même structure :

- Un énoncé ;
- Un ou plusieurs documents ;
- Des questions, avec ou sans propositions de réponses.

Les QCM demandent à l'apprenant de répondre à une série de questions, où pour chacune, plusieurs réponses sont proposées. Pour les QCM, zéro, une ou plusieurs réponses peuvent être correctes. Les questions et/ou les réponses peuvent être composées de textes, d'illustrations ou des deux. Les activités de ce type peuvent servir à faire travailler un apprenant en langue sur les accords, la conjugaison ; en mathématiques sur le calcul, les conversions ; en histoire sur les dates ; pour les apprenants plus jeunes, les QCM peuvent servir à faire travailler la lecture, la reconnaissance de sons, etc. et plus généralement, dans n'importe quelle matière pour contrôler la connaissance d'un cours ou d'un point précis du cours. Ce type d'exercices est destiné à toutes les disciplines et tous les niveaux (cf. Exemples A-48 à A-50 de l'Annexe D, page 303).

Les QROC sont des questions pour lesquelles aucune proposition de réponse n'est fournie. Les réponses attendues sont courtes. Les questions peuvent être composées de textes et/ou d'illustrations (cf. Exemple A-51 de l'Annexe D, page 304).

L'étude de documents consiste à fournir un texte avec une série de questions ouvertes. Cette activité peut servir en français, en histoire/géographie à partir du collège, etc.

Les étapes pour générer des activités du type « Questions / réponses » sont donc :

- Définition de la consigne ;
- Choix du ou des documents (textes et/ou illustrations) transversaux à l'activité ;
- Définition des questions et, si besoin, pour chacune, définition d'un ensemble de propositions de réponse ;
- Mise en page pour former l'activité souhaitée (QCM, QROC...).

Nous venons de présenter une typologie d'activités papier pouvant être proposées à des apprenants de tous niveaux scolaires ou universitaires et dans toutes disciplines. Certains types d'activités sont spécifiques à des disciplines scientifiques, mais ils restent, pour la majorité, communs à plusieurs niveaux et plusieurs domaines d'enseignement.

Pour générer les exercices de cette typologie, il est nécessaire de disposer de générateurs d'exercices indépendants du domaine. Nous utilisons pour cela les principes de l'approche GEPPE TO.

6.3.3. PRINCIPE DE L'APPROCHE GEPPE TO_P

Dans la section précédente, nous avons présenté la typologie des activités papier sur laquelle nous nous basons pour instancier l'approche GEPPE TO, afin de personnaliser des activités papier. Nous nommons cette déclinaison de l'approche GEPPE TO pour la personnalisation d'activités papier GEPPE TO_P, avec P pour « Paper ». Avant de voir un exemple d'utilisation de

l'approche GEPPETO_p et de détailler les différents modèles et processus définis pour cette approche, nous définissons l'ensemble des termes utilisés.

La typologie d'activités papier proposée contient huit catégories d'activités (cf. A à H sur la Figure 6-2). Chacune des catégories a conduit à la définition d'un patron d'exercices. Certains de ces patrons d'exercices sont décomposables en patrons opérationnels.

Plus formellement, un **patron d'exercices** (par exemple Organisation d'éléments, C sur la Figure 6-2) définit une catégorie d'activités papier se générant toutes avec le même générateur d'exercices. Un **patron opérationnel** (par exemple Classement d'objets, C1 sur la Figure 6-2) spécifie une sous-catégorie d'activités papier se générant avec le générateur d'un patron (ici C), mais avec des types de contraintes de génération particulières et donc une interface de génération adaptée. La structure générique des patrons et l'ensemble des métadonnées communes à tous les patrons sont définies dans une **structure de patrons**.

Créer une **structure d'exercices** consiste à associer à un patron d'exercices des contraintes de génération précises et créer un **exercice** consiste à attribuer des valeurs respectant ces contraintes aux paramètres de la structure d'exercices. Les exercices créés seront constitués d'éléments d'énoncé et d'éléments de réponse pour l'apprenant, ainsi que de la solution pour l'enseignant.

Les modèles en jeu dans GEPPETO_p et leurs relations sont représentés sur la Figure 6-4. Ainsi, notre approche fait appel à une structure de patrons qui décrit huit patrons d'exercices. Certains patrons d'exercices sont composés de plusieurs patrons opérationnels. Chacun des patrons d'exercices peut être utilisé par un générateur d'exercices pour définir des structures d'exercices. Chaque structure d'exercices peut à son tour être utilisée par le générateur pour créer un ensemble d'exercices.

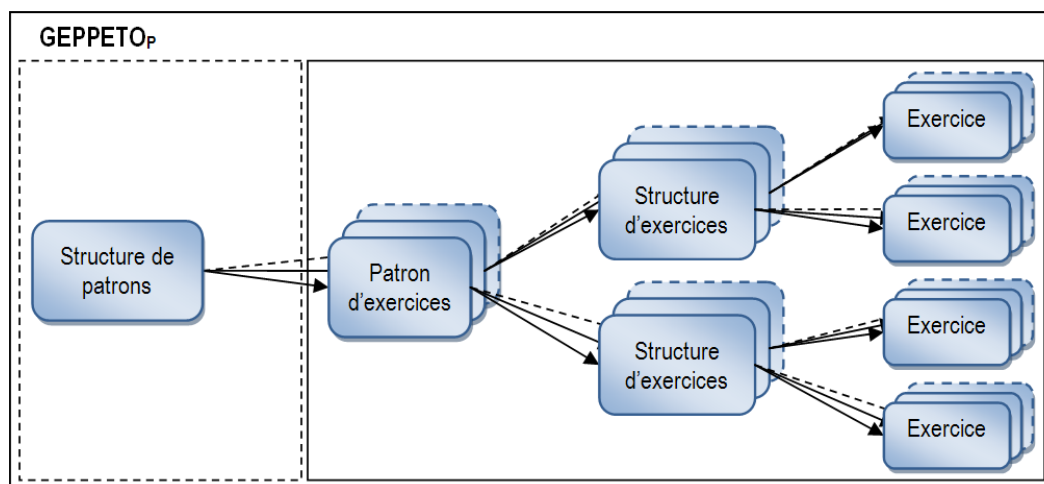


Figure 6-4 : Représentation graphique de l'approche GEPPETO_p.

La structure des patrons et les patrons d'exercices sont indépendants du domaine pour lequel un exercice va être généré. Ce sont des modèles génériques, en quelque sorte des boîtes vides qui donnent un cadre de travail, que les enseignants configurent avec les contenus correspondant à leurs besoins. Ainsi, lorsqu'un enseignant va utiliser un générateur, qui utilise lui-même un patron d'exercices, l'enseignant va créer une structure d'exercices contenant ses propres contraintes. Ces structures d'exercices ne sont donc pas génériques. Elles contiennent des données propres à un enseignant, pour une discipline donnée et pour un niveau d'étude donné. Les exercices créés à partir d'une structure ne seront donc pas non plus génériques et porteront sur la discipline correspondant à la structure d'exercices.

A. EXEMPLE D'UTILISATION

Pour mieux comprendre l'approche GEPPETO_p, donnons un exemple de son utilisation par un enseignant souhaitant personnaliser les activités papier qu'il propose à ses élèves.

Prenons un enseignant qui souhaite créer des exercices pour que ses élèves travaillent en mathématiques sur la relation d'ordre dans l'ensemble des nombres relatifs. Pour cela, il sélectionne le patron « Organisation d'éléments » (cf. C sur la Figure 6-5), puis affine son choix en choisissant le patron opérationnel « Classement d'objets » (cf. C1 sur la Figure 6-5).

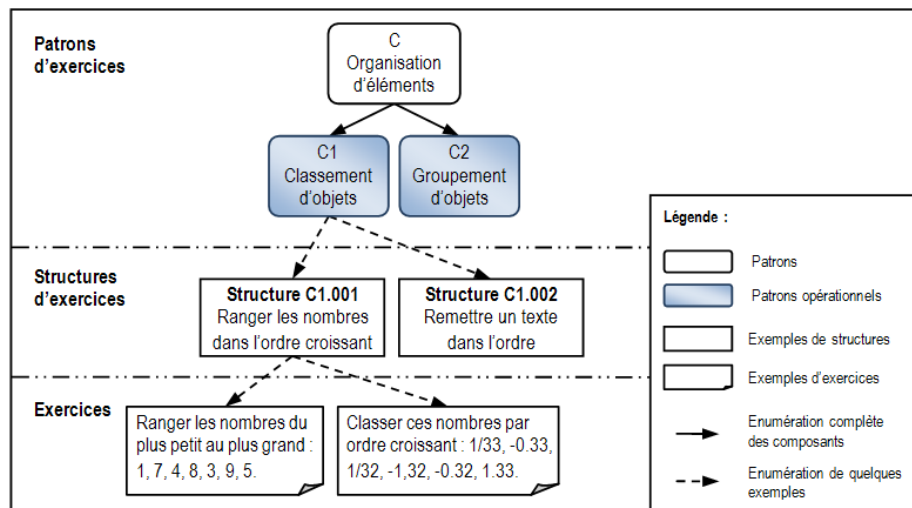


Figure 6-5 : Personnalisation d'activités papier avec l'approche GEPPETO_p.

À partir de ce patron opérationnel, le système mettant en œuvre l'approche GEPPETO_p présente à l'enseignant une interface lui permettant de spécifier les contraintes de génération d'exercices. C'est à ce moment là que l'enseignant formule son souhait de travailler sur des nombres relatifs et spécifie que l'organisation de ces nombres doit se faire par ordre croissant. Il précise également des contraintes sur le choix des nombres qui seront contenus dans l'exercice (par exemple pas de fraction, pas de multiple de 10). Toutes ces contraintes sont mémorisées dans une structure d'exercices (cf. Structure C1.001 sur la Figure 6-5). C'est à partir de cette structure d'exercices que le système génère les exercices. Il peut ainsi générer des exercices différents à partir d'une même structure d'exercices.

B. STRUCTURE DE PATRONS

Le premier modèle GEPPETO_p est la structure de patrons qui est constituée de la définition générale d'un patron et des métadonnées communes à tous les patrons.

La définition générale d'un patron est la suivante : un patron est l'ensemble des éléments discriminants que doit contenir une structure d'exercices pour que l'exercice qu'elle permet de créer appartienne à une classe précise d'exercices. Une structure d'exercices va contenir les différents champs que le générateur d'exercices va devoir remplir pour créer un exercice.

Il faut noter que ce que nous appelons « patron » ne correspond pas au « pattern » de l'approche Design Patterns [Alexander et al. 1997]. Ces patterns respectent toujours un même format de base : un nom, la description d'un problème, la description des solutions à ce problème, des exemples d'utilisation du pattern et des patterns apparentés. Il est possible d'augmenter ce format avec des données propres à chaque domaine [Delozanne et al. 2007]. Dans notre contexte, le problème consiste à créer un exercice d'un certain type en respectant les souhaits

d'un enseignant et la solution consiste à utiliser les processus de GEPPETO_P avec un patron contenant les différents éléments constitutifs de l'exercice. Nous pourrions donc éventuellement comparer nos patrons à une sous-partie des patterns utilisés dans l'approche Design Patterns. Cependant, n'ayant pas utilisé cette approche pour aborder le problème de l'adaptation des activités, nous n'avons pas mis en œuvre les règles de description des patterns et nous ne pouvons donc pas évaluer nos patrons selon les principes des Design Patterns.

Les métadonnées communes aux différents patrons, contenues dans notre structure de patrons, permettent de caractériser les structures d'exercices qui sont créées à partir des patrons. La Figure 6-6 liste ces métadonnées et fournit pour chacune une description. Elle indique également la source de chaque métadonnée puisque certaines métadonnées peuvent être calculées par le système exploitant la structure de patrons alors que d'autres doivent être fournies par le créateur des structures d'exercices. Enfin, la Figure 6-6 indique le statut, obligatoire ou optionnel, de la métadonnée.

	Métadonnée	Description	Source	Obligatoire
1	Identifier	Référence non ambiguë permettant d'identifier chaque structure d'exercices créée à partir d'un patron donné.	Système	X
2	Creator	Nom du créateur de la structure d'exercices.	Système	X
3	Date_Creation	Date de création de la structure d'exercices.	Système	X
4	Date_LastModification	Date de dernière mise à jour de la structure d'exercices.	Système	X
5	Status	Indique si une structure d'exercices créée par un enseignant peut être partagée (<i>public</i>) ou non (<i>private</i>) avec d'autres enseignants.	Créateur	X
6	Name	Nom textuel donné à la structure d'exercices.	Créateur	X
7	LevelStudy	Niveau d'étude (scolaire ou universitaire) auquel sont destinés les exercices issus de la structure d'exercices, par exemple « CP » ou « 6 ^{ème} » pour l'enseignement français.	Créateur	X
8	Duration	Temps, estimé par le créateur, de travail effectué par l'apprenant pour un exercice issu de la structure d'exercices.	Créateur	X
9	Language	Langue dans laquelle les exercices issus de la structure d'exercices seront exprimés.	Créateur	X
10	Discipline	Discipline sur laquelle porte la structure d'exercices, par exemple « Mathématiques ».	Créateur	X
11	Subject	Précise la discipline, par exemple « Addition ».	Créateur	
12	TeachingObjectives	Liste les objectifs pédagogiques visés par les exercices créés à partir de la structure d'exercices.	Créateur	
13	Constraints	Texte expliquant les contraintes de la structure d'exercices.	Créateur	
14	Keywords	Liste de mots-clefs caractérisant la structure d'exercices.	Créateur	
15	Description	Commentaire général sur la structure d'exercices.	Créateur	

Figure 6-6 : Métadonnées de la structure de patrons.

C. PATRONS D'EXERCICES

À partir de la typologie des activités que nous avons proposée (cf. Figure 6-2), basée sur les points communs dans la structure des activités papier, nous avons défini huit patrons d'exercices. Nous présentons ici le premier patron d'exercices, le patron « Travail sur texte » correspondant à la catégorie A de la Figure 6-2, en le décrivant techniquement et en donnant des exemples de structures d'exercices pouvant être créées, et permettant de générer certaines des activités papier présentées en Annexe D page 291. Cette présentation correspond aux modèles des trois derniers niveaux de la Figure 6-4 présentant les modèles GEPPETO_p. Les autres patrons d'exercices, correspondant aux catégories B à H de la Figure 6-2, sont présentés de manière analogue dans l'Annexe F page 307.

Pour présenter nos modèles, et notamment les patrons d'exercices, nous utilisons une représentation graphique des XML Schema contenant leurs définitions. La sémantique de la représentation des XML Schema est fournie en Annexe E page 305, nous ne la redonnons donc pas sur chacune des figures présentant les modèles. De plus, les structures d'exercices montrées en exemple ont été créées avec le logiciel Adapte (cf. Chapitre 8) qui les enregistre au format XML.

PATRON A – TRAVAIL SUR TEXTE

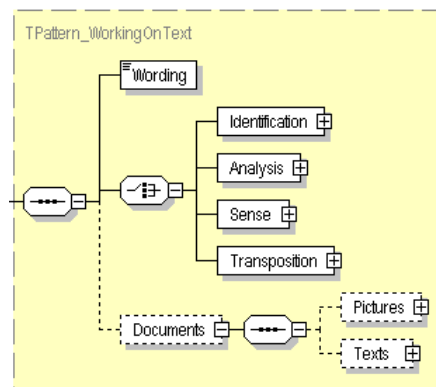


Figure 6-7 : Patron d'exercices « Travail sur texte ».

Le patron « Travail sur texte » permet de créer des activités papier appartenant à la catégorie A de la typologie (cf. Figure 6-2). Ce patron, présenté Figure 6-7, précise qu'une activité de travail sur texte est formée d'une consigne, *Wording*, du contenu d'un des quatre patrons opérationnels rattachés au patron « Travail sur texte », *Identification*, *Analysis*, *Sense* ou *Transposition*, et éventuellement de documents qui sont soit des images, *Pictures* (cf. Figure 6-8), soit des textes *Texts* (cf. Figure 6-9).

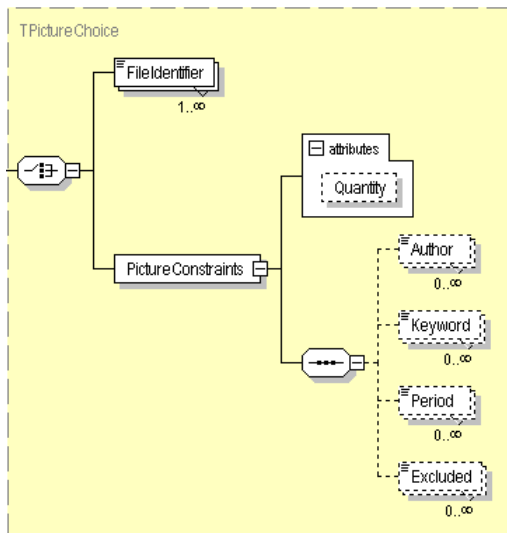


Figure 6-8 : Choix d'une image dans un patron d'exercices.

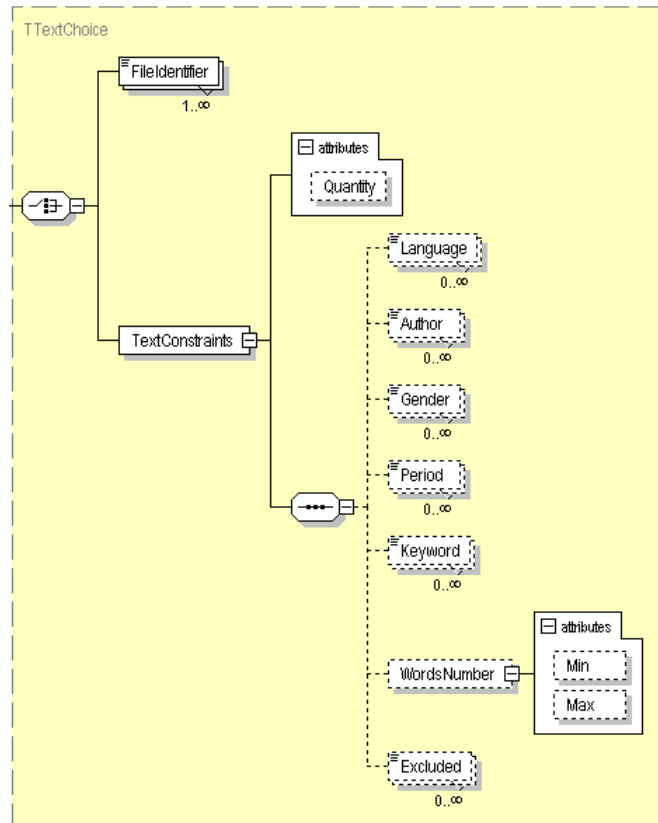


Figure 6-9 : Choix d'un texte dans un patron d'exercices.

Quel que soit le type du document, il peut être choisi expressément lors de la création d'une structure d'exercices ou être choisi lors de la génération des exercices. Si le choix est fixé dans la structure d'exercices, alors le champ *FileIdentifier* contiendra la référence unique du document et tous les exercices générés contiendront ce document. Si le choix est fait lors de la génération de l'exercice, la structure d'exercices contient un ensemble de contraintes permettant d'orienter le système dans le choix du document. Ces contraintes peuvent porter sur l'auteur du document, *Author*, sur la date à laquelle il a été créé, *Period*, sur des mots-clés décrivant le document, *Keyword*, et dans le cas des textes, sur la langue de celui-ci, *Language*, sur son genre, *Gender*, ou sur la longueur du texte, *WordsNumber*. Il est également possible d'exclure certains documents grâce au champ *Excluded*. Le choix du document pourra selon les contraintes varier d'un exercice à l'autre.

Le patron opérationnel « Identifier des parties du texte », présenté Figure 6-10, décrit une activité comme un ensemble non vide d'éléments *Text* associé au nombre *TextsNumber*. Ces éléments s'ajoutent à ceux contenus dans le patron « Travail sur texte ».

Chaque élément *Text* contient les contraintes permettant de créer une activité sur un texte. Le fait de pouvoir avoir plusieurs éléments *Text* permet de créer une structure d'exercices qui permettra de générer des activités portant sur des textes différents. Nous reviendrons sur ce point dans les exemples. Un élément *Text* contient donc des indications permettant de choisir un texte, *TextChoice*, des indications permettant de sélectionner les parties du texte sur lesquelles porteront l'activité, *PartChoice*, et des informations concernant la présentation de l'activité, *Layout*.

L'élément *TextChoice* permet de choisir le texte soit de façon déterminée, soit en fonction de contraintes (cf. Figure 6-9).

L'élément *PartChoice* permet de choisir des parties du texte selon les deux mêmes principes. Dans le premier cas, les parties du texte sont déterminées de manière définitive en fournissant, pour chaque partie, *Part*, sa position dans le texte, *Position*, et sa longueur, *Length*. Dans le second cas, les parties du texte sont choisies lors de la génération des exercices en respectant l'ensemble des contraintes *PartConstraints* : la contrainte *Word* indique si les parties du texte à sélectionner doivent être des mots entiers, des parties de mots ou plusieurs mots successifs ; la contrainte *Number* permet de préciser si les nombres écrits en chiffres arabes peuvent être sélectionnés, doivent être sélectionnés ou ne doivent pas appartenir à la sélection ; les contraintes *Capital* et *Punctuation* permettent le même choix concernant les mots commençant par une majuscule et les signes de ponctuation ; la contrainte *Length* permet de préciser s'il faut inclure ou exclure de la sélection les mots d'une certaine taille ; la contrainte *Subject* permet de sélectionner des mots portant sur un ou plusieurs thèmes ; les contraintes *Nature* et *Function* permettent de sélectionner des mots ou groupes de mots selon leurs natures grammaticales ; la contrainte *Occurrence* permet de sélectionner toutes les occurrences d'une chaîne de caractères donnée ; enfin la contrainte *Excluded* permet d'exclure des parties du texte de la sélection automatique. Il n'est pas nécessaire de préciser toutes les contraintes pour définir une structure d'exercices. De plus, cet ensemble de contraintes a été défini en étudiant les activités papier contenues dans notre corpus. Il peut être augmenté pour permettre de sélectionner des parties du texte selon d'autres critères.

Le nombre *PartsNumber* détermine le nombre de parties à choisir dans le texte. Ce nombre est toujours supérieur à un. Si des parties sont prédéterminées, le nombre *FixedPartsNumber* permet d'indiquer combien de parties sont à choisir dans les éléments *Part*. Ainsi, il est possible de définir un certain nombre de parties, de déterminer des contraintes pour en sélectionner d'autres et ensuite de demander de prendre dans le texte un nombre *PartsNumber* de parties dont le nombre *FixedPartsNumber* dans les parties *Part* et le reste selon les contraintes *PartConstraints*.

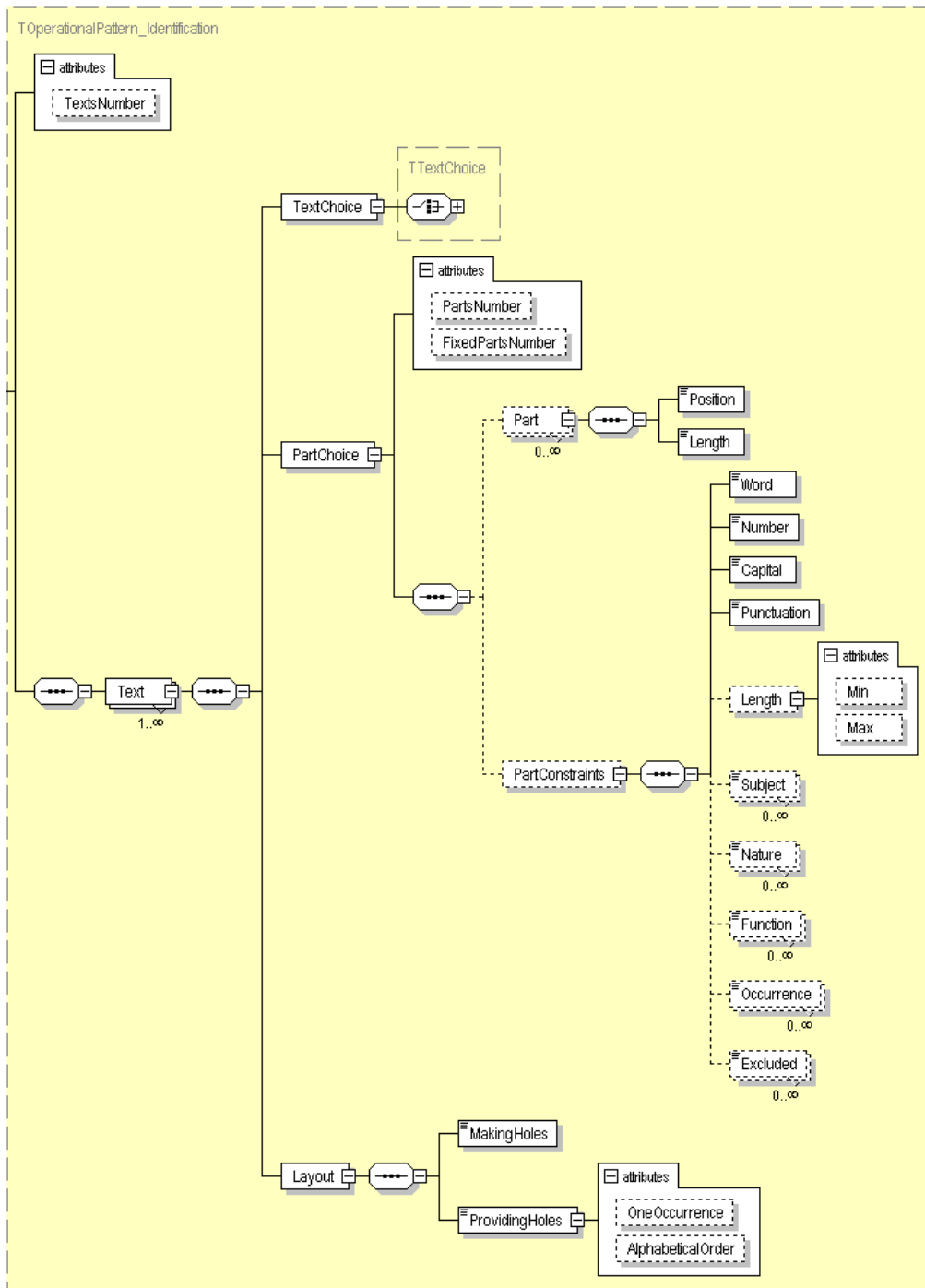


Figure 6-10 : Patron opérationnel « Identification des parties du texte ».

L'élément *Layout* permet d'indiquer, avec le booléen *MakingHoles*, si les éléments sélectionnés dans le texte doivent être supprimés pour former un texte à trous ou être maintenus pour demander à l'apprenant de les retrouver. Dans le cas des textes à trous, il est de plus nécessaire de préciser si les parties supprimées doivent être fournies à l'apprenant, avec le booléen *ProvidingHoles*. Dans l'affirmative, il est possible de spécifier s'il faut fournir la liste des parties triées par ordre alphabétique, *AlphabeticalOrder*, ou ne fournir qu'une occurrence de chaque partie, *OneOccurrence*.

Le nombre *TextsNumber* détermine le nombre de textes qu'un exercice contiendra. Ce nombre est toujours supérieur à un. Si tous les textes sont définis à l'avance (*FileIdentifier* dans l'élément *TextChoice*) et que les parties à sélectionner dans ce texte sont également toutes prédéfinies (les

deux nombres *PartsNumber* et *FixedPartsNumber* sont égaux entre eux et égaux au nombre d'éléments *Part*), le nombre *TextsNumber* est inférieur ou égal au nombre d'éléments *Text* contenus dans la structure d'exercices.

Nous allons à présent voir deux exemples de structures d'exercices créées avec le patron opérationnel « Identification des parties du texte ». Le premier est fourni par la Figure 6-11. Cette structure d'exercices permet de générer l'Exemple A-1 de l'Annexe D page 291. La consigne de l'exercice est fournie dans l'élément *Wording*. L'unique texte est le texte 45 de la base de données, qui correspond au sonnet de Ronsard intitulé « Quand vous serez bien vieille ». Les mots à sélectionner sont précisés dans les éléments *Part*. Le booléen *MakingHole* indique qu'il faut supprimer les mots du texte et le booléen *ProvidingHoles* indique qu'il faut fournir les mots à l'apprenant. Le nombre *PartsNumber* étant égal au nombre d'éléments *Part*, tous les mots indiqués dans les éléments *Part* sont à supprimer. Comme il est spécifié que les mots ne doivent pas être triés par ordre alphabétique, et qu'il faut en fournir la liste complète, ils seront donnés à l'apprenant dans un ordre aléatoire. Les contraintes contenues dans cette structure de profils sont telles que seul l'ordre dans lequel les mots seront présentés à l'apprenant peut varier lors de la génération d'un exercice.

```

<ExercisesStructure>
  <Metadata>...</Metadata>
  - <WorkingOnText>
    <Wording>Des mots de ce texte ont disparu, saurez-vous les remettre en place ?</Wording>
    - <Identification TextsNumber="1">
      - <Text>
        - <TextChoice>
          <FileIdentifier>45</FileIdentifier>
          </TextChoice>
        - <PartChoice PartsNumber="10" FixedPartsNumber="10">
          - <Part>
            <Position>23</Position>
            <Length>7</Length>
            </Part>
          - <Part>
            <Position>45</Position>
            <Length>9</Length>
            </Part>
          - <Part>
            <Position>145</Position>
            <Length>7</Length>
            </Part>
          </PartChoice>
        - <Layout>
          <MakingHoles>true</MakingHoles>
          <ProvidingHoles AlphabeticalOrder="false" OneOccurrence="false">true</ProvidingHoles>
          </Layout>
        </Text>
      </Identification>
    </WorkingOnText>
  </ExercisesStructure>

```

Figure 6-11 : Structure d'exercices issue du patron opérationnel « Identification des parties du texte » associé au patron « Travail sur texte ».

Le second exemple de structure d'exercices, fourni par la Figure 6-12, permet de créer l'Exemple A-2 de l'Annexe D page 291. Cette structure stipule que pour créer un exercice, il faut choisir deux textes dans la base de données. Ces textes doivent être de Jean de la Fontaine et être en langue française. Ensuite il faut supprimer sept mots dans chaque texte. Ces mots sont choisis aléatoirement en excluant les nombres et la ponctuation. Ils ne seront pas donnés à l'apprenant. Cette structure d'exercices, contrairement à la précédente, contient un faible nombre de contraintes, autant pour le choix des textes que pour le choix des parties du texte. Elle permet donc de créer des exercices très diversifiés, autant au niveau des textes fournis que des mots supprimés.


```

<ExercisesStructure>
  <Metadata>...</Metadata>
  - <WorkingOnText>
    <Wording>Remplir les trous.</Wording>
    - <Identification TextsNumber="2">
      - <Text>
        - <TextChoice>
          - <TextConstraints Quantity="1">
            <Language>Français</Language>
            <Author>Jean De La Fontaine</Author>
          </TextConstraints>
        </TextChoice>
      - <PartChoice PartsNumber="7" FixedPartsNumber="0">
        - <PartConstraints>
          <Word>OneWord</Word>
          <Number>Excluded</Number>
          <Capital>Possible</Capital>
          <Punctuation>Excluded</Punctuation>
        </PartConstraints>
      </PartChoice>
    </Text>
    <Layout>
      <MakingHoles>true</MakingHoles>
      <ProvidingHoles AlphabeticalOrder="false" OneOccurrence="false">false</ProvidingHoles>
    </Layout>
  </WorkingOnText>
</ExercisesStructure>

```

Figure 6-12 : Structure d'exercices issue du patron opérationnel « Identification des parties du texte » associé au patron « Travail sur texte ».

Le patron opérationnel « Analyse du texte », présenté Figure 6-13, décrit une activité comme un ensemble non vide d'éléments *Text* associé au nombre *TextsNumber*. Ces éléments s'ajoutent à ceux contenus dans le patron « Travail sur texte ».

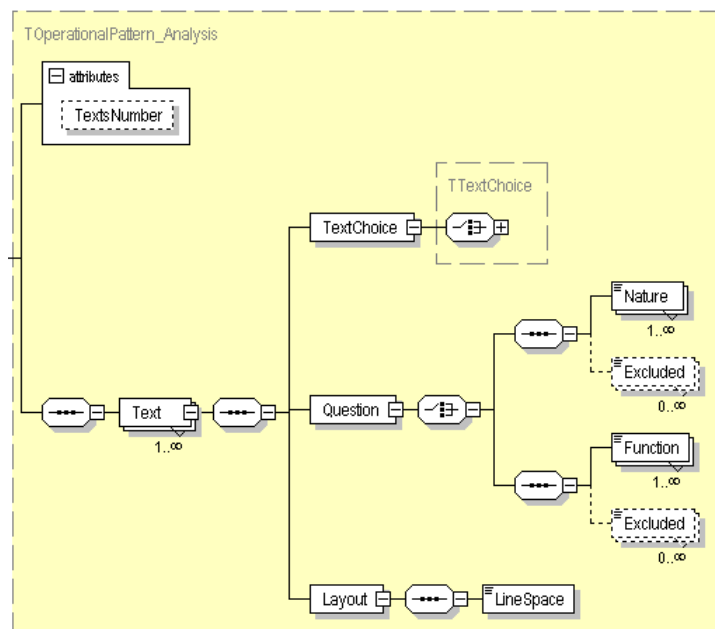


Figure 6-13 : Patron opérationnel « Analyse du texte ».

L'élément *Text* contient des indications permettant de choisir un texte, *TextChoice*, des indications sur les actions à faire par l'apprenant sur ce texte, *Question*, et des informations concernant la présentation de l'activité, *Layout*.

L'élément *TextChoice* est défini comme précédemment (cf. Figure 6-9 et son explication page 134).

L'élément *Question* permet d'indiquer si l'apprenant doit analyser le texte pour déterminer la nature, *Nature*, ou la fonction, *Function*, des parties du texte. Il est possible d'exclure certaines natures ou fonctions grâce au champ *Excluded*.

L'élément *LineSpace* contenu dans l'élément *Layout* permet d'indiquer la valeur de l'interligne à utiliser pour présenter le texte à l'apprenant, afin de lui laisser ou non la place d'annoter le texte.

Le nombre *TextsNumber* détermine le nombre de textes qu'un exercice contiendra. Ce nombre est toujours supérieur à un. Si tous les textes sont définis à l'avance (*FileIdentifier* dans l'élément *TextChoice*), le nombre *TextsNumber* est inférieur ou égal au nombre d'éléments *Text* contenus dans la structure d'exercices.

Nous allons à présent voir un exemple de structure d'exercices créée avec le patron opérationnel « Analyse du texte ». Cet exemple, donnée en Figure 6-14, permet de générer l'Exemple A-8 de l'Annexe D page 293. La consigne de l'exercice est fournie dans l'élément *Wording*. L'unique texte est le texte 34 de la base de données. L'apprenant devra analyser le texte pour identifier les compléments d'objet direct (COD) et les compléments d'objet indirect (COI). Le texte sera présenté sans intercaler de lignes puisque la consigne indique de souligner et non d'annoter le texte.

```

<ExercisesStructure>
  <Metadata>...</Metadata>
  - <WorkingOnText>
    <Wording>Souligne les COD d'un trait et les COI de deux traits.</Wording>
    - <Analysis TextsNumber="1">
      - <Text>
        - <TextChoice>
          <FileIdentifier>34</FileIdentifier>
        </TextChoice>
      - <Question>
        <Function>COD</Function>
        <Function>COI</Function>
      </Question>
    - <Layout>
      <LineSpace>simple</LineSpace>
    </Layout>
  </Text>
</Analysis>
</WorkingOnText>
</ExercisesStructure>

```

Figure 6-14 : Structure d'exercices issue du patron opérationnel « Analyse du texte » associé au patron « Travail sur texte ».

Le patron opérationnel « Sens du texte », présenté Figure 6-15, décrit une activité comme la combinaison de l'élément *Language* et soit d'un ensemble d'éléments *ElementIdentifier*, soit d'un élément *ElementChoice*. Cette combinaison est associée au nombre *SensesNumber*. Ces éléments s'ajoutent à ceux contenus dans le patron « Travail sur texte ».

L'élément *Language* permet d'indiquer la langue dans laquelle sont fournis les éléments pour lesquels un sens sera demandé, ainsi que la langue dans laquelle ce sens devra être fourni. Pour des exercices de traduction, ces deux langues seront différentes. Pour des exercices de définition, ces deux langues pourront être identiques ou différentes.

L'élément *ElementIdentifier* permet de choisir un mot ou un texte précis ainsi que les sens (définition ou traduction) corrects pour ce mot ou ce texte. En effet, un mot peut avoir plusieurs définitions, et pour une discipline donnée, seules certaines de ces définitions peuvent avoir du sens. De même, un mot ou un texte peuvent être traduits de plusieurs manières. Ainsi, les éléments *SenseWordId* et *SensTextId* permettent de sélectionner les réponses qui sont correctes pour le créateur de la structure d'exercices.

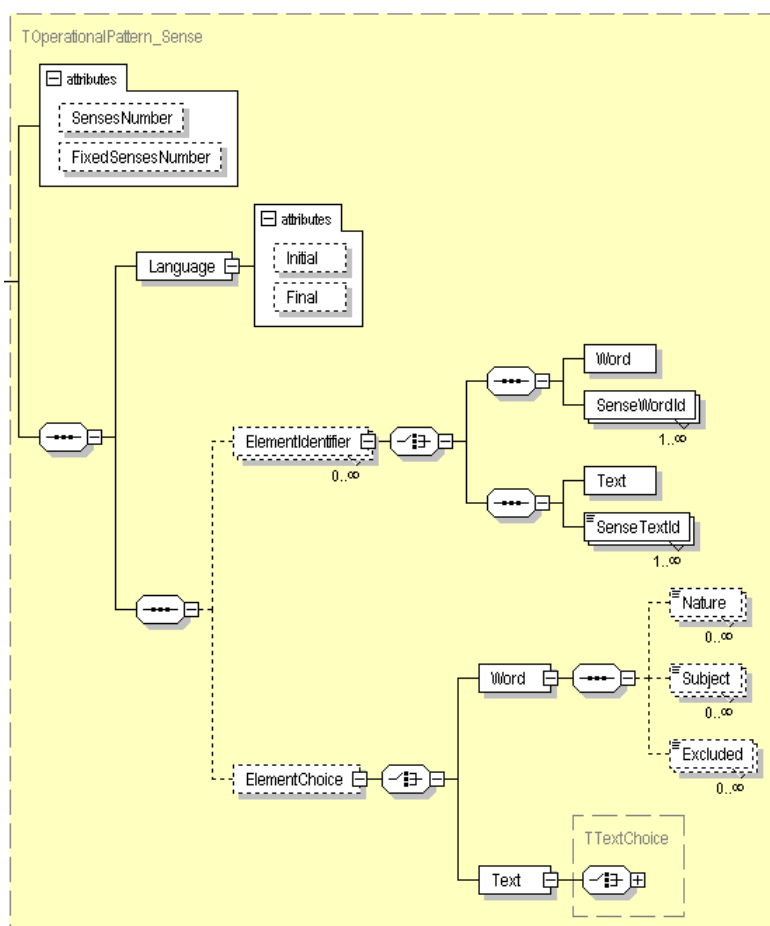


Figure 6-15 : Patron opérationnel « Sens du texte ».

L'élément *ElementChoice* contient des contraintes permettant de choisir soit des mots, *Word*, soit des textes, *Text*, selon un ensemble de contraintes. Les contraintes concernant le choix des mots peuvent porter sur leur thème, *Subject*, ou leur nature, *Nature*, par exemple des adjectifs ou des verbes. La contrainte *Excluded* permet d'exclure des mots de la sélection automatique. Les contraintes concernant le choix des textes ont été présentées précédemment (cf. Figure 6-9 et son explication page 134).

Le nombre *SensesNumber* détermine le nombre d'éléments, mots ou textes, qu'un exercice contiendra. Ce nombre est toujours supérieur à un. Si des éléments sont prédéterminés, le nombre *FixedSensesNumber* permet d'indiquer combien d'éléments sont à choisir dans les éléments *ElementIdentifier*. Ainsi, il est possible de définir un certain nombre d'éléments, de déterminer des contraintes pour en sélectionner d'autres et ensuite de demander de prendre un nombre *SensesNumber* d'éléments dont le nombre *FixedSensesNumber* dans les éléments *ElementIdentifier* et le reste selon les contraintes *ElementChoice*.

Nous allons à présent voir deux exemples de structures d'exercices créées avec le patron opérationnel « Sens du texte ». Le premier, fourni sur la Figure 6-16, est une structure d'exercices permettant de générer des exercices de traduction de l'anglais vers le français, comme indiqué dans l'élément *Language*. La consigne de l'exercice est fournie dans l'élément *Wording*. Le nombre *SensesNumber* indique qu'il y a un élément à traduire. Cet élément est un texte qui sera choisi dans la base de données, comme l'indique l'élément *ElementChoice*, en respectant les contraintes données dans *TextConstraints*. Ainsi le texte devra être en anglais, l'auteur devra être William Shakespeare et le texte devra avoir moins de 100 mots.

```

<ExercisesStructure>
  <Metadata>...</Metadata>
  - <WorkingOnText>
    <Wording>Donner la traduction du texte de William Shakespeare.</Wording>
  - <Sense SensesNumber="1" FixedSensesNumber="0">
    <Language Initial="Anglais" Final="Français" />
  - <ElementChoice>
    - <Text>
      - <TextConstraints Quantity="1">
        <Language>Anglais</Language>
        <Author>William Shakespeare</Author>
        <WordNumber Max="100" />
      </TextConstraints>
    </Text>
  </ElementChoice>
</Sense>
</WorkingOnText>
</ExercisesStructure>

```

Figure 6-16 : Structure d'exercices issue du patron opérationnel « Sens du texte » associé au patron « Travail sur texte ».

Cette structure d'exercices permet de générer des exercices de traduction portant sur des textes différents comme le montre les Exemples A-9 et A-10 de l'Annexe D page 294.

Un second exemple est fourni sur la Figure 6-17. Cette structure d'exercices permet de générer des exercices de définition puisque, dans l'élément *Language*, la langue de départ et celle d'arrivée sont les mêmes. La consigne de l'exercice est fournie dans l'élément *Wording*. Le nombre *SensesNumber* indique qu'il y a quatre éléments à définir. Ces éléments sont des mots, *Word*, choisis aléatoirement, *ElementChoice*, en respectant les contraintes portant sur le thème des mots, *Subject*. Ainsi les mots doivent porter sur la biologie, et plus précisément être en rapport avec le corps humain et ses cellules.

```

<ExercisesStructure>
  <Metadata>...</Metadata>
  - <WorkingOnText>
    <Wording>Donner la définition des termes suivants :</Wording>
  - <Sense SensesNumber="4" FixedSensesNumber="0">
    <Language Initial="Français" Final="Français" />
  - <ElementChoice>
    - <Word>
      <Subject>Biologie</Subject>
      <Subject>Corps humain</Subject>
      <Subject>Cellule</Subject>
    </Word>
  </ElementChoice>
</Sense>
</WorkingOnText>
</ExercisesStructure>

```

Figure 6-17 : Structure d'exercices issue du patron opérationnel « Sens du texte » associé au patron « Travail sur texte ».

Cette structure d'exercices, comme la précédente, permet de créer des exercices très diversifiés en ce qui concerne la liste des mots à définir, comme le montrent les Exemples A-11 et A-12 de l'Annexe D page 294.

Le patron opérationnel « Transposition du texte », présenté Figure 6-18, décrit une activité comme un ensemble non vide d'éléments *Text* associé au nombre *TextsNumber*. Ces éléments s'ajoutent à ceux contenus dans le patron « Travail sur texte ».

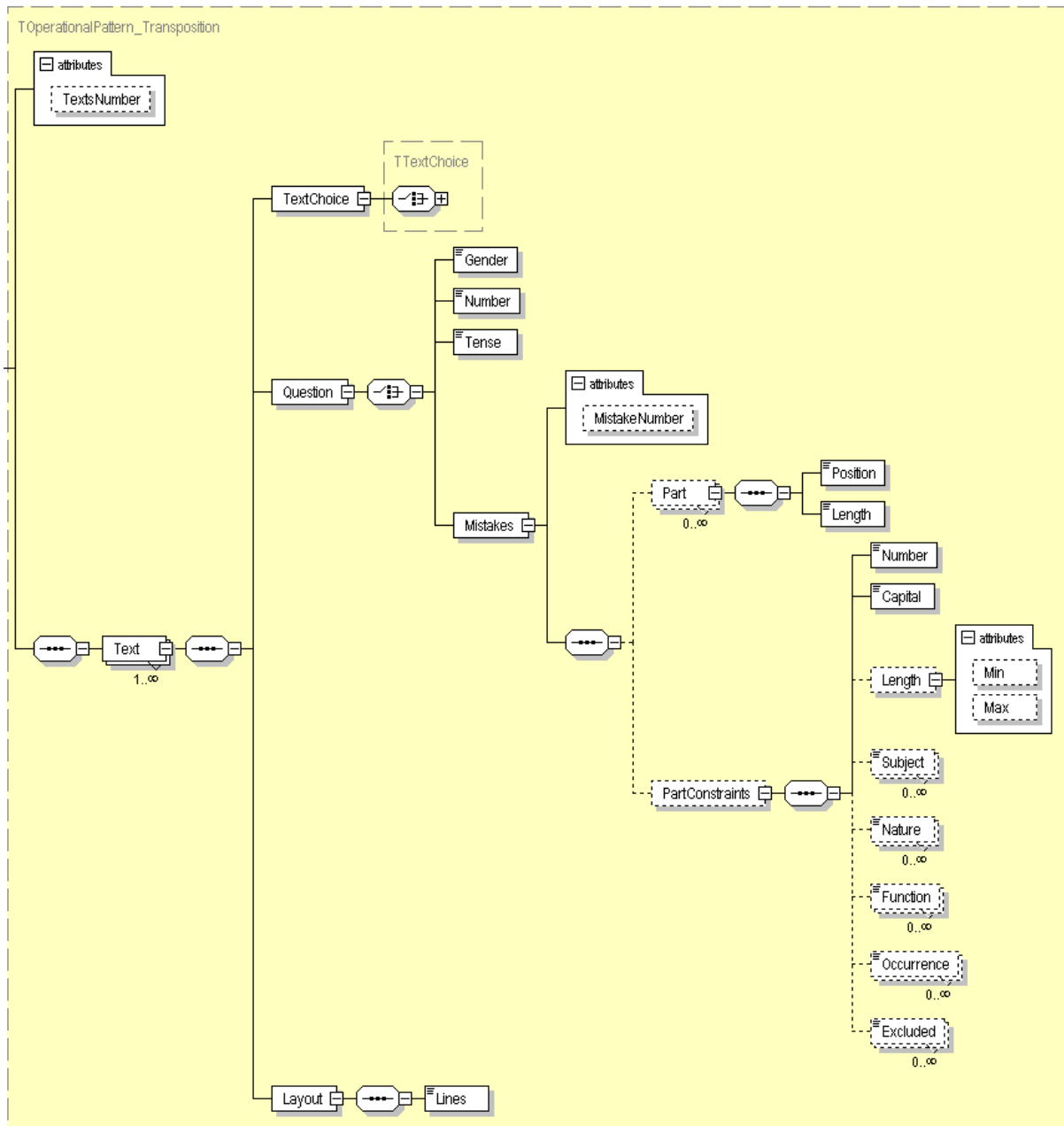


Figure 6-18 : Patron opérationnel « Transposition du texte ».

L'élément *Text* contient des indications permettant de choisir un texte, *TextChoice*, des indications sur les actions à faire par l'apprenant sur ce texte, *Question*, et des informations concernant la présentation de l'activité, *Layout*.

Les éléments *TextsNumber* et *TextChoice* sont définis comme précédemment (cf. respectivement page 134 et 132).

L'élément *Question*, permet d'indiquer le changement que l'apprenant doit faire sur le texte. Ce changement peut porter sur le genre des mots, *Gender*, par exemple changer tout pronom masculin sujet (il, ils) en pronom féminin (elle, elles) ; sur le nombre des mots, *Number*, par exemple changer tous les singuliers en pluriels ; sur le temps du texte, *Tense*, par exemple passer un texte du présent au futur ; et enfin sur la correction de fautes dissimulées dans le texte, *Mistakes*. Ces fautes peuvent être soit définies manuellement en changeant des parties du texte, *Part*, soit créés automatiquement selon des contraintes, *PartConstraints*. Ces contraintes peuvent

être de changer certains nombres, *Number*, les mots commençant par une majuscule, *Capital*, les mots faisant une certaine taille, *Length*, portant sur un thème donné, *Subject*, ayant une nature donnée, *Nature*, par exemple les adjectifs, une fonction précise donnée, *Function*, ou de changer certaines occurrences d'un mot, *Occurrence*, par exemple inverser certaines occurrences de « ce » par « se » et inversement. La contrainte *Excluded* permet de préciser des mots à ne pas modifier.

L'élément *Layout* permet d'indiquer s'il faut fournir les lignes pour que les apprenants aient la place de recopier le texte.

Nous allons à présent voir un exemple de structure d'exercices créée avec le patron opérationnel « Transposition du texte ». La structure d'exercices, fournie Figure 6-19, permet de générer des exercices demandant à l'apprenant de mettre un texte à l'imparfait, comme l'indiquent les éléments *Question* et *Tense*. La consigne de l'exercice est fournie dans l'élément *Wording*. Le nombre *TextsNumber* indique qu'il n'y a qu'un seul texte à transformer. Ce texte est choisi en respectant les consignes, *TextConstraints*, indiquant que le temps doit être écrit en français, au présent et doit comporter entre 30 et 40 mots.

```

<ExercisesStructure>
  <Metadata>...</Metadata>
  - <WorkingOnText>
    <Wording>Recopie ce texte à l'imparfait de l'indicatif.</Wording>
  - <Transposition TextsNumber="1">
    - <Text>
      - <TextChoice>
        - <TextConstraints Quantity="1">
          <Language>Français</Language>
          <Keyword>Présent</Keyword>
          <WordNumber Min="30" Max="40" />
        </TextConstraints>
      </TextChoice>
    - <Question>
      <Tense>Imparfait de l'indicatif</Tense>
    </Question>
  - <Layout>
    <Lines>false</Lines>
  </Layout>
</Text>
</Transposition>
</WorkingOnText>
</ExercisesStructure>

```

Figure 6-19 : Structure d'exercices issue du patron opérationnel « Transposition du texte » associé au patron « Travail sur texte ».

Cette structure d'exercices permet de générer des exercices portant sur des textes différents, comme le montre les Exemples A-15 et A-16 de l'Annexe D page 295.

Toutes les contraintes de sélection d'éléments (texte, image, mot, lettre, nombre), de groupement d'objets (comme dans le patron opérationnel « Classement d'objets »), etc., contenues dans les patrons d'exercices ont été identifiées suite à l'étude des activités papier de notre corpus. Elles ne couvrent probablement pas toutes les activités papier pouvant exister et peuvent donc être complétées.

D. GÉNÉRATEURS SEMI-AUTOMATIQUES

Nous avons présenté précédemment la typologie des activités papier que nous nous proposons de personnaliser (cf. section 6.3.2), ainsi que les différents patrons d'exercices formalisant la structure des activités contenues de notre typologie (cf. section 6.3.3.c.). Pour générer ces types d'activités papier, proposés dans plusieurs disciplines et pour plusieurs niveaux d'étude, il est nécessaire de disposer de générateurs d'exercices indépendants du domaine. L'étude que nous

avons faite des générateurs d'exercices existants (cf. section 1.3.1) nous a permis de les classer en trois catégories : les générateurs automatiques, les générateurs semi-automatiques et les générateurs manuels. Afin de déterminer le type de générateurs le plus adéquat dans le cadre de nos travaux de recherche, nous avons rappelé, dans la section 6.3.1, leurs forces et leurs faiblesses.

Dans l'approche GEPPETO_p, le plus intéressant semble d'intégrer des générateurs semi-automatiques puisqu'ils permettent de créer un grand nombre d'exercices en autorisant la personnalisation de leur génération.

La question est de savoir si cela est techniquement faisable mais aussi acceptable pour tous les patrons d'exercices que nous avons identifiés (cf. Figure 6-2). En effet, même si une génération semi-automatique est possible, il faut que le résultat soit utilisable par les utilisateurs. Il faut pour cela pouvoir créer des exercices de bonne qualité et répondant aux besoins pédagogiques précis des utilisateurs. Dans le cas des QCM par exemple, la génération automatique de questions portant sur un texte est bien établie, et a été appliquée à des domaines divers comme la compréhension de la lecture dans l'apprentissage des langues [Mostow et al. 2004]. Il semble tout à fait possible de contraindre cette génération pour répondre à des critères spécifiés par l'utilisateur. Mais la génération automatique de questions ne portant pas sur des documents mais sur des thèmes ou des connaissances précises est plus compliquée à mettre en œuvre. Il faudrait en effet des bases de connaissances de très grande taille pour couvrir toutes les disciplines et tous les niveaux d'études. Une solution intermédiaire consiste à proposer à l'enseignant un générateur manuel permettant de fournir la base de connaissances pour un exercice (une série de questions avec, pour chacune, plusieurs propositions de réponse). Le système choisit ensuite un certain nombre de questions et leurs réponses, en remplaçant certains termes par des termes équivalents pour faire varier les énoncés. Cette solution permet, pour une même structure d'exercices, de générer des QCM différents : ils n'auront que certaines questions en commun, et pour celles-ci, ils n'auront pas forcément les mêmes propositions de réponses. Cette démarche, consistant à demander à l'enseignant de fournir les bases de connaissances des générateurs semi-automatiques, sera utilisée dans les cas où les générateurs semi-automatiques classiques ne peuvent pas être mis en place dans notre contexte générique.

Notre choix se portant sur des générateurs semi-automatiques, nous avons ensuite étudié la possibilité, pour chaque patron d'exercices, de se servir de générateurs existants.

Pour les exercices de types « Travail sur texte », « Travail sur illustration », « Organisation d'éléments », « Tables » et « Questions-Réponses » (cf. A, B, C, D et H de la Figure 6-2), les générateurs que nous avons pu étudier nécessitent une saisie complète des exercices par l'enseignant ou génèrent des exercices sans laisser les enseignants intervenir dans le processus de création. En les utilisant, nous n'aurions donc pas pu proposer aux enseignants une part d'aléatoire dans la génération de leurs exercices. Nous ne les utilisons donc pas dans l'approche GEPPETO_p.

Concernant les exercices de type « Expression algébrique » (cf. E sur la Figure 6-2), l'environnement APLUSIX possède un générateur d'exercices [Bouhineau et al. 2005b]. Ce générateur s'appuie sur une hiérarchie de patrons d'exercices qui définit l'ensemble des 436 types d'exercices qu'il peut générer. Un type d'exercice est caractérisé par le problème algébrique posé (calculer, développer, factoriser, résoudre) et par la forme de l'expression algébrique. Ce générateur fait donc ce que nous souhaitons, il est suffisamment large pour fournir une grande variété d'exercices portant sur divers problèmes et divers types d'expressions algébriques. Le problème de ce générateur est qu'il est automatique : les enseignants ne peuvent donc pas intervenir dans la création d'un exercice. Nous ne pouvons donc pas l'utiliser directement, mais nous pouvons nous en inspirer, en conservant la hiérarchie

des patrons d'exercices et en utilisant l'interface proposée à l'apprenant (la carte de tests de la Figure 1-5, page 39) pour qu'il choisisse une famille d'exercices. Les principes de cette interface peuvent être intégrés au générateur pour permettre à l'enseignant de contraindre, selon ses propres buts pédagogiques, la génération des exercices.

Pour les exercices de type « Problème scientifique » (cf. F sur la Figure 6-2), nous avons intégré GENAMBRE, le générateur d'AMBRE-ENSEIGNANT [Duclosson et al. 2005b] implémenté pour créer des problèmes additifs dans l'EIAH AMBRE-ADD [Nogry et al. 2004]. En fournissant les bases de connaissances nécessaires, ce générateur peut être utilisé de manière générique et ainsi fournir des exercices portant non seulement sur des problèmes additifs mais aussi sur des problèmes de dénombrement, de thermodynamique, etc.

Concernant les exercices de type « Démonstration » (cf. G sur la Figure 6-2), il est nécessaire d'intégrer un ou plusieurs démonstrateurs pour pouvoir fournir une solution à l'apprenant. De nombreux travaux visant à proposer des démonstrateurs ont été menés [Vivet 1984, Pastre et al. 1989, Py 2001, Spagnol 2001]. À l'heure actuelle, nous n'avons pas suffisamment exploré cette piste pour permettre leurs mises en œuvre dans notre approche.

Certains générateurs semi-automatiques permettent de créer des exercices correspondant à plusieurs patrons, comme PÉPIGEN [Prévit et al. 2007], mais ce sont des générateurs spécifiques à une discipline et souvent à un niveau d'étude donné. Nous ne les utilisons donc pas non plus dans l'approche GEPPETOP.

Ainsi, à chacun des huit patrons d'exercices présentés dans la section 6.3.3.c et dans l'Annexe F va correspondre un générateur d'exercices qui crée l'exercice pour l'apprenant et la correction pour l'enseignant. Cette correction sera soit définie par le générateur lorsque que cela est possible, soit saisie par l'enseignant. De même, si certaines contraintes ne sont pas spécifiées par l'enseignant, elles seront définies aléatoirement par le système. De plus, lors de la génération d'un exercice, la structure d'exercices pourra contenir des contraintes de re-génération permettant de ne pas fournir plusieurs fois le même exercice pour la même structure. Tous les générateurs proposés dans le cadre de l'approche GEPPETOP respectent une architecture générique (cf. Figure 6-20) que nous détaillons avant de présenter sa déclinaison pour chacun des huit générateurs.

L'architecture générique des générateurs d'exercices permet de distinguer quatre niveaux (cf. Figure 6-20) : le **niveau général** contient les connaissances indépendantes du domaine pour lequel on souhaite générer un exercice, par exemple les connaissances permettant d'écrire un énoncé en langue naturelle. Le **niveau domaine** contient les connaissances propres au domaine d'application, par exemple des connaissances de calcul. Le **niveau génération** contient les processus propres à la création d'un exercice : la définition de contraintes sur un patron d'exercices mémorisées dans une structure d'exercices, l'instanciation de cette structure pour générer un exercice et sa solution puis la mise en page permettant de fournir des exercices ayant une présentation homogène. Enfin, le **niveau exercice** contient les ressources propres à l'exercice créé parmi lesquelles la structure d'exercices et son instanciation qui contient un énoncé d'exercice et sa correction.

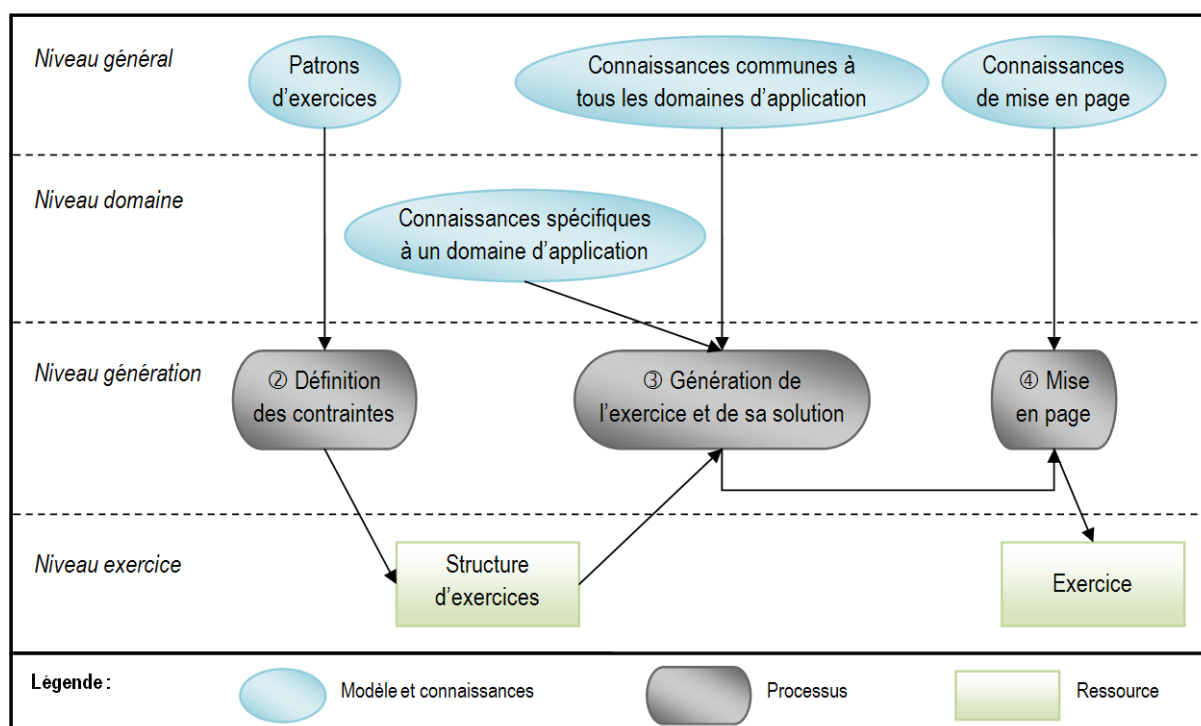


Figure 6-20 : Architecture générique des générateurs d'exercices de l'approche GEPPETO.

Le processus de génération comporte trois étapes. Nous nommons ici ces étapes pour pouvoir faire le lien entre l'architecture générique des générateurs d'exercices, sa déclinaison pour chacun des huit générateurs nécessaires à la génération des exercices de notre typologie et les processus de l'approche GEPPETO. L'étape 1, consistant à proposer une interface adaptée pour définir des contraintes, n'est pas présente dans l'architecture des générateurs, nous en verrons la raison dans la section 6.3.4 page 148. Les étapes de génération sont les suivantes :

- Étape 2 : la définition des contraintes par l'utilisateur du générateur ;
- Étape 3 : la génération de l'exercice et de sa solution par le système ;
- Étape 4 : la mise en page de l'exercice par le système.

Pour illustrer la généralité de cette architecture, nous l'avons appliquée aux générateurs d'exercices nécessaires pour générer les exercices de notre typologie d'activités papier. Certaines bases de connaissances contenues dans les générateurs sont communes à plusieurs générateurs. C'est le cas par exemple des bases de textes, d'images, des connaissances de mise en page mais aussi des connaissances sur les mots, les lettres ou la conjugaison. Nous expliquons dans la suite de cette section comment nous avons instancié l'architecture générique pour le générateur « Travail sur texte », les autres générateurs étant présentés en Annexe G page 333. Dans la description de ces générateurs, nous parlons d'utilisateur pour signifier que ces générateurs peuvent être utilisés par un enseignant mais également par un autre système.

GÉNÉRATEUR « TRAVAIL SUR TEXTE »

Ce générateur doit permettre de fournir des exercices d'identification de parties du texte (texte à trous, pendu, identification des fautes, etc.), des exercices d'analyse de texte (analyse grammaticale d'un texte), des exercices de définition ou de traduction et enfin des exercices de transpositions de texte (passer un texte au futur, changer le genre des pronoms sujet, etc.). Ces exercices portent sur un texte ou une liste de mots. Le texte et les mots sont choisis semi-automatiquement en respectant les contraintes définies par l'utilisateur du générateur. Les actions à faire sur le texte sont diverses : suppression de mots, changement de temps,

traduction... Nous présentons ici les bases de connaissances nécessaires à ce générateur, ainsi que les différentes étapes du processus de génération.

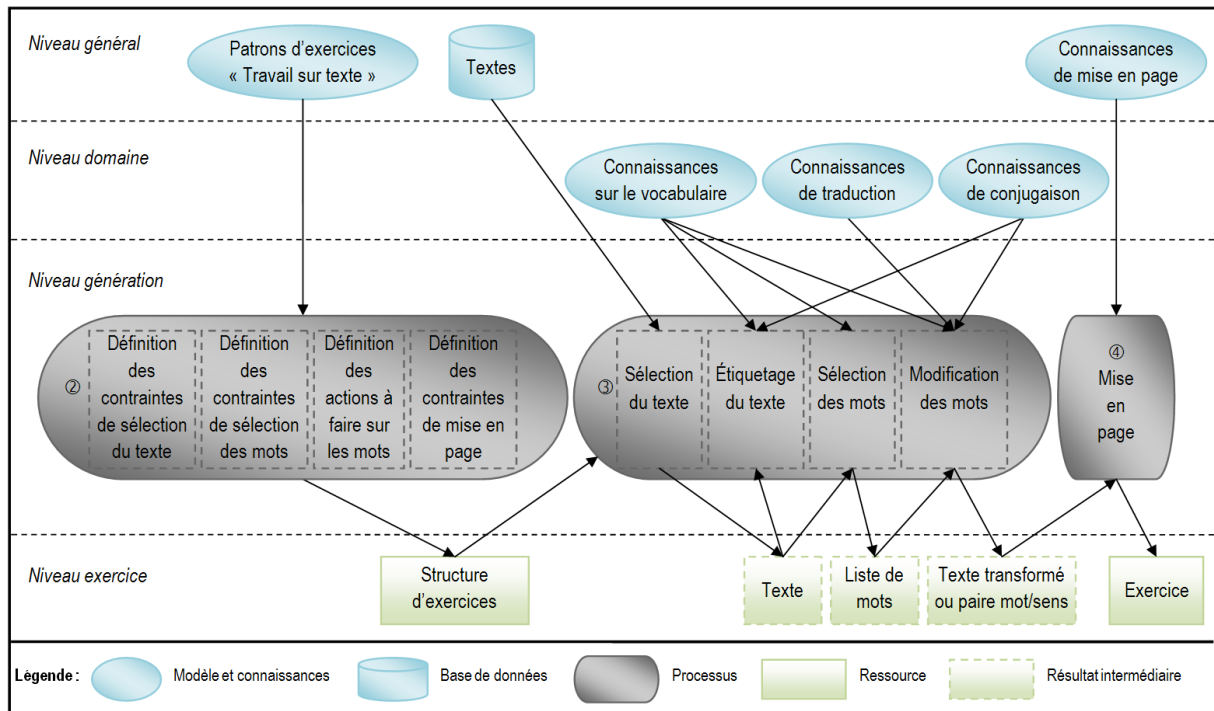


Figure 6-21 : Architecture du générateur « Travail sur texte ».

NIVEAU GÉNÉRAL : LES BASES DE CONNAISSANCES INDÉPENDANTES DU DOMAINE

Patron d'exercices. Il s'agit du patron d'exercices « Travail sur texte » présenté dans la section 6.3.3.c page 127.

Textes. La base de données des textes contient tous les textes traités par le système : les textes d'origine, ainsi que ceux importés et saisis par les utilisateurs. Tous les mots des textes sont définis (genre, définition, pluriel...) dans la base de connaissances sur le vocabulaire. Comme il est possible de le voir sur l'extrait présenté Figure 6-22 de la base de textes du logiciel Adapte mettant en œuvre ce générateur (cf. Chapitre 8), chaque texte est étiqueté à l'aide des métadonnées suivantes : son identifiant, son auteur, son titre, l'année de sa parution, son genre (conte, récit, poésie, dialogue, etc.), la langue dans laquelle il est écrit et le nombre de mots qu'il contient. Ces métadonnées permettent de choisir un texte en fonction de contraintes. La référence unique et le nombre de mots du texte sont calculés automatiquement, les autres métadonnées, optionnelles, sont fournies par l'utilisateur.

Connaissances de mise en page. Ces connaissances permettent de présenter un exercice en fournissant l'énoncé à l'apprenant et la correction à l'utilisateur.

NIVEAU DOMAINE : LES BASES DE CONNAISSANCES DÉPENDANTES DU DOMAINE

Connaissances sur le vocabulaire. Ces connaissances correspondent à une base de mots, où chaque mot est étiqueté par la langue dont il est issu (français, anglais...), le nombre de lettres qu'il contient, son genre, son nombre, sa valeur du pluriel ou du singulier selon son nombre, sa nature (pronom, verbe, adjectif...), la famille de mots à laquelle il appartient, les thèmes auxquels il peut se rapporter et sa ou ses définitions.

Connaissances de traduction. Ces connaissances correspondent, pour chaque langue, à une base de mots où chaque mot est étiqueté par ses différentes traductions.

```

<BaseDeTextes idMax="45">
- <texte id="1" auteur="Jean de la Fontaine" titre="Le Corbeau et le Renard" annee="?" genre="conte" langue="francais"
  nombreMots="131">
  Maître Corbeau, sur un arbre perché, Tenait en son bec un fromage. Maître Renard, par l'odeur alléché, Lui tint à peu près
  ce langage : "Hé ! bonjour, Monsieur du Corbeau. Que vous êtes joli ! que vous me semblez beau ! Sans mentir, si votre
  ramage se rapporte à votre plumage, Vous êtes le Phénix des hôtes de ces bois." A ces mots le Corbeau ne se sent pas
  de joie ; Et pour montrer sa belle voix, Il ouvre un large bec, laisse tomber sa proie. Le Renard s'en saisit, et dit : "Mon bon
  Monsieur, Apprenez que tout flatteur Vit aux dépens de celui qui l'écoute : Cette leçon vaut bien un fromage, sans doute."
  Le Corbeau, honteux et confus, Jura, mais un peu tard, qu'on ne l'y prendrait plus.
</texte>
<texte id="2" auteur="William Shakespeare" titre="?" annee="?" genre="theatre" langue="anglais"
  nombreMots="10">To be or not to be: that is the question</texte>
+ <texte id="3" auteur="Jean de la Fontaine" titre="La Cigale et la Fourmi" annee="?" genre="conte" langue="francais"
  nombreMots="109"></texte>
...
- <texte id="45" auteur="Pierre de Ronsard" titre="Quand vous serez bien vieille" annee="1578" genre="poesie"
  langue="francais" nombreMots="109">
  Quand vous serez bien vieille, au soir à la chandelle, Assise auprès du feu, dévidant et filant, Direz chantant mes vers, en
  vous émerveillant : "Ronsard me célébrait du temps que j'étais belle." Lors vous n'aurez servante oyant telle nouvelle, Déjà
  sous le labeur à demi sommeillant, Qui au bruit de Ronsard ne s'aïlle réveillant, Bénissant votre nom de louange
  immortelle. Je serai sous la terre, et fantôme sans os Par les ombres myrteux je prendrai mon repos ; Vous serez au foyer
  une vieille accroupie, Regrettant mon amour et votre fier dédain. Vivez, si m'en croyez, n'attendez à demain : Cueillez dès
  aujourd'hui les roses de la vie.
</texte>
</BaseDeTextes>

```

Figure 6-22 : Extrait de la base de textes d'Adapte.

Connaissances de conjugaison. Ces connaissances sont propres à la langue choisie pour faire travailler l'apprenant. Elles portent sur la conjugaison des verbes à tous les temps et tous les modes.

L'ÉTAPE 2 DU PROCESSUS DE GÉNÉRATION : LA DÉFINITION DES CONTRAINTES

Définition des contraintes de sélection du texte. La sélection d'un texte peut se faire de quatre manières : la saisie du texte, l'importation du texte, la sélection du texte dans la base de textes ou la définition de contraintes de sélection du texte dans la base de textes. Les contraintes peuvent porter sur chacune des métadonnées permettant de décrire un texte dans la base de textes. Dans le cas de la saisie ou de l'importation du texte, le système doit vérifier que tous les mots du texte sont bien dans la base de mots (connaissances sur le vocabulaire). En cas de mots inconnus, l'utilisateur, aidé par le système, doit fournir toutes les connaissances relatives à chacun des mots. À ces contraintes de sélection de texte, il faut ajouter le nombre de textes à fournir à l'apprenant dans l'exercice.

Définition des contraintes de sélection des mots. Pour contraindre la sélection des mots, l'utilisateur indique comment choisir ces mots dans le texte ou dans la base de mots (connaissances sur le vocabulaire). Dans le cas de la saisie ou de l'importation d'un texte, il peut indiquer précisément quels mots seront sélectionnés. Il peut aussi demander au système de sélectionner automatiquement des mots selon certaines contraintes (taille des mots, exclure les mots commençant par une majuscule, exclure les nombres, les verbes...). Dans le cas de la sélection automatique du texte, les mots seront choisis automatiquement selon des contraintes. Dans le cas où l'exercice ne porte pas sur un texte (par exemple, exercice de définition ou de traduction de mots), l'utilisateur peut fournir une liste de mots et/ou définir des contraintes de sélection de mots dans la base. Aux contraintes de sélection de mots, il faut ajouter, pour chaque texte ou liste de mots, le nombre des mots à sélectionner (toutes les occurrences, n mots, etc.).

Définition des actions à faire sur les mots. L'utilisateur choisit s'il veut créer un exercice de définition, de traduction, de transposition de texte, d'analyse de texte ou d'identification de parties du texte.

Définition des contraintes de mise en page. Ces contraintes dépendent du type d'exercices à créer. Pour les exercices d'identification, il faut indiquer si les mots à identifier sont à supprimer du texte, et si oui, s'il faut fournir la liste de mots à l'apprenant et sous quelle forme (tous les mots, une occurrence de chaque mot, dans quel ordre). Pour les exercices d'analyse de texte, il faut indiquer s'il faut écrire le texte une ligne sur deux pour permettre à l'apprenant d'annoter le texte. Pour les exercices portant sur le sens des mots ou d'un texte, il faut indiquer si les mots ou le texte à fournir sont ceux de la langue d'origine, de la langue d'arrivé, ou parfois l'une, parfois l'autre, puisqu'il est intéressant dans le cas « X a pour sens Y » de demander à l'apprenant de donner parfois X, parfois Y. Pour les exercices de transposition de texte, il faut indiquer s'il faut fournir des lignes permettant à l'apprenant de recopier le texte sur la feuille d'exercices.

L'ÉTAPE 3 DU PROCESSUS DE GÉNÉRATION : LA GÉNÉRATION DE L'EXERCICE ET DE SA SOLUTION

Sélection du texte. Le choix du texte peut se faire aléatoirement parmi les textes fournis par l'utilisateur ou dans la base de textes en respectant les contraintes de sélection. Il y a autant de choix à faire que l'exercice contiendra de textes.

Étiquetage du texte. Pour chaque mot du texte, il faut trouver l'information nécessaire à la construction de l'exercice. Par exemple, identifier les verbes du texte se fait à partir des connaissances de conjugaison, pour trouver l'infinitif des verbes du texte, et des connaissances de vocabulaire, pour savoir qu'un mot est un verbe.

Sélection des mots. Si l'exercice porte sur un texte, le système choisit les mots dans le texte selon les contraintes de sélection des mots contenues dans la structure d'exercices. Si l'exercice porte sur une liste de mots, le système les choisit dans la base de mots selon des contraintes et/ou dans la liste fournie par l'utilisateur.

Modification des mots. Le système cherche dans les bases de connaissances le(s) sens des mots choisis (définition, traduction) ou les transforme grâce aux connaissances de vocabulaire (par exemple singulier, pluriel) ou les conjugue grâce aux règles de conjugaison de la base de conjugaison.

L'ÉTAPE 4 DU PROCESSUS DE GÉNÉRATION : LA MISE EN PAGE DE L'EXERCICE

Mise en page. Cette étape dépend de l'exercice que l'on souhaite créer. Par exemple, pour le texte à trous, le système regardera si la liste des mots sélectionnés doit être fournie ou non et présentera ensuite l'énoncé, le texte, et le cas échéant la liste de mots, selon les contraintes de mise en page définies par l'utilisateur et stockées dans la structure d'exercices. Pour les définitions et les traductions, le système affiche soit le mot, soit son sens, pour que l'apprenant fournisse la partie manquante. Le système fournit également la correction pour l'utilisateur.

Lors de la création d'un exercice, toutes les étapes du processus de génération ne sont pas forcément utiles. Cela dépend du type d'exercices à créer. Par exemple, pour créer un texte à trous, l'utilisateur va effectuer les quatre sous-étapes de la définition de contraintes mais le système n'étiquètera les mots du texte que si les contraintes de sélection d'un mot le demandent. De même, la sous-étape de modification des mots n'est pas nécessaire dans ce cas. Autre exemple : pour créer un exercice de définition de mots, le système fera seulement les sous-étapes de sélection des mots, de modification des mots et de mise en page. Les autres sous-étapes de la génération de l'exercice sont inutiles.

Des exemples de structures d'exercices et d'exercices issus de ces structures ont été fournis lors de la définition du patron d'exercices « Travail sur texte » (cf. section 6.3.3.c page127).

6.3.4. BILAN SUR L'ADAPTATION DES ACTIVITÉS PAPIER

Dans la section 6.2, nous avons présenté les principes de l'approche GEPPETO. Cette approche est résumée sur la première ligne de la Figure 6-23 ci-dessous. Elle contient un méta-modèle d'activités qui est utilisé par un processus de création de modèles pour définir des modèles d'activités papier ou logicielles. Chaque modèle d'activités peut être utilisé par un processus de définition de contraintes pour créer des contraintes sur activités. Ce processus a deux étapes : la proposition d'une interface, adaptée au modèle de l'activité, permettant de contraindre les activités (cf. ① sur la Figure 6-23) et l'enregistrement de ces contraintes (cf. ② sur la Figure 6-23). Un ensemble de contraintes sur activités peut ensuite être utilisé par un processus de génération d'activités pour créer des activités pédagogiques, qu'elles soient papier ou logicielles. Ce processus a également deux étapes : l'utilisation des contraintes pour créer une activité (cf. ③ sur la Figure 6-23) et la mise en page de cette activité (cf. ④ sur la Figure 6-23).

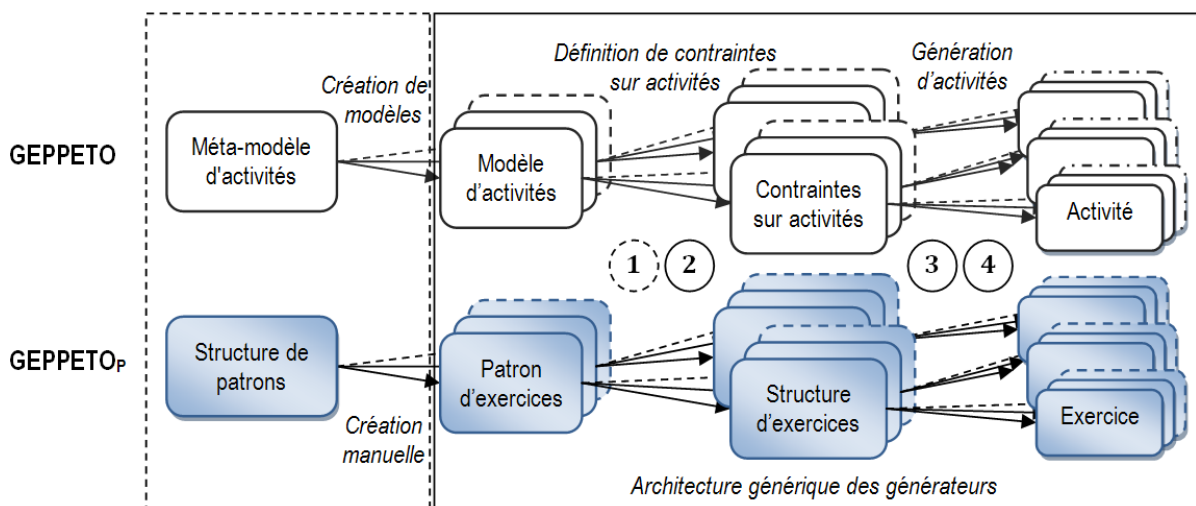


Figure 6-23 : Parallèle entre l'approche GEPPETO et sa déclinaison GEPPETO_P.

Dans la section 6.3.3, nous avons présenté les modèles et processus de l'approche GEPPETO_P, déclinaison de l'approche GEPPETO pour la personnalisation des activités papier. Nous soulignons à présent le lien entre les modèles et processus de ces deux approches.

La structure de patrons correspond au méta-modèle d'activités pour les activités papier. À partir de cette structure de patrons, nous avons défini un ensemble de patrons d'exercices, correspondant aux modèles d'activités. Nous avons établi le processus de définition des modèles manuellement.

Les processus de définition de contraintes et de génération d'activités sont intégrés dans chacun des générateurs d'exercices proposés. Ces générateurs respectent une architecture générique dans laquelle on retrouve trois des quatre étapes composant les deux processus. La première étape, correspondant à la proposition d'une interface adaptée au modèle de l'activité, est faite en amont des générateurs. Elle correspond au choix du générateur adapté au patron d'exercices, et donc au modèle de l'activité. La seconde étape permet de définir des contraintes sur activités, qui correspondent aux structures d'exercices pour les activités papier. La troisième étape crée un exercice et la quatrième met en page cet exercice de manière à fournir d'une part un énoncé pour l'apprenant et d'autre part, sa correction pour l'enseignant.

6.4. GEPPETOs: ADAPTER DES EIAH

Dans le cadre de nos recherches, nous nous intéressons à la personnalisation d'activités logicielles que nous définissons comme une séquence d'activités sur un logiciel pédagogique. Personnaliser une activité logicielle revient donc à personnaliser le contenu de la séquence, mais aussi le logiciel pour qu'il soit adapté à l'apprenant. Nous allons voir dans cette section ce qui est personnalisable dans un EIAH et comment cette personnalisation est possible. Nous proposons ensuite une classification des EIAH que nous souhaitons personnaliser, avant de définir l'approche GEPPETOs permettant la personnalisation de ces EIAH. Nous terminons cette section par un bilan.

6.4.1. PERSONNALISER QUOI ET COMMENT ?

Pour déterminer les grands types de personnalisation dans un EIAH, ainsi que la manière dont elle peut être mise en place, nous avons regardé un large éventail de logiciels pédagogiques. Parmi ceux-ci, nous avons retenu une sélection de trente logiciels de manière à avoir des logiciels de types, de provenance, de public cible et de domaine variés.

Ainsi, ces trente logiciels sont destinés à une utilisation individuelle par un apprenant. Ils sont de différents types : tuteurs intelligents comme ANDES [VanLehn et al. 2005] ou AMBRE-ADD [Nogry et al. 2004], micromondes comme APLUSIX [Nicaud et al. 2003], simulateurs comme TELEOS [Vu Minh et al. 2006] ou encore exercices comme EXOMATIKS [Exomatiks]. Ils sont issus de la recherche comme SQL-TUTOR [Mitrovic 1998] ou ACTIVEMATH [Melis et al. 2001], des pratiques d'enseignants et/ou du développement libre comme LILIMATH [LiliMath 2000] et MATHENPOCHE [Mathenpoche 2002] ou issus du commerce comme ORTHOGRAMME [Orthogramme 1999]. Ils sont de plus destinés à des publics variés allant de l'école maternelle au lycée, ainsi que pour la formation professionnelle. Ils portent sur des domaines d'apprentissage divers (mathématiques, chirurgie orthopédique, allemand...). La liste complète des logiciels étudiés et leur description sont fournies en Annexe A page 283.

Nous avons effectué une étude systématique des logiciels pédagogiques et établi que dans ces logiciels, la personnalisation peut porter sur 5 facettes : les **activités** proposées, la **séquence des activités**, les **fonctionnalités** disponibles, l'**interface** du logiciel et les **rétroactions** proposées aux apprenants. La Figure 6-24 indique pour chaque facette la manière de la personnaliser.

Facette de la personnalisation	Manière d'agir sur une facette
Les activités proposées	En générant de nouvelles activités ou en choisissant une activité adaptée parmi celles contenues dans le logiciel.
La séquence des activités	En choisissant le nombre d'activités et l'ordre dans lequel les activités sont proposées.
Les fonctionnalités disponibles	En autorisant l'accès à l'aide, au diagnostic, aux compagnons, aux outils de dessins, de calculs, etc. ; en modifiant la temporalité d'accès : constamment, à certains moments, avec un nombre d'accès autorisés, etc.
L'interface du logiciel	En choisissant la langue du logiciel, en paramétrant la police (taille, style, etc.), en adaptant les couleurs (pour les dyslexiques ou les daltoniens par exemple), en activant la synthèse vocale, etc.
Les rétroactions proposées aux apprenants	En autorisant les feedbacks et en changeant les messages qu'ils contiennent pour adapter le vocabulaire ou le contenu de ceux-ci.

Figure 6-24 : Facettes de la personnalisation d'un EIAH.

Après identification de ces cinq facettes de la personnalisation d'un EIAH, nous nous sommes intéressée à la façon de mettre en œuvre cette personnalisation par un enseignant ou par le module pédagogique d'un EIAH. Pour cela, nous avons étudié comment les EIAH pouvaient être configurés avant d'être utilisés par un apprenant.

Un EIAH est classiquement paramétré par un **fichier de configuration** ou par une **interface** de configuration. Les activités peuvent alors être générées automatiquement ou choisies dans une liste prédéfinie dépendant éventuellement des contraintes exprimées par le paramétrage.

La Figure 6-25 résume les différentes possibilités de personnalisation des logiciels pédagogiques. On retrouve les facettes de la personnalisation en colonne et les moyens de mettre en place cette personnalisation en ligne. Chaque case contient des exemples de logiciels permettant de personnaliser une facette du logiciel selon un moyen donné. Les cases relatives à la personnalisation de la dernière facette, les rétroactions, sont vides puisque nous n'avons pas trouvé dans notre corpus initial de logiciel permettant cette personnalisation. Néanmoins, lors d'échanges scientifiques avec des chercheurs du cluster ISLE - Personnalisation des EIAH³ [ISLE-EIAH 2005], cette facette est apparue dans le cahier des charges de certains EIAH en cours de développement. Sur cette figure, les logiciels indiqués comme paramétrables par des interfaces sont ceux qui sont paramétrables uniquement via leur interface, sans création de fichier de configuration. Pour les autres, ceux possédant une interface de configuration agissant sur des fichiers ou ceux ne possédant que des fichiers de configuration, ils sont indiqués comme paramétrables par des fichiers de configuration.

Par exemple, le logiciel AMBRE-ADD [Nogry et al. 2004] permet de personnaliser les activités proposées grâce à un générateur ou en choisissant les activités parmi celles contenues dans l'application. Ces activités peuvent être combinées pour former des séquences d'utilisation stockées dans des fichiers de configuration. De plus, les fonctionnalités et l'interface du logiciel peuvent être modifiées en agissant sur les fichiers de configuration. Ce logiciel possède de plus un logiciel administrateur permettant de créer ou modifier ces fichiers de configuration.

³ Projet financé par la région Rhône-Alpes.

		Facettes de la personnalisation				
		Activité	Séquence d'activités	Fonctionnalité	Interface	Rétroaction
Moyen de mettre en place la personnalisation	Fichier de configuration	AMBRE-ADD [Nogry et al. 2004] EXOMATIKS [Exomatiks] ROBOTEACH [Leroux 1998]	AMBRE-ADD [Nogry et al. 2004] ROBOTEACH [Leroux 1998] TRI [Jean-Daubias 2007] MATHENPOCHE [Mathenpoche 2002]	AMBRE-ADD [Nogry et al. 2004] TRI [Jean-Daubias 2007]	AMBRE-ADD [Nogry et al. 2004] EXOMATIKS [Exomatiks]	
	Interface de configuration	ACTIVEMATH [Melis et al. 2001] ANDES [VanLehn et al. 2005] LILIMATH [LiliMath 2000] APLUSIX [Nicaud et al. 2003]		ROBOTEACH [Leroux 1998] APLUSIX [Nicaud et al. 2003] TABLES MULTIMEDIA [Camier 2003]	ACTIVEMATH [Melis et al. 2001]	

Figure 6-25 : Exemples d'EIAH illustrant les moyens de mettre en œuvre les différentes facettes de la personnalisation des EIAH.

Enfin, pour savoir comment agir de l'extérieur de l'EIAH sur son paramétrage, nous avons étudié les différents fichiers de configuration et les interfaces ayant un impact sur le paramétrage de onze des trente logiciels (cf. quelques exemples en Annexe H page 345). Cette étude nous a permis de voir qu'il est possible de décrire les paramètres intervenant dans le paramétrage d'un EIAH grâce à un formalisme commun : un ensemble de paramètres lié à un ensemble de règles qui contraignent ces paramètres. Ces paramètres appartiennent à des types prédéfinis et leurs valeurs sont dépendantes de l'EIAH à paramétrer.

En conclusion de l'étude que nous avons faite sur 30 EIAH, nous pouvons observer que la personnalisation d'un EIAH peut porter sur 5 facettes : le choix ou la création des activités, l'agencement de ces activités pour former des séquences de travail, les fonctionnalités offertes à l'apprenant, l'interface et les rétroactions proposées par l'EIAH.

Les paramètres agissant sur ces facettes peuvent être représentés dans des fichiers de configuration ou modifiables via l'interface de l'EIAH.

Chacun de ces paramètres peut être décrit grâce à un formalisme commun pour pouvoir être présenté de façon uniforme à un enseignant et, ainsi, permettre une personnalisation plus aisée d'EIAH hétérogènes.

Cette présentation uniforme des paramètres agissant sur les EIAH doit être accompagnée des compétences associées à chacun des paramètres. En effet, pour qu'un enseignant puisse modifier le paramétrage d'un EIAH, il a besoin de savoir quelle compétence est associée à chaque paramètre et quelle influence va avoir le changement d'un paramètre sur l'EIAH.

6.4.2. CLASSIFICATION DES EIAH SELON LEUR PERSONNALISATION

Dans notre contexte de personnalisation unifiée des activités d'apprentissage, nous souhaitons personnaliser les EIAH de manière externalisée. De ce fait, nous souhaitons instancier l'approche GEPPETO pour qu'un système personnalise les EIAH. Nous avons donc classé les EIAH en fonction de leur possibilité de personnalisation par un logiciel externe à l'EIAH.

Dans cette classification et dans les propositions que nous présentons dans la suite de cette section, nous utilisons le terme d'EIAH dans une acception réduite de sa signification. Nous l'employons pour parler des logiciels pédagogiques, issus de la recherche, du commerce, du développement libre ou des pratiques d'enseignants, mettant en œuvre une situation permettant l'apprentissage de connaissances par un apprenant. Nous excluons donc les logiciels mettant en œuvre des situations d'apprentissage collectives ou collaboratives. Nos propositions ne sont valables que pour des logiciels pédagogiques pouvant être personnalisés.

Suite à l'étude de trente logiciels pédagogiques présentée dans la section précédente, nous proposons une classification des EIAH selon leur personnalisation. La Figure 6-26 présente cette classification : on peut y voir les différentes façons dont un EIAH peut être personnalisé et de quelle manière cette personnalisation peut être faite par un système externe à l'EIAH.

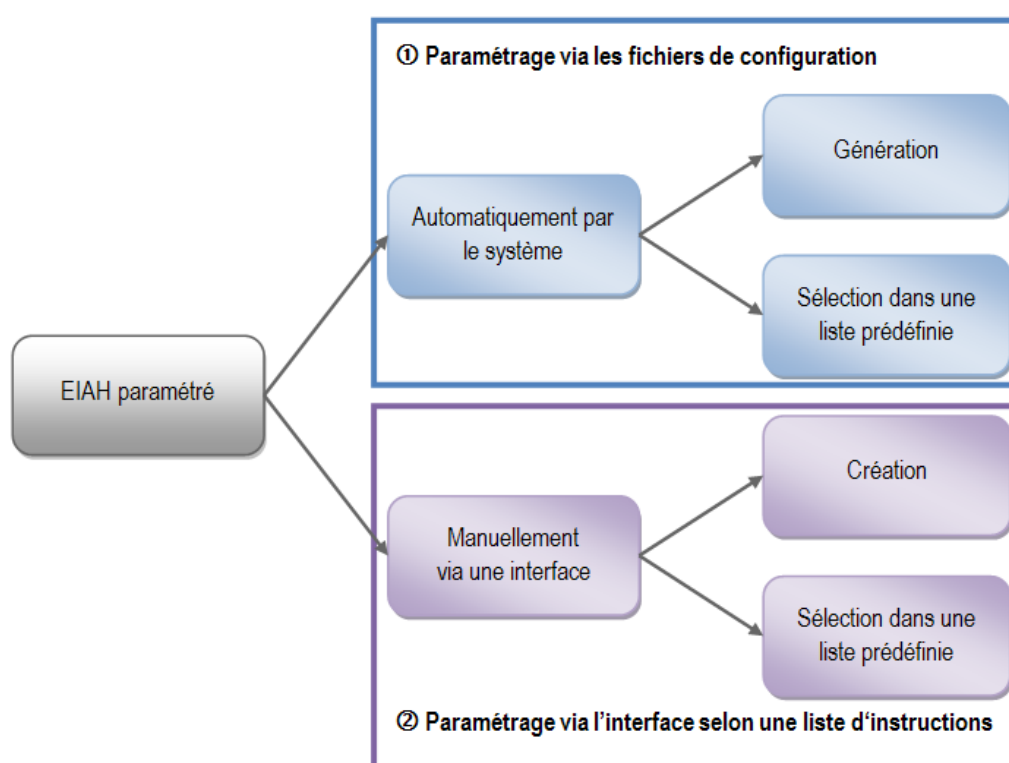


Figure 6-26: Classification des EIAH selon la manière de les personnaliser à travers un système externe à l'EIAH.

Pour personnaliser un EIAH de manière externalisée, il y a donc deux possibilités : modifier les fichiers de configuration (cf. ① sur la Figure 6-26) ou fournir une liste d'instructions à suivre sur l'interface de paramétrage (cf. ② sur la Figure 6-26).

Pour modifier les fichiers de configuration de l'EIAH, il est nécessaire d'avoir des droits d'accès et d'écriture sur ces fichiers de configuration. De plus, les valeurs contenues dans ces fichiers de configuration peuvent être générées ou choisies dans une liste, car si certains EIAH possèdent des générateurs d'exercices qui peuvent être paramétrés, lancés et dont le résultat de la

génération peut servir à créer des séquences d'activités, d'autres EIAH ne possèdent pas de générateur, les activités sont prédéfinies dans le logiciel, et pour personnaliser le logiciel, il faut donc choisir une activité dans la liste des activités existantes.

Dans le cas de la personnalisation des EIAH via des fichiers de configuration, la personnalisation correspond à l'ensemble des fichiers de configuration de l'EIAH, modifiés pour contenir les valeurs propres à chaque apprenant.

Dans le cas de la personnalisation via une interface, la personnalisation correspond à un ensemble d'instructions précisant à l'enseignant ou à l'apprenant les choix à faire sur l'interface de paramétrage pour personnaliser le système.

Dans les deux cas, nous pouvons dire que personnaliser un EIAH de manière externalisée nécessite de connaître les paramètres impactant le paramétrage du logiciel, ainsi qu'une description technique indiquant comment, à partir des valeurs de chacun des paramètres, soit modifier ou créer les fichiers de configuration, soit fournir la liste des instructions à suivre sur l'interface de configuration.

6.4.3. PRINCIPE DE L'APPROCHE GEPPETO_s

Dans la section précédente, nous avons présenté la classification des EIAH selon leur personnalisation, classification sur laquelle nous nous basons pour instancier l'approche GEPPETO afin de permettre une personnalisation externalisée des EIAH. Nous nommons cette déclinaison de l'approche GEPPETO pour la personnalisation d'activités logicielles GEPPETO_s, avec S pour « Software ». GEPPETO_s est ainsi l'équivalent de l'approche GEPPETO_p que nous venons de présenter permettant la personnalisation des activités papier. Avant de voir un exemple d'utilisation de l'approche GEPPETO_s et de détailler les différents modèles et processus définis pour cette approche, nous définissons l'ensemble des termes utilisés.

La personnalisation externalisée d'un EIAH nécessite de pouvoir obtenir certaines connaissances sur l'EIAH. Tout d'abord, les propriétés caractérisant les activités et l'environnement permettent, pour chaque EIAH, de connaître le type d'activités proposées, ainsi que la manière dont il est possible de choisir ou de générer les activités de cet EIAH. Ces propriétés permettent également de savoir si l'EIAH propose des séquences d'activités personnalisables, des fonctionnalités paramétrables, etc. Pour personnaliser les activités et l'environnement proposés aux apprenants, il est donc nécessaire de disposer d'une description de l'ensemble des paramètres relatifs aux activités, aux séquences d'activités, aux fonctionnalités et/ou à l'interface de l'EIAH. Ensuite, les compétences pédagogiques associées à ces propriétés permettent d'aider l'enseignant à faire ses choix pédagogiques lors du paramétrage de l'EIAH. Enfin, pour être capable d'agir sur un EIAH, il faut pouvoir modifier les fichiers permettant sa personnalisation. Il est donc nécessaire d'avoir les informations techniques concernant l'EIAH, comme la présence ou non d'un générateur d'exercices, la place et le contenu des fichiers de configuration.

L'étude des normes et standards existants (cf. section 2.3) nous a permis de voir que ceux-ci ne permettaient pas à un expert de décrire un EIAH selon cette granularité.

Afin d'assister l'expert dans sa tâche complexe d'identification des connaissances nécessaires à la personnalisation externalisée d'un EIAH, nous proposons un cadre facilitateur de l'acquisition : le méta-modèle AKEPI (Acquisition of Knowledge Enabling Personalization of ILEs (Interactive Learning Environments)), un méta-modèle pour l'acquisition des connaissances permettant la personnalisation des EIAH. Ce méta-modèle définit les types d'informations

nécessaires pour personnaliser un EIAH et se compose de deux parties : la description pédagogique d'un EIAH et sa description technique.

Dans l'approche GEPPETO_s (cf. Figure 6-27), nous associons deux processus au méta-modèle AKEPI : le premier, le **processus AKASI**, permet d'instancier le **méta-modèle AKEPI** avec les connaissances propres à un EIAH *x* afin d'obtenir le **modèle OKEP/x** (Operational Knowledge Enabling Personalization of the ILE *x*), un modèle des connaissances opérationnelles permettant la personnalisation de l'EIAH *x*, et l'autre, le **processus OPIKSI**, permet d'utiliser le modèle OKEP/*x* afin de personnaliser l'EIAH *x*.

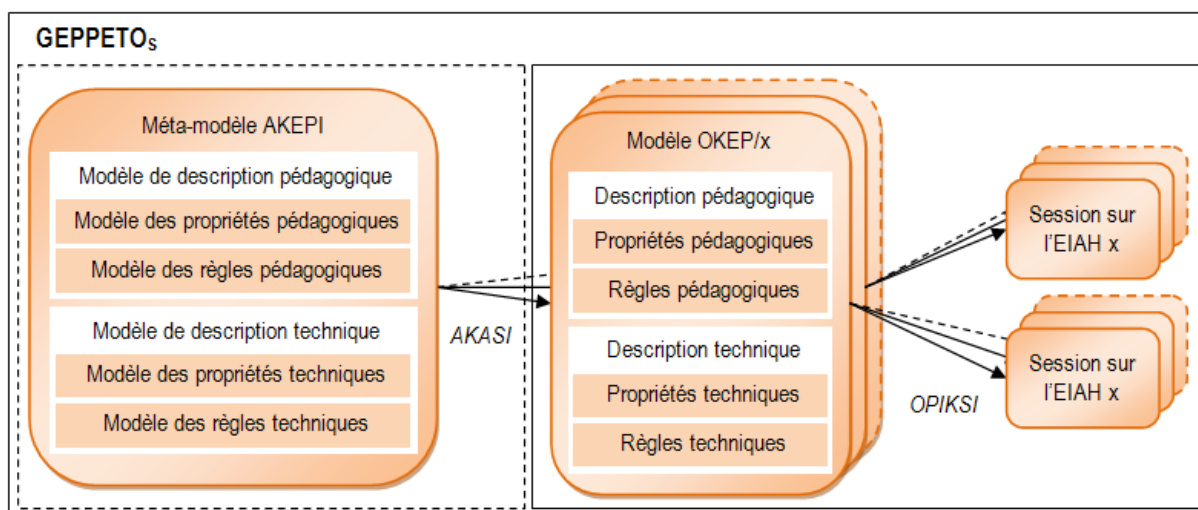


Figure 6-27 : Représentation graphique de l'approche GEPPETO_s.

A. EXEMPLES D'UTILISATION

Donnons un exemple concret d'utilisation pour mieux comprendre l'approche GEPPETO_s. Prenons un enseignant de CE2. Celui-ci souhaite utiliser différents logiciels en classe avec ses élèves : le logiciel AMBRE-ADD [Nogry et al. 2004] pour que ses élèves travaillent sur les problèmes additifs, le logiciel TRI [Jean-Daubias 2007] pour sensibiliser ses élèves au tri des déchets et le logiciel ABALECT [Chevé 2005] pour faire travailler ses élève en lecture.

Le logiciel AMBRE-ADD possède un module apprenant dont une copie d'écran est fournie sur la Figure 6-28 et un module enseignant permettant, entre autres, de créer des exercices adaptés aux besoins pédagogiques d'un enseignant (cf. Figure 6-29), de combiner ces exercices pour créer des séquences de travail (cf. Figure 6-31) et d'affecter les séquences à un apprenant donné (cf. Figure 6-34). Tous les choix faits par l'enseignant dans le module enseignant sont mémorisés dans des fichiers de configuration servant au module apprenant.

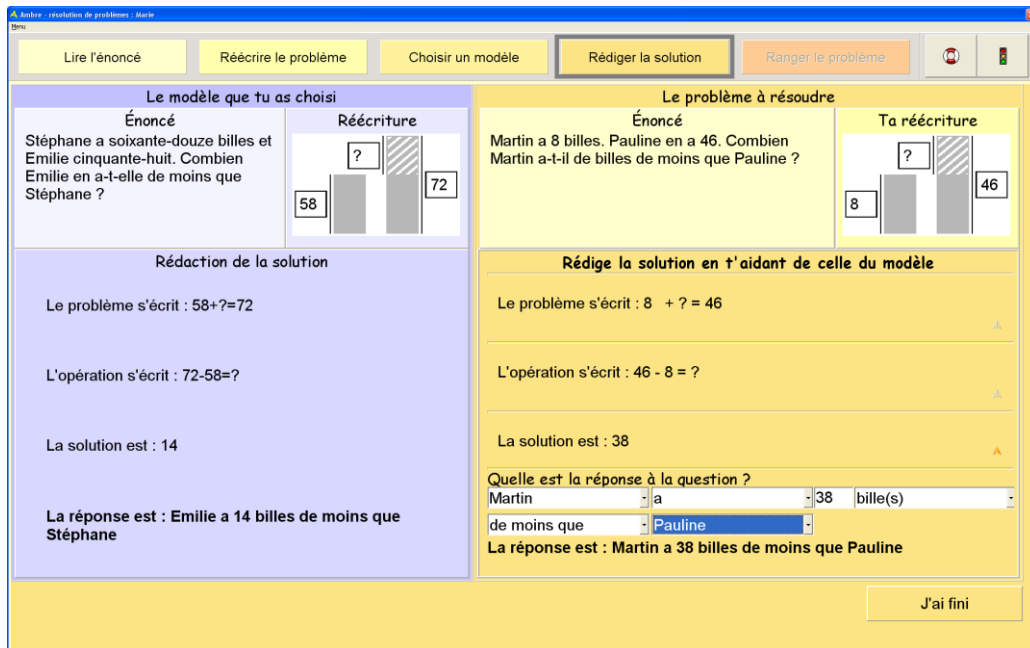


Figure 6-28 : Étape de rédaction de la solution du problème dans AMBRE-ADD [AMBRE-add 2005].

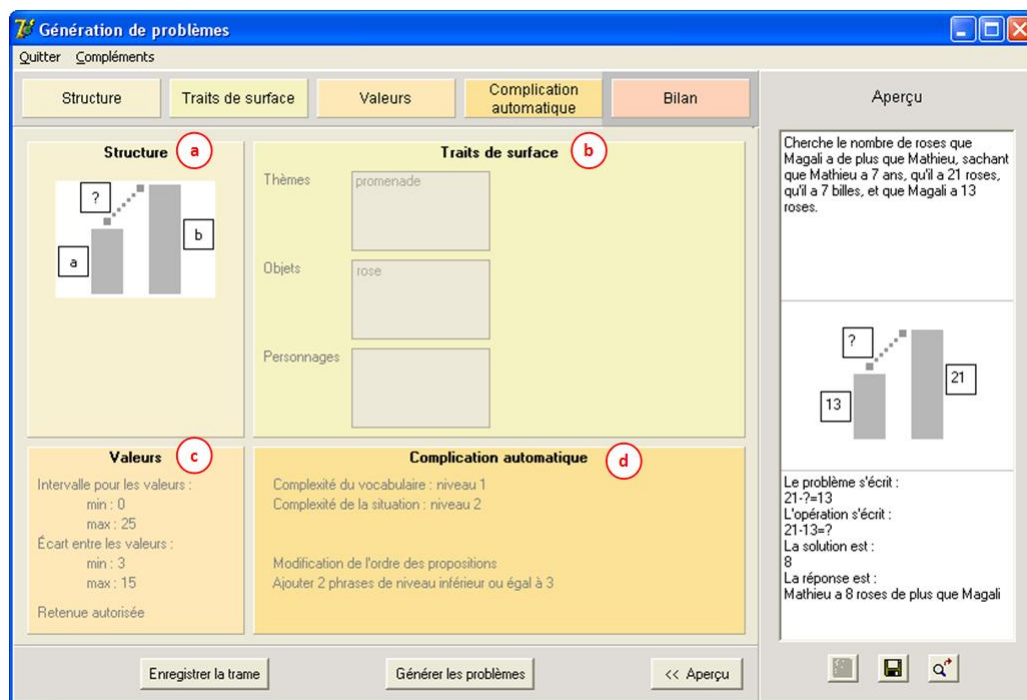


Figure 6-29 : Écran bilan de AMBRE-ENSEIGNANT récapitulant les contraintes de génération d'un problème [AMBRE-enseignant 2004].

```

consult('..\..\..\Commun\themes_fr').
appel_generation(
  1  [['c_comparaison_dif', 'de plus'],
  2  ['promenade',
     ['rose'],
     []],
     [0,0,0,0],
     [0,25],
     [3,15],
     [intervalle,1],
     3  [0,1,1,0,3,2]],
     4  apercu).

```

Figure 6-30 : Fichier de configuration du générateur GENAMBRE [AMBRE-enseignant 2004].

Ainsi, lorsque notre enseignant veut créer des exercices, les choix qu'il fait à l'interface (cf. Figure 6-29) sont traduits dans un prédicat Prolog (cf. Figure 6-30) pour être utilisés le moment venu par le générateur GENAMBRE. Sur cet exemple, on peut ainsi remarquer que l'enseignant a choisi :

- pour la structure de l'exercice, un problème de comparaison où l'inconnue porte sur la différence entre les deux nombres (cf. a sur la Figure 6-29), ce qui est traduit par le paramètre « *c_comparason_dif* » dans le prédicat Prolog (cf. ① sur la Figure 6-30);
- pour les traits de surface, d'avoir comme thème la promenade et comme objet des roses (cf. b sur la Figure 6-29), ce qui est traduit par les paramètres « *promenade* » et « *rose* » dans le prédicat Prolog (cf. ② sur la Figure 6-30);
- pour les valeurs du problème, de prendre des valeurs entre 0 et 25 et que l'écart entre les valeurs du problème soit compris entre 3 et 5 (cf. c sur la Figure 6-29). On retrouve ces choix dans les paramètres « *[0,25]* », « *[3,15]* » et « *intervalle* » du prédicat Prolog (cf. ③ sur la Figure 6-30).
- pour la complication automatique de l'énoncé, d'utiliser un vocabulaire de niveau 1, une situation de complexité 2, de modifier l'ordre des propositions et d'ajouter deux phrases inutiles à la résolution du problème (cf. d sur la Figure 6-29). On retrouve ces choix dans les paramètres « *[0,1,1,0,3,2]* » du prédicat Prolog (cf. ④ sur la Figure 6-30).

De même, lorsque l'enseignant assemble aléatoirement les exercices a12, b112, b123 et b312 (cf. Figure 6-31) pour créer la séquence 3 (cf. Figure 6-32), ses choix sont mémorisés dans un fichier XML (cf. Figure 6-33).

Enfin, lorsqu'il affecte les séquences 2 et 3 à l'élève Clothilde (cf. Figure 6-34), ses choix sont également mémorisés dans un fichier XML (cf. Figure 6-35).

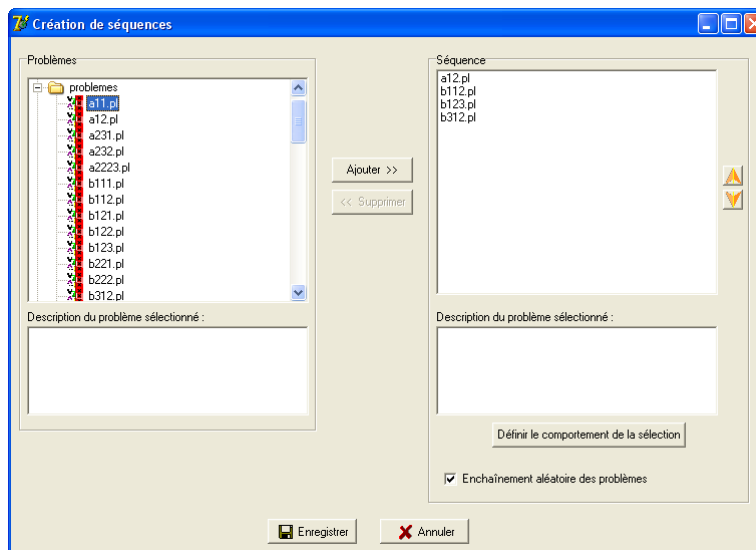


Figure 6-31 : Écran de création de séquences d'exercices de AMBRE-ENSEIGNANT [AMBRE-enseignant 2004].

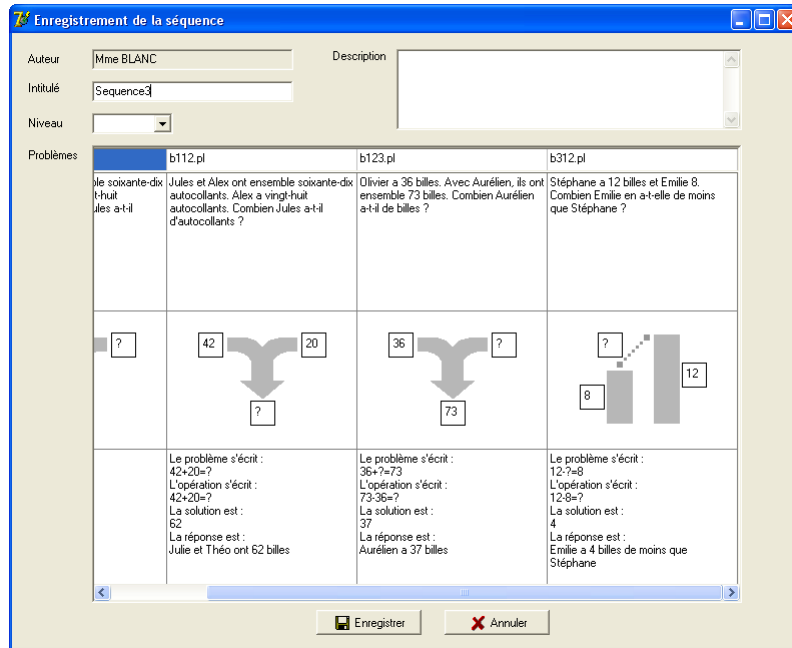


Figure 6-32 : Écran de sauvegarde d'une séquence d'exercices de AMBRE-ENSEIGNANT [AMBRE-enseignant 2004].

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
- <Sequences>
  <NbSequences>3</NbSequences>
  + <Sequence1>
  + <Sequence2>
  - <Sequence3>
    <Nom>Sequence3</Nom>
    <Problemes>a12.pl/b112.pl/b123.pl/b312.pl</Problemes>
    <ClassesPb>c_comparaison_dif/c_reunion_operande/c_reunion_res</ClassesPb>
    <Aleatoire>oui</Aleatoire>
    <Description />
  </Sequence3>
</Sequences>
```

Figure 6-33 : Fichier de configuration de AMBRE-ADD contenant les séquences d'exercices créées avec AMBRE-ENSEIGNANT [AMBRE-add 2005].

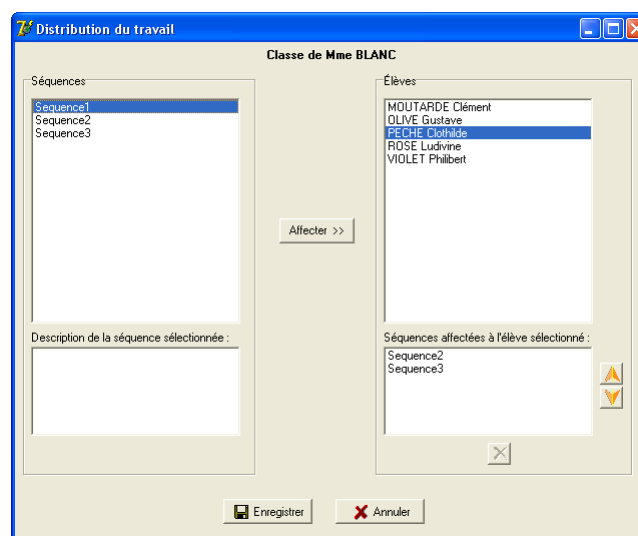


Figure 6-34 : Écran d'affectation d'une séquence d'exercices à un élève dans AMBRE-ENSEIGNANT [AMBRE-enseignant 2004].

```

<?xml version="1.0" encoding="ISO-8859-1" ?>
- <Classe>
  <NbElevés>5</NbElevés>
+ <Eleve1>
+ <Eleve2>
- <Eleve3>
  <Nom>PECHE</Nom>
  <Prenom>Clothilde</Prenom>
  <Sexe>Féminin</Sexe>
  <Sequences>Sequence2/Sequence3</Sequences>
  <Sequence2>b111.pl/f31.pl/b123.pl/a11.pl/g11.pl/a231.pl</Sequence2>
  <Sequence3>b312.pl/b123.pl/b112.pl/a12.pl</Sequence3>
</Eleve3>
+ <Eleve4>
+ <Eleve5>
</Classe>

```

Figure 6-35 : Fichier de configuration de AMBRE-ADD contenant la liste des élèves d'un enseignant et les séquences d'exercices affectées aux élèves dans AMBRE-ENSEIGNANT [AMBRE-add 2005].

Le logiciel TRI, dont une copie d'écran est fournie sur la Figure 6-36, ne possède lui aucune interface enseignant, mais chaque enseignant peut modifier ses fichiers de configuration pour personnaliser les activités que pourront faire ses élèves. Ce fichier de configuration (cf. Figure 6-37) indique les différentes étapes (cours, exercices) que doit faire l'apprenant (par exemple ① sur la Figure 6-37) ou indique que l'apprenant pourra naviguer librement dans le logiciel (par exemple ② sur la Figure 6-37).



Figure 6-36 : Activité de TRI [Jean-Daubias 2007].

```

Pour utiliser le fichier des séquences, il faut mettre la balise '[élève]'
suivi d'un espace et du nom de l'élève.
Puis définir à la ligne d'après la séquence pour cet élève, sinon inscrire
'libre', pour que ce dernier puisse librement choisir ses activités.

Modèle à utiliser pour la séquence d'un élève :
# 1 Cours : les différentes poubelles
# 2 Cours : recyclage
# 3 Jeu : correspondance déchets-poubelles
# 4 Jeu : correspondance couleurs-poubelles
# 5 Jeu : chercher l'intrus
# 6 Jeu : recyclage
# 7 Jeu : la chanson des poubelles
# 0 Quitter

Remarque : En utilisant '# 0 Quitter' l'élève quitte immédiatement le logiciel
quand il a fini sa séquence, à défaut, il revient sur la page d'accueil et pourra
choisir ses activités lui-même.

Compléments :
niveau 1/2 : niveau initial, pas de changement de niveau pendant l'exercice
aide : aide disponible
diagnostic : diag disponible
=====

[élève] Maëlys
libre

[élève] Timéo
# 1 Cours : les différentes poubelles
niveau 1
aide
diagnostic
# 2 Cours : recyclage
niveau 2
# 3 Jeu : correspondance déchets-poubelles
# 7 Jeu : la chanson des poubelles
# 0 Quitter

[élève] Maria
# 1 Cours : les différentes poubelles

[élève] Kevin
libre

[élève] Autre
Libre

```

Figure 6-37 : Fichier de configuration des séquences de travail de TRI [Jean-Daubias 2007].

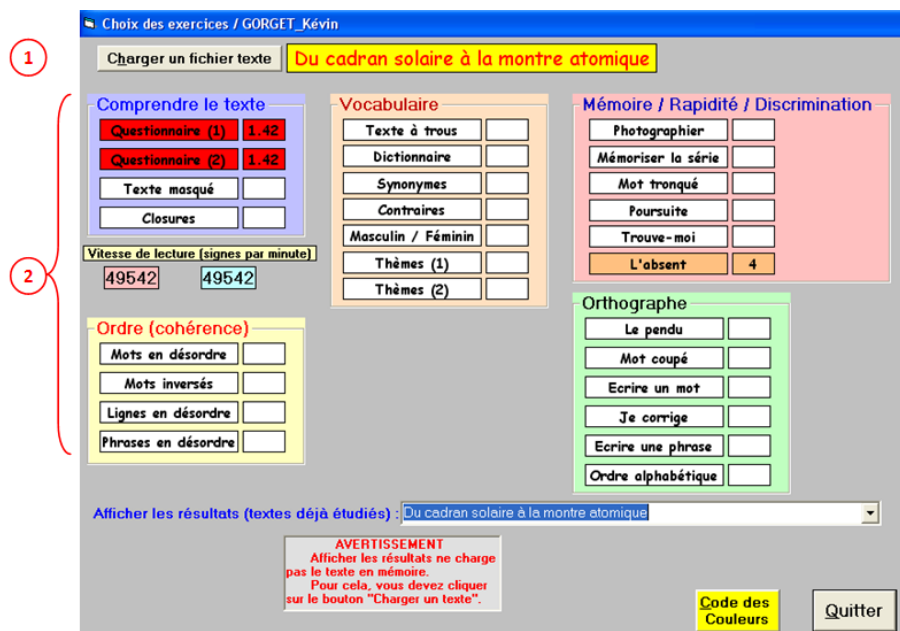


Figure 6-38 : Écran de choix d'un exercice par l'apprenant dans ABALECT [Chevé 2005].

Le logiciel ABALECT, dont une copie d'écran est fournie sur la Figure 6-38, ne possède, lui, aucune interface enseignant permettant de configurer le parcours des apprenants et n'est pas non plus paramétrable par des fichiers de configuration. Pour guider ses élèves, l'enseignant doit donc leur indiquer le texte à choisir (cf. ① sur la Figure 6-38), selon le niveau auquel le texte est destiné (cf. Figure 6-39), puis leur dire quel exercice faire (cf. ② sur la Figure 6-38) en leur précisant éventuellement la catégorie (comprendre le texte, cohérence, vocabulaire, mémoire, orthographe) à laquelle appartient l'exercice.

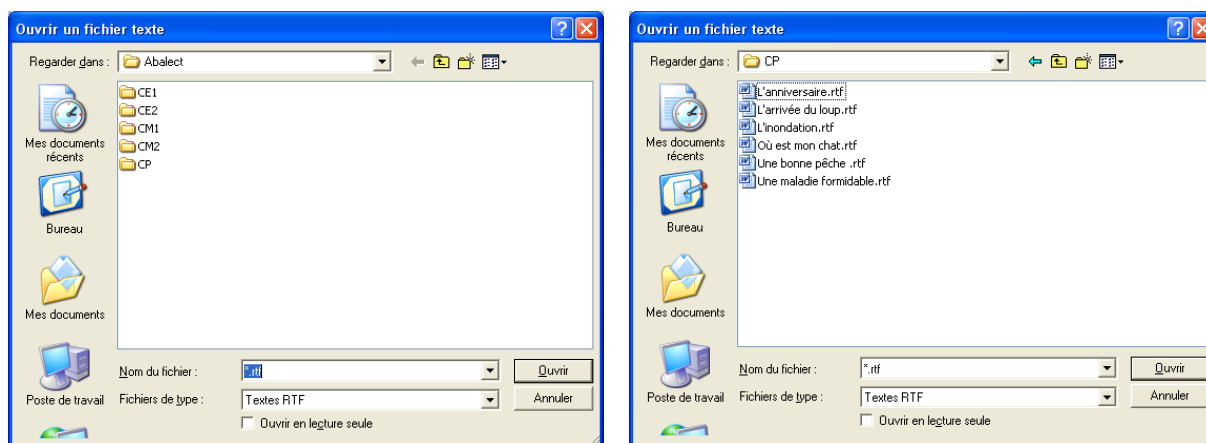


Figure 6-39 : Fenêtres de choix d'un texte selon son niveau dans ABALECT [Chevé 2005].

Ces exemples montrent bien la diversité des personnalisations possibles pour les logiciels pédagogiques. Avec l'approche GEPPETO, nous souhaitons fournir à l'enseignant une interface unique permettant de personnaliser différents logiciels pédagogiques. Ainsi l'enseignant n'aura plus besoin de connaître les différentes interfaces de personnalisation ou le contenu des fichiers de configuration pour adapter les logiciels, il n'utilisera plus qu'un seul outil lui permettant de faire les choix adéquats pour chaque logiciel. Cet outil lui proposera en sortie, dans les cas de AMBRE-ADD et de TRI, les fichiers de configuration pour chacun de ses élèves, et pour ABALECT, une liste d'instructions à suivre par chaque apprenant lui indiquant les choix à faire à l'interface pour obtenir un ou plusieurs exercices.

B. DÉFINITION FORMELLE DU MÉTA-MODÈLE AKEPI

Le méta-modèle AKEPI (Acquisition of Knowledge Enabling Personalization of ILEs) définit les types d'informations qu'un expert doit fournir pour créer, pour un EIAH x , le modèle OKEP/ x permettant sa personnalisation externalisée. Ces informations portent sur la description pédagogique et la description technique des EIAH (cf. Figure 6-27). La description pédagogique regroupe la liste des propriétés paramétrables d'un système, avec les compétences qui leur sont associées et la liste des règles permettant de gérer ces propriétés. La description technique regroupe toutes les informations nécessaires pour agir concrètement sur le système : localisation du système, chemin d'accès au générateur d'exercices ou à la base d'exercices, place et contenu des fichiers de configuration et les règles permettant de remplir ces fichiers.

Dans la suite de cette section, nous présentons le modèle de description pédagogique d'un EIAH et le modèle de description technique d'un EIAH, constituant le méta-modèle AKEPI. Par la suite, nous montrerons comment ce méta-modèle peut être utilisé pour définir les connaissances permettant la personnalisation d'un EIAH donné.

Il est à noter que comme les patrons d'exercices papier, le méta-modèle AKEPI est défini en XML Schema. Nous le présentons en utilisant une représentation graphique des schémas XML. La

sémantique de la représentation des XML Schema est fournie en Annexe E page 305, nous ne la redonnons donc pas sur chacune des figures présentant le méta-modèle.

B.1. LE MODÈLE DE DESCRIPTION PÉDAGOGIQUE

Le modèle de description pédagogique d'un EIAH est composé du modèle des propriétés pédagogiques et du modèle des règles pédagogiques (cf. Figure 6-27 page 154).

LE MODÈLE DES PROPRIÉTÉS PÉDAGOGIQUES

La Figure 6-40 présente la partie du méta-modèle AKEPI définissant les types de propriétés pédagogiques. Nous pouvons voir que ces propriétés sont organisées selon les cinq facettes de la personnalisation que nous avons identifiées dans de notre étude (cf. section 6.4.2). Formellement, le modèle *PedagogicalProperties* sera défini pour un EIAH identifié par *NameOfILE*. Ce modèle contiendra obligatoirement la description du contenu pédagogique de cet EIAH, *PedagogicalContent*, puis selon les EIAH, on pourra ajouter la description de l'organisation pédagogique, *PedagogicalOrganization*, des fonctionnalités, *Functionalities*, des rétroactions, *Feedback*, et/ou de l'interface, *Customization*.

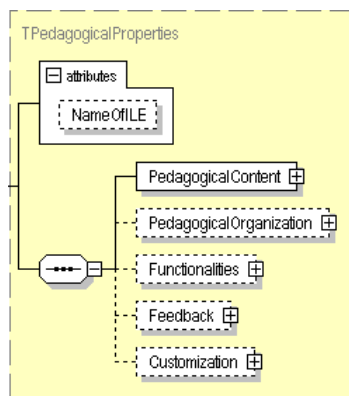


Figure 6-40 : Modèle des propriétés pédagogiques du méta-modèle AKEPI.

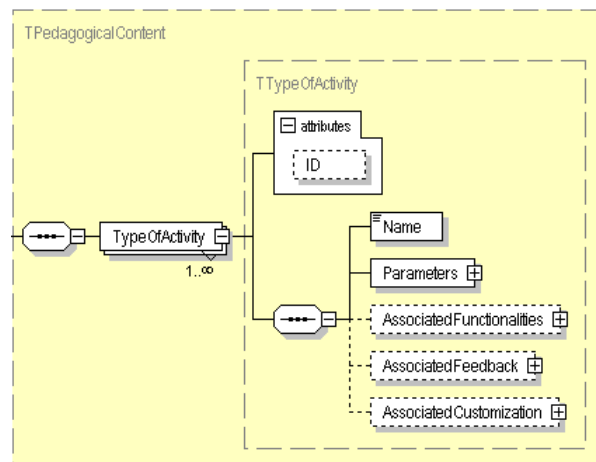
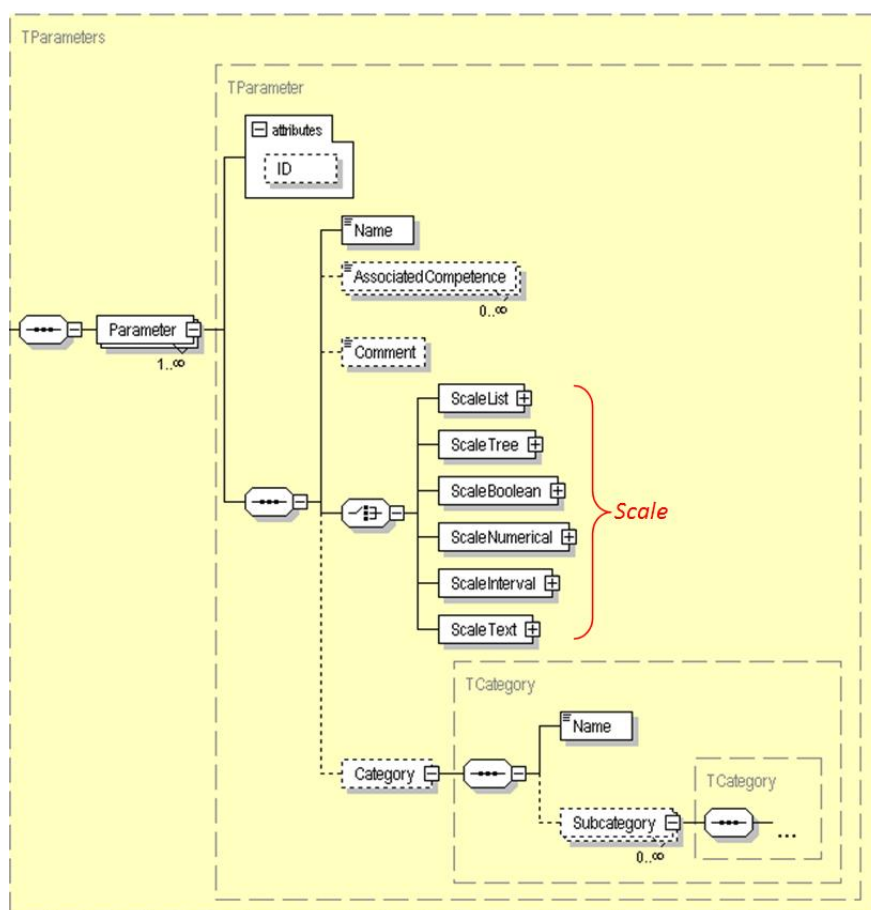


Figure 6-41 : Élément *PedagogicalContent* du méta-modèle AKEPI.

Le contenu pédagogique permet de décrire la façon dont les différents types d'activités vont pouvoir être choisis ou générés. Il est donc composé de la description d'un ou plusieurs types d'activités, *TypeOfActivity*.

Un type d'activités est défini, comme nous pouvons le voir sur la Figure 6-41, par son nom, *Name* ; la liste des paramètres, *Parameters*, permettant de choisir une activité de ce type dans une base de données ou de la générer avec le générateur contenu dans l'EIAH ; et éventuellement les paramètres des fonctionnalités, rétroactions ou de l'interface qui peuvent être associés à ce type d'activités (*AssociatedFunctionalities*, *AssociatedFeedback* et *AssociatedCustomization*). À chaque *TypeOfActivity* est associé un identifiant unique *ID*.

Figure 6-42 : Élément *Parameter* du méta-modèle AKEPI.

La liste des paramètres contient un ou plusieurs paramètres (cf. Figure 6-42), chaque paramètre étant défini par un nom, *Name* ; les compétences pédagogiques qui peuvent y être associées, *AssociatedCompetence* ; éventuellement un commentaire *Comment* ; une échelle respectant l'un des six types d'échelles proposés couvrant les différentes possibilités rencontrées dans les systèmes étudiés, *Scale* ; et éventuellement la catégorie, voire la sous-catégorie à laquelle il appartient, *Category*. Cette dernière précision permet d'organiser les paramètres de manière à regrouper les paramètres ayant des liens entre eux. De plus, comme les types d'activités, chaque paramètre possède un identifiant unique *ID*.

Le type d'échelles *ScaleList* (cf. Figure 6-43) permet de définir des listes énumérées de valeurs. Il comprend trois booléens, *Variable*, *Ordered* et *MultipleSelection*, qui permettent pour le premier, d'indiquer si la liste énumérée par l'expert pour un paramètre donné peut être augmentée des valeurs définies par l'enseignant ; pour le second, d'indiquer si la liste énumérée est ordonnée ou non ; et pour le troisième, d'indiquer si l'enseignant pourra choisir plusieurs valeurs pour ce paramètre. Il comprend ensuite une ou plusieurs valeurs, chacune définie par un nom et éventuellement des compétences associées.

Le type d'échelles *ScaleTree* (cf. Figure 6-44) permet de définir des valeurs organisées sous forme d'arbre. Il comprend, comme pour le type *ScaleList*, les booléens *Variable* et *MultipleSelection*. Il contient ensuite un ensemble de nœuds où chacun est composé d'un identifiant unique *ID*, d'un nom, éventuellement de compétences associées. Chaque nœud peut contenir un ou plusieurs nœuds afin de représenter l'arbre des valeurs.

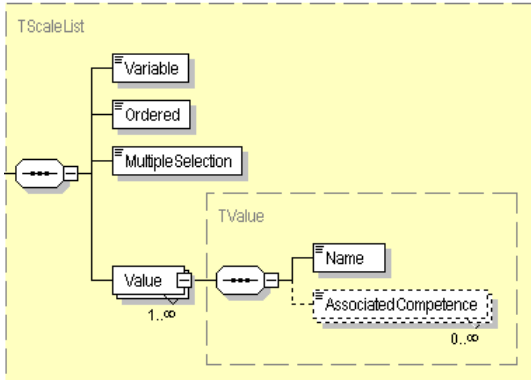


Figure 6-43 : Description d'une échelle de type *ScaleList* dans le méta-modèle AKEPI.

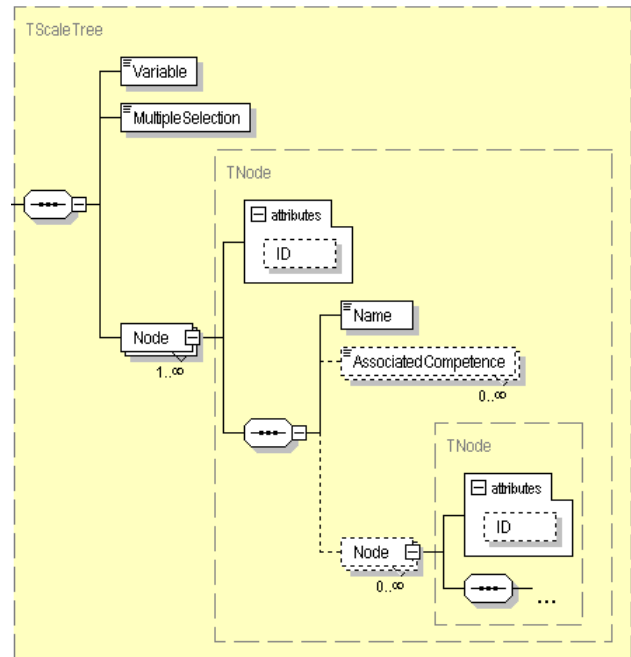


Figure 6-44 : Description d'une échelle de type *ScaleTree* dans le méta-modèle AKEPI.

Le type d'échelles *ScaleBoolean* (cf. Figure 6-45) peut être précisé en indiquant les compétences qui peuvent être associées à chacune des valeurs, *AssociatedCompetenceIfTrue* et *AssociatedCompetenceIfFalse*.

Le type d'échelles *ScaleNumerical* (cf. Figure 6-46) contient cinq champs optionnels. Les nombres *LowerBound* et *UpperBound* permettent de réduire l'intervalle des valeurs, initialement défini de $-\infty$ à $+\infty$. Le pas, *Step*, permet d'indiquer la granularité qu'aura la valeur du paramètre (entier, défini à 0,1 près, etc.). Les deux derniers champs permettent d'indiquer les compétences associées, en précisant quelles compétences sont associées aux nombres proches de la borne inférieure, *AssociatedCompetenceLowerBound*, et quelles compétences sont associées à la borne supérieure, *AssociatedCompetenceUpperBound*.

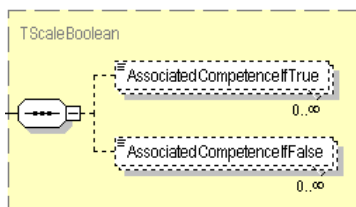


Figure 6-45 : Description d'une échelle de type *ScaleBoolean* dans le méta-modèle AKEPI.

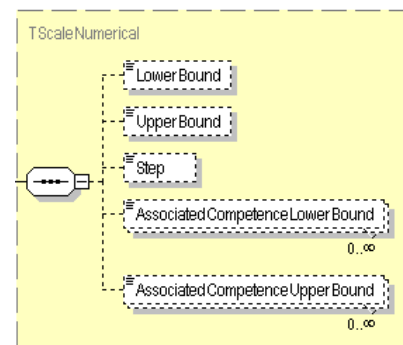


Figure 6-46 : Description d'une échelle de type *ScaleNumerical* dans le méta-modèle AKEPI.

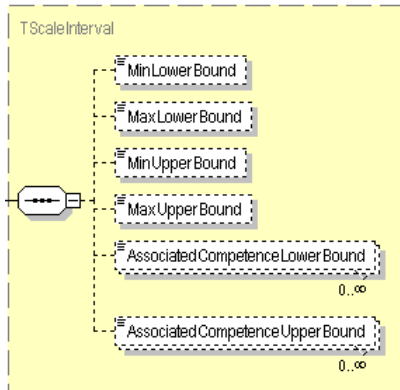


Figure 6-47 : Description d'une échelle de type *ScaleInterval* dans le méta-modèle AKEPI.

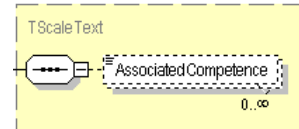


Figure 6-48 : Description d'une échelle de type *ScaleText* dans le méta-modèle AKEPI.

Le type d'échelles *ScaleInterval* (cf. Figure 6-47) contient six champs optionnels. Les deux premiers, *MinLowerBound* et *MaxLowerBound*, permettent de délimiter la valeur de la borne minimale de l'intervalle. Les deux suivants, *MinUpperBound* et *MaxUpperBound*, permettent de délimiter la valeur de la borne maximale de l'intervalle. Les deux derniers champs permettent d'indiquer les compétences associées à chacune des bornes de l'intervalle, *AssociatedCompetenceLowerBound* et *AssociatedCompetenceUpperBound*.

Enfin, le type d'échelles *ScaleText* (cf. Figure 6-48) permet de préciser que la valeur sera un champ texte, sans autre structure particulière. Cette échelle peut être augmentée de compétences associées.

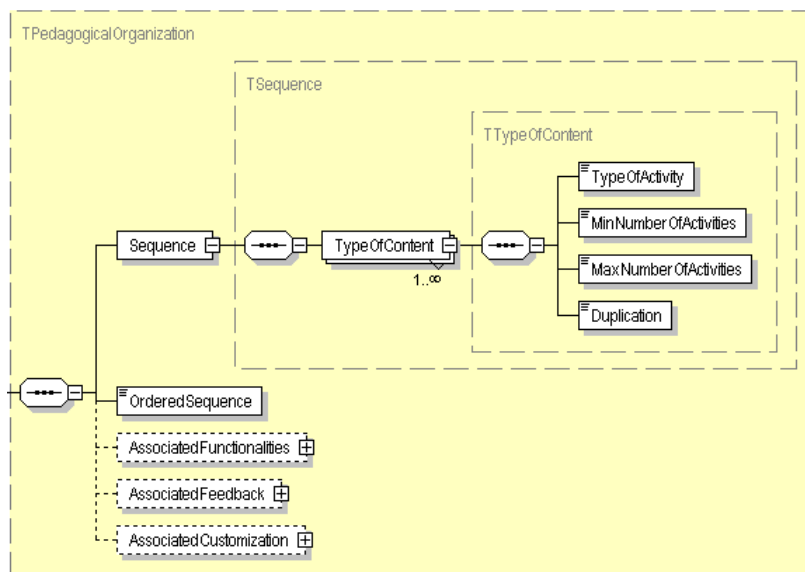


Figure 6-49 : Élément *PedagogicalOrganization* du méta-modèle AKEPI.

L'organisation pédagogique (cf. Figure 6-49) indique la manière dont sont formées les séquences d'activités. Ainsi, une séquence est formée d'un ou plusieurs *TypeOfContent*. Chacun de ces *TypeOfContent* fait référence à un *TypeOfActivity* défini précédemment, ainsi qu'à deux entiers, *MinNumberOfActivities* et *MaxNumberOfActivities*, qui indiquent le nombre d'activités de ce type pouvant être fournies consécutivement à l'apprenant. Le booléen *Duplication* permet d'indiquer si deux activités d'un même *TypeOfActivity* sélectionnées ou générées à partir du même ensemble de valeurs pour les paramètres qui les décrivent seront ou non strictement identiques. Le booléen *OrderedSequence* indique si les *TypeOfContent* doivent être fournis dans l'ordre

indiqué ou s'ils peuvent être fournis aléatoirement à l'apprenant. Enfin, des paramètres décrivant des fonctionnalités, des rétroactions ou des éléments de l'interface pouvant être associés à l'organisation pédagogique peuvent être ajoutés (*AssociatedFunctionalities*, *AssociatedFeedback* et *AssociatedCustomization*).

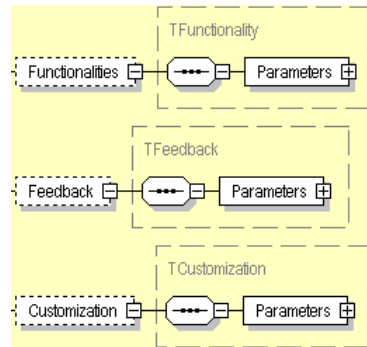


Figure 6-50 : Éléments *Functionalities*, *Feedback* et *Customization* du méta-modèle AKEPI.

Les éléments fonctionnalités, rétroactions et interface contiennent une description des paramètres permettant de les modifier (cf. Figure 6-50). Chacun de ces paramètres respecte la définition d'un paramètre que nous avons présentée précédemment (cf. Figure 6-42).

LE MODÈLE DES RÈGLES PÉDAGOGIQUES

La description pédagogique que nous venons de présenter regroupe les propriétés personnalisables d'un EIAH. Le méta-modèle AKEPI associe à ces propriétés des méta-règles permettant à l'expert de formaliser les liens et contraintes pouvant lier les propriétés qu'il a définies. Ainsi, l'expert peut définir des contraintes sur les propriétés en utilisant des règles dont les conditions et les conclusions respectent le format indiqué Figure 6-51. Une méta-règle est construite en utilisant pour sa condition un des quatre formats de la colonne de gauche et pour sa conclusion un des quatre formats de la colonne de droite.

SI	ALORS
Value(parameter i_1) = X_1	Value(parameter j_1) = Y_1
Value(parameter i_1) $\in \{X_1 .. X_n\}$	Le paramètre j_1 sera inaccessible
Value(parameter i_1) est non définie	ValueDomain (parameter j_1) = $\{Y_a .. Y_b\}$ avec $a \geq m$ et $b \leq n$, où m et n sont les bornes initiale
C_1 et C_2 où C_i est une contrainte sur la valeur d'un paramètre	C_1 et C_2 où C_i est une contrainte sur la valeur ou le domaine de valeur d'un paramètre

Figure 6-51 : Format des conditions (partie gauche) et des conclusions (partie droite) des règles pédagogiques du méta-modèle AKEPI.

B.2. LE MODÈLE DE DESCRIPTION TECHNIQUE

Sur le même principe que le modèle de description pédagogique, le modèle de description technique d'un EIAH est composé du modèle des propriétés techniques et du modèle des règles techniques (cf. Figure 6-27 page 154) que nous présentons maintenant.

LE MODÈLE DES PROPRIÉTÉS TECHNIQUES

La Figure 6-52 présente la partie du méta-modèle AKEPI définissant les propriétés techniques. Ces propriétés listent tous les éléments utiles à la mise en œuvre de la personnalisation du logiciel.

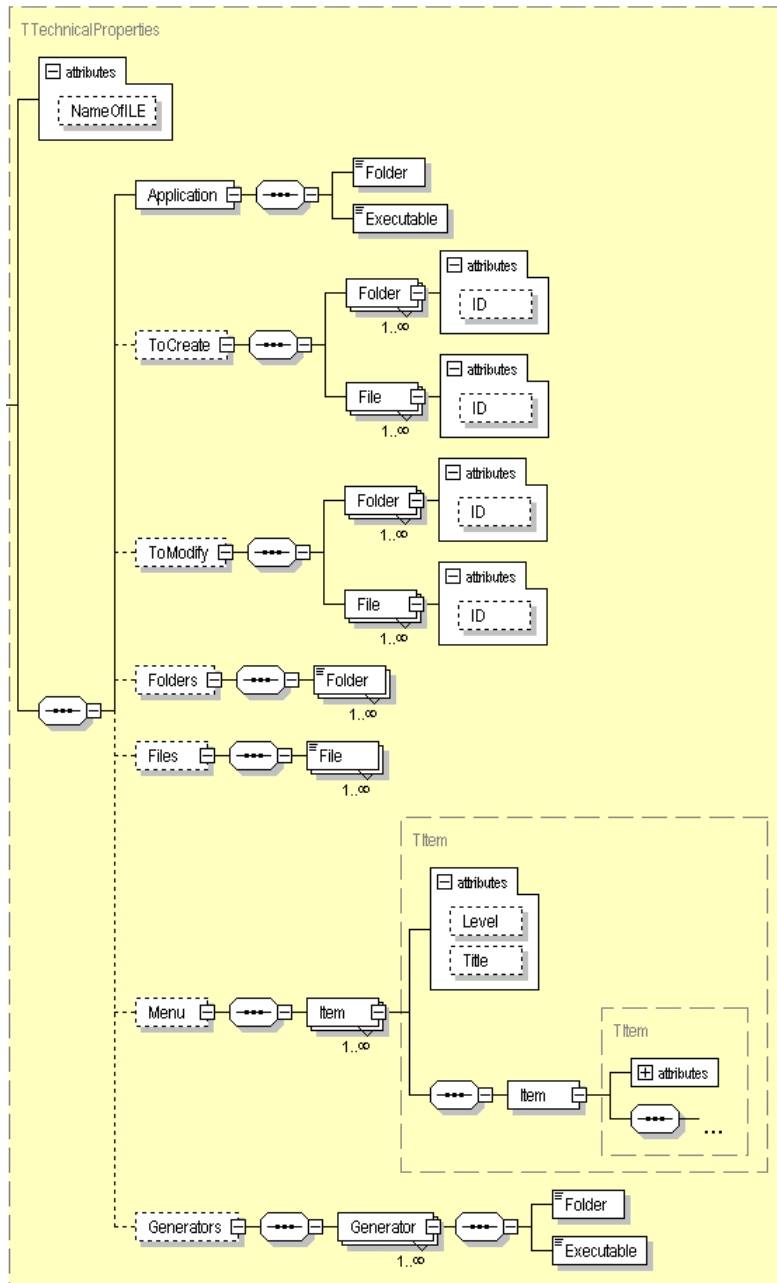


Figure 6-52 : Modèle des propriétés techniques du méta-modèle AKEPI.

Ainsi, *Application* indique le dossier, *Folder*, et l'exécutable, *Executable*, permettant d'ouvrir l'EIAH. *ToCreate* et *ToModify* listent les dossiers et fichiers de configuration qui doivent être respectivement créés et/ou modifiés lors de la personnalisation de l'EIAH. *Folders* et *Files* listent les dossiers et fichiers qui seront utiles, par exemple les dossiers contenant tous les exercices, ou un fichier permettant d'obtenir certaines données. *Menu* permet de décrire les menus de l'interface devant être manipulés par l'enseignant ou l'apprenant lors de la personnalisation. Enfin, *Generators* liste les générateurs disponibles dans l'application.

LE MODÈLE DES RÈGLES TECHNIQUES

Les règles techniques permettent d'une part de décrire comment créer et/ou modifier les dossiers et fichiers contenus dans les champs *ToCreate* et *ToModify* des propriétés techniques et d'autre part de décrire le contenu des listes d'instructions qui seront fournies à l'enseignant et/ou l'apprenant pour paramétrer le logiciel si la modification des fichiers de configuration ne suffit pas ou n'est pas possible.

Cette description contient le format des noms de fichiers ou de dossiers à créer, ainsi que la place dans les fichiers de chacun des paramètres décrits dans le modèle pédagogique de l'EIAH. Pour chaque paramètre, les règles techniques permettent de plus d'indiquer comment convertir la valeur du paramètre pour l'intégrer au fichier. Cette conversion peut être neutre, dans ce cas la valeur du paramètre sera mise telle quelle dans le fichier, ou calculée à l'aide de formules de conversion utilisant les branchements conditionnels IF THEN ELSE ou CASE OF.

C. PROCESSUS AKASI

Nous venons de présenter les principes du méta-modèle AKEPI permettant d'acquérir les connaissances nécessaires à la personnalisation des EIAH. Nous indiquons ici comment ce méta-modèle peut être utilisé pour créer un modèle propre à un EIAH donné en présentant le processus AKASI (Acquisition of Knowledge for A Specific ILE) que nous lui associons.

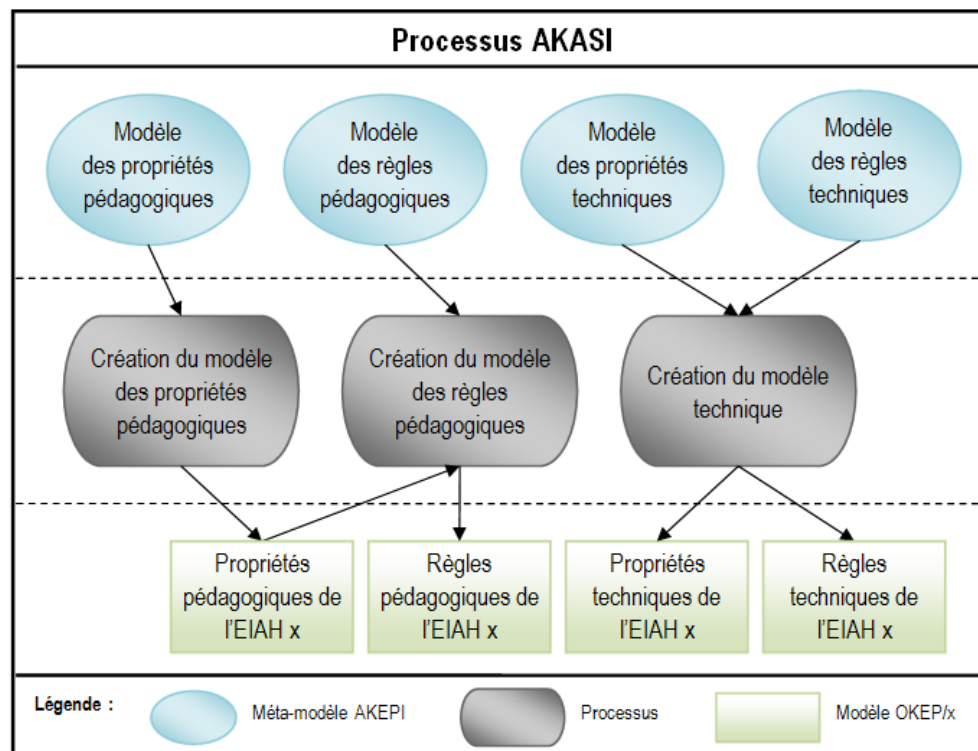


Figure 6-53: Processus AKASI utilisant le méta-modèle AKEPI pour créer un modèle OKEP/x propre à l'EIAH x.

Le processus AKASI permettant la définition des connaissances propres à un EIAH pour créer un modèle OKEP/x se découpe en trois étapes (cf. Figure 6-53). Ces connaissances sont fournies par un expert ou par le concepteur de l'EIAH x.

La **création du modèle des propriétés pédagogiques** consiste à acquérir, à l'aide d'une interface basée sur le méta-modèle AKEPI, les propriétés pédagogiques d'un EIAH x. Ces propriétés regroupent l'ensemble des paramètres influant sur la personnalisation des activités

contenues dans l'EIAH, leur organisation pour former des séquences de travail mais également l'ensemble des paramètres impactant sur les fonctionnalités, les rétroactions ou l'interface de l'EIAH.

La **création du modèle des règles pédagogiques** consiste à acquérir les règles de gestion des propriétés pédagogiques de l'EIAH x. Ces règles respectent le format des règles pédagogiques défini dans le méta-modèle AKEPI.

La **création du modèle technique** consiste à acquérir les propriétés et règles techniques relatives à l'EIAH x. Les propriétés permettent d'indiquer le dossier dans lequel se trouve l'application, les éventuels générateurs d'activités, la description des menus de l'application, ainsi que la liste des fichiers à créer pour personnaliser le logiciel. Les règles permettent de créer un convertisseur des contraintes faites par les enseignants en un fichier de configuration de l'EIAH ou une liste d'instructions à suivre sous forme d'un fichier.

D. EXEMPLES DE MODÈLES OKEP/X

Le processus AKASI permet à un expert de créer, pour chaque EIAH x, un modèle OKEP/x (Operational Knowledge Enabling Personalization of the ILE x). Les modèles OKEP/x respectent la structure définie dans le méta-modèle AKEPI présenté dans la section 6.4.3.b page 160. Un modèle OKEP/x peut être utilisé par le processus OPIKSI, présenté dans la section suivante, pour générer une interface permettant aux enseignants de définir des contraintes de configuration pour l'EIAH x.

Dans la section 6.4.3.a page 154, nous avons présenté trois exemples d'EIAH. La configuration des deux premiers, AMBRE-ADD et TRI, se fait par l'intermédiaire de fichiers de configuration alors que celle du dernier, ABALECT, se fait directement via l'interface utilisée par l'apprenant. En utilisant l'approche GEPPETO_s, la personnalisation des EIAH sera proposée pour les deux premiers sous forme de fichiers de configuration modifiés et dans le troisième cas, sous forme d'un fichier contenant les instructions à suivre par les apprenants. Nous présentons maintenant les modèles OKEP/ABALECT et OKEP/AMBRE-ADD pour illustrer la diversité des modèles OKEP/x pouvant être créés selon la manière de personnaliser les EIAH et selon ce qui peut être personnalisé dans les EIAH (les activités, les séquences d'activités, les fonctionnalités, les rétroactions et l'interface).

D.1. LE MODÈLE OKEP/ ABALECT

Dans le logiciel ABALECT, un enseignant doit, pour faire travailler ses élèves, choisir un texte puis indiquer le type d'exercices à effectuer (cf. Figure 6-38). Le modèle OKEP/ABALECT contient donc, dans sa description pédagogique, les propriétés décrivant une activité et la façon de combiner ces activités pour créer des séquences de travail (cf. Figure 6-54), ainsi que les règles permettant de gérer ces propriétés (cf. Figure 6-55). Le logiciel ne permet de modifier ni son interface, ni ses fonctionnalités, ni ses rétroactions.

```

<PedagogicalProperties>
- <PedagogicalContent>
- <TypeOfActivity>
  <Name>Exercice</Name> ①
  - <Parameters>
    - <Parameter ID="P001">
      <Name>Niveau</Name>
      <AssociatedCompetence>Classe de l'élève</AssociatedCompetence> ②
      - <ScaleList>
        <Variable>false</Variable>
        <Ordered>false</Ordered>
        <MultipleSelection>false</MultipleSelection>
        - <Value>
          <Name>CP</Name>
          </Value>
        - <Value>
          <Name>CE1</Name>
          </Value>
        - <Value>
          <Name>CE2</Name>
          </Value>
        - <Value>
          <Name>CM1</Name>
          </Value>
        + <Value>
          </ScaleList>
        </Parameter>
      - <Parameter ID="P002"> ③
        <Name>Titre du texte</Name>
        - <ScaleList>
          <Variable>false</Variable>
          <Ordered>false</Ordered>
          <MultipleSelection>false</MultipleSelection>
          - <Value>
            <Name>L'anniversaire</Name>
            </Value>
          - <Value>
            <Name>L'arrivée du loup</Name>
            </Value>
          ...
          - <Value>
            <Name>Une maison pour l'hiver</Name>
            </Value>
          </ScaleList>
        </Parameter>
    - <Parameter ID="P003"> ④
      <Name>Type d'exercice</Name>
      - <ScaleList>
        <Variable>false</Variable>
        <Ordered>false</Ordered>
        <MultipleSelection>false</MultipleSelection>
        - <Value>
          <Name>Comprendre le texte</Name>
          </Value>
        - <Value>
          <Name>Ordre</Name>
          <AssociatedCompetence>cohérence</AssociatedCompetence>
          </Value>
        - <Value>
          <Name>Vocabulaire</Name>
          </Value>
        - <Value>
          <Name>Mémoire / Rapidité / Destination</Name>
          </Value>
        - <Value>
          <Name>Orthographe</Name>
          </Value>
        </ScaleList>
      </Parameter>
    - <Parameter ID="P004"> ⑤
      <Name>Nom de l'exercice</Name> ⑥
      - <ScaleList>
        <Variable>false</Variable> ⑦
        <Ordered>false</Ordered>
        <MultipleSelection>true</MultipleSelection>
        - <Value>
          <Name>Questionnaire 1</Name>
          </Value>
        ...
        - <Value>
          <Name>Ordre alphabétique</Name>
          </Value>
        </ScaleList>
      </Parameter>
    </Parameters>
  </TypeOfActivity>
</PedagogicalContent>
- <PedagogicalOrganization>
  - <Sequence>
    - <TypeOfContent>
      <TypeOfActivity>A001</TypeOfActivity>
      <MinNumberOfActivities>1</MinNumberOfActivities>
      <MaxNumberOfActivities>100</MaxNumberOfActivities> ⑧
      <Duplication>false</Duplication>
    </TypeOfContent>
  </Sequence>
  <OrderedSequence>true</OrderedSequence> ⑨
</PedagogicalOrganization>
</PedagogicalProperties>

```

Figure 6-54 : Extrait des propriétés pédagogiques du modèle OKEP/ABALECT.

Une activité de type « Exercice » (cf. ① sur la Figure 6-54) est décrite selon quatre paramètres. Le premier indique le niveau scolaire pour lequel le texte est destiné (cf. ② sur la Figure 6-54), le deuxième indique le nom de chaque texte (cf. ③ sur la Figure 6-54), le troisième le type de l'exercice (cf. ④ sur la Figure 6-54) et le quatrième le nom de l'exercice (cf. ⑤ sur la Figure 6-54). Tous ces paramètres sont décrits grâce à des listes énumérées non ordonnées possédant un nombre fixe d'éléments (cf. ⑥ sur la Figure 6-54). Pour les trois premiers paramètres, un enseignant ne pourra choisir qu'une seule valeur alors que pour le quatrième, il pourra choisir plusieurs valeurs (cf. ⑦ sur la Figure 6-54). Cette définition de l'activité de type « Exercice » revient à dire qu'un enseignant pourra contraindre la sélection de l'activité en choisissant un niveau pour le texte et/ou un nom de texte et/ou un type d'exercices et/ou un ou plusieurs nom d'exercices.

Une séquence de travail d'ABALECT est constituée de une à cent activités de type « Exercice » (cf. ⑧ sur la Figure 6-54) dans un ordre donné (cf. ⑨ sur la Figure 6-54).

```

<PedagogicalRules>
- <Rule ID="R1">
- <Condition>
- <Parameter ID="P001" TypeOfCondition="Value">
  <Value>CP</Value>
</Parameter>
</Condition>
- <Conclusion>
- <Parameter ID="P002" TypeOfConclusion="Domain">
  <Value>L'anniversaire</Value>
  <Value>L'arrivée du loup</Value>
  <Value>L'inondation</Value>
  <Value>Où est mon chat</Value>
  <Value>Une bonne pêche</Value>
  <Value>Une maladie formidable</Value>
</Parameter>
</Conclusion>
</Rule>
+ <Rule ID="R2">
+ <Rule ID="R3">
+ <Rule ID="R4">
+ <Rule ID="R5">
- <Rule ID="R6">
- <Condition>
- <Parameter ID="P003" TypeOfCondition="Value">
  <Value>Comprendre le texte</Value>
</Parameter>
</Condition>
- <Conclusion>
- <Parameter ID="P004" TypeOfConclusion="Domain">
  <Value>Questionnaire</Value>
  <Value>Questionnaire 2</Value>
  <Value>Texte masqué</Value>
  <Value>Closures</Value>
</Parameter>
</Conclusion>
</Rule>
+ <Rule ID="R7">
+ <Rule ID="R8">
+ <Rule ID="R9">
+ <Rule ID="R10">
</PedagogicalRules>

```

Figure 6-55 : Extrait des règles pédagogiques du modèle OKEP/ABALECT.

Les règles pédagogiques (cf. Figure 6-55) permettent de gérer les quatre propriétés relatives aux activités. Ainsi les règles R1 à R5 permettent d'indiquer quel texte est destiné à quel niveau. Par exemple, la règle R1 de la Figure 6-55 traduit la Figure 6-39 en indiquant les textes destinés aux élèves de CP. Les règles R6 à R10 permettent d'indiquer quel exercice (questionnaire, texte masqué, etc.) appartient à quel type d'exercices (comprendre le texte, ordre, etc.). Ces règles traduisent l'interface présentée sur la Figure 6-38.

Pour traduire les contraintes exprimées par les enseignants sur les propriétés pédagogiques du modèle OKEP/ABALECT sous forme d'un fichier de personnalisation (les instructions à suivre par l'apprenant dans le cas présent), l'expert définit une description technique du logiciel. Cette définition contient un ensemble de propriétés (cf. Figure 6-56) et de règles (cf. Figure 6-57).

Les propriétés techniques indiquent le dossier dans lequel se trouve l'application (cf. ① sur la Figure 6-56), le nom de l'exécutable (cf. ② sur la Figure 6-56) et les fichiers à créer pour chaque personnalisation (cf. ③ sur la Figure 6-56). Dans le cas d'ABALECT, il faut créer un fichier « NomEleve.bil » par élève. Ces propriétés contiennent également la description des dossiers et des menus utiles pour créer la liste des instructions à effectuer via l'interface de l'application (cf. ④ sur la Figure 6-56).

```

<TechnicalProperties NameOfFILE="Abalect">
- <Application>
  <Folder>C:\Program Files\Abalect\</Folder> ①
  <Executable>Abalect.exe</Executable> ②
</Application>
- <ToCreate>
  <File ID="F001">NomEleve.bil</File> ③
</ToCreate>
- <Folders>
  <Folder>CP\</Folder>
  <Folder>CE1\</Folder>
  <Folder>CE2\</Folder>
  <Folder>CM1\</Folder>
  <Folder>CM2\</Folder>
</Folders>
- <Menus>
  <Item Level="1" Title="Enseignant" />
  <Item Level="1" Title="Elève" />
  <Item Level="1" Title="Quitter" />
</Menus>
</TechnicalProperties>

```

Figure 6-56 : Propriétés techniques du modèle OKEP/ ABALECT.

```

<TechnicalRules>
- <FileCreation ID="F001">
  - <FileName>
    <Var>EleveEprofilea</Var> ①
    <Text>.bil</Text>
  </FileName>
  <FileContent Empty="true" />
</FileCreation>
- <PedagogicalContent ID="A001"> ②
  <Mode>Interface</Mode>
  <XSL>exercice.xsl</XSL>
  - <RulesXSL>
    <Text>Pour le texte "</Text>
    <Var>P002</Var>
    <Text>",</Text> faites les exercices suivants :</Text>
    <Return />
    <Var Separator="#32#,">P004</Var>
    <Text>.</Text>
    <Return />
  </RulesXSL>
</PedagogicalContent>
- <PedagogicalOrganization> ③
  <mode>Interface</mode>
  <XSL>sequence.xsl</XSL>
  - <RulesXSL>
    <Text>Selectionner le menu Elève.</Text>
    <Return />
    <Text>Choisissez votre nom dans la liste.</Text>
    <Return />
    <PedagogicalContent ID="A001" min="1" max="100" />
    <Return />
  </RulesXSL>
</PedagogicalOrganization>
</TechnicalRules>

```

Figure 6-57 : Règles techniques du modèle OKEP/ ABALECT

Les règles techniques indiquent comment créer le fichier « NomEleve.bil ». Il faut pour cela créer un fichier vide dont le nom sera composé du nom de l'élève et de l'extension « .bil » (cf. ① sur la Figure 6-57). Elles indiquent également comment construire les feuilles de style nécessaires à la création des fichiers de personnalisation. Pour ABALECT, il faut créer une feuille de personnalisation par élève en utilisant la feuille de style « sequence.xsl » (cf. ③ sur la Figure 6-57), qui utilise elle-même la feuille de style « exercice.xsl » (cf. ② sur la Figure 6-57). Ces feuilles de style sont fournies en Annexe I page 349. Pour créer la feuille de style correspondant à la séquence (cf. ③ sur la Figure 6-57), il faut écrire le texte contenu dans les balises <Text>, puis intégrer pour chaque élève autant d'activités de type A001 que nécessaire. Ces exercices sont intégrés en utilisant la feuille de style de l'activité A001 (cf. ② sur la Figure 6-57) qui indique qu'il faut écrire les textes compris entre les balises <Text> et remplacer les variables P002 et P004 par les valeurs des paramètres correspondants.

Le modèle OKEP/ ABALECT dans son intégralité est fourni en Annexe I page 349.

D.2. LE MODÈLE OKEP/ AMBRE-ADD

Revenons à présent sur le premier EIAH de l'exemple présenté dans la section 6.4.3.a page 154 : le logiciel AMBRE-ADD [Nogry et al. 2004]. Dans ce logiciel, il est possible de paramétrer la création des problèmes, de créer des séquences avec les problèmes créés ou avec ceux contenus par défaut dans le logiciel, et de modifier l'interface et l'accès aux fonctionnalités (aide, diagnostic...).

La Figure 6-58 est un extrait des propriétés pédagogiques du modèle OKEP/ AMBRE-ADD montrant les trois premiers paramètres intervenant lors de la création de problèmes. Ces paramètres influent sur le choix du schéma permettant de résoudre le problème (cf. a sur la Figure 6-29). Ainsi pour choisir un schéma, l'enseignant doit spécifier tout d'abord la classe du

problème (cf. ① sur la Figure 6-58). Ce paramètre pourra prendre ses valeurs dans la liste énumérée, *ScaleList*, « Réunion, Changement, Comparaison ». Cette liste a une taille fixe, c'est-à-dire que l'enseignant ne pourra pas choisir une autre valeur que les trois contenues dans la liste. Le second paramètre permet d'indiquer la place de l'inconnue sur le schéma (cf. ② sur la Figure 6-58). Ce paramètre pourra prendre ses valeurs dans la liste énumérée « Résultat, Opérande, Opérateur, Différence, Max des deux, Min des deux ». Certaines valeurs de la liste énumérée sont complétées par les compétences qui leur sont associées. Ainsi, si l'enseignant choisit de placer l'inconnue au niveau du résultat, la compétence associée lui indique que l'apprenant travaillera sur une addition, s'il choisit de placer l'inconnue au niveau de l'opérande, l'apprenant travaillera sur une opération à trou, etc. Le troisième paramètre permet d'indiquer l'opérateur souhaité (cf. ③ sur la Figure 6-58). Ce paramètre pourra prendre ses valeurs dans la liste énumérée « Ajouter, Retrancher, Inconnu ». Ces trois paramètres sont tous associés à la catégorie « Structure ». Cette catégorie sera utilisée lors de la génération de l'interface de configuration proposée à l'enseignant. Elle servira à regrouper les paramètres ayant des liens entre eux.

```

<PedagogicalProperties NameOfILE="AMBRE-add">
- <PedagogicalContent>
- <TypeOfActivity ID="A001">
  <Name>Problème</Name>
  - <Parameters>
    - <Parameter ID="P001"> ①
      <Name>Classe du problème</Name>
      <AssociatedCompetence />
    - <ScaleList>
      <Variable>>false</Variable>
      <Ordered>>false</Ordered>
      <MultipleSelection>>false</MultipleSelection>
      - <Value>
        <Name>Réunion</Name>
        <AssociatedCompetence />
      </Value>
      - <Value>
        <Name>Changement</Name>
        <AssociatedCompetence />
      </Value>
      - <Value>
        <Name>Comparaison</Name>
        <AssociatedCompetence />
      </Value>
    </ScaleList>
    - <Category>
      <Name>Structure</Name>
    </Category>
  </Parameters>
  - <Parameter ID="P002"> ②
    <Name>Place de l'inconnue</Name>
    <AssociatedCompetence />
    - <ScaleList>
      <Variable>>false</Variable>
      <Ordered>>false</Ordered>
      <MultipleSelection>>false</MultipleSelection>
      - <Value>
        <Name>Résultat</Name>
        <AssociatedCompetence />
      </Value>
      - <Value>
        <Name>Opérande</Name>
        <AssociatedCompetence />
      </Value>
      - <Value>
        <Name>Opérateur</Name>
        <AssociatedCompetence />
      </Value>
    </ScaleList>
  </Parameter>
  - <Value>
    <Name>Différence</Name>
    <AssociatedCompetence />
  </Value>
  - <Value>
    <Name>Max des deux</Name>
    <AssociatedCompetence />
  </Value>
  - <Value>
    <Name>Min des deux</Name>
    <AssociatedCompetence />
  </Value>
</ScaleList>
- <Category>
  <Name>Structure</Name>
</Category>
</Parameter>
- <Parameter ID="P003"> ③
  <Name>Opérateur</Name>
  <AssociatedCompetence />
  - <ScaleList>
    <Variable>>false</Variable>
    <Ordered>>false</Ordered>
    <MultipleSelection>>false</MultipleSelection>
    - <Value>
      <Name>Ajouter</Name>
      <AssociatedCompetence />
    </Value>
    - <Value>
      <Name>Retrancher</Name>
      <AssociatedCompetence />
    </Value>
    - <Value>
      <Name>Inconnu</Name>
      <AssociatedCompetence />
    </Value>
  </ScaleList>
  - <Category>
    <Name>Structure</Name>
  </Category>
</Parameter>
</TypeOfActivity>
</PedagogicalContent>
</PedagogicalProperties>

```

Figure 6-58 : Extrait des paramètres de création d'un problème avec le générateur GENAMBRE.

La description pédagogique du modèle AMBRE-ADD contient 22 autres paramètres permettant de générer une activité de type problème. À ces paramètres est associé un ensemble de règles

dont un extrait est fourni Figure 6-59. La règle R1 contraint le domaine de valeur du paramètre « Place de l'inconnue » (cf. ② sur la Figure 6-58) en fonction de la valeur affectée au paramètre « Classe du problème » (cf. ① sur la Figure 6-58). Ainsi, si l'enseignant choisit la valeur « Réunion » pour le paramètre « Classe du problème », il ne pourra choisir pour le paramètre « Place de l'inconnue » que les valeurs « Résultat » ou « Opérande ». La règle R2 contraint également la valeur du paramètre « Place de l'inconnue » en fonction de la valeur affectée au paramètre « Classe du problème ».

```

<PedagogicalRules>
- <Rule ID="R1">
- <Condition>
- <Parameter ID="P001" TypeOfCondition="Value">
  <Value>Réunion</Value>
</Parameter>
</Condition>
- <Conclusion>
- <Parameter ID="P002" TypeOfConclusion="Domain">
  <Value>Résultat</Value>
  <Value>Opérande</Value>
</Parameter>
</Conclusion>
</Rule>
- <Rule ID="R2">
- <Condition>
- <Parameter ID="P001" TypeOfCondition="Value">
  <Value>Changement</Value>
</Parameter>
  <Parameter ID="P003" TypeOfCondition="NotDefined" />
</Condition>
- <Conclusion>
- <Parameter ID="P002" TypeOfConclusion="Value">
  <Value>Opérateur</Value>
</Parameter>
</Conclusion>
</Rule>
</PedagogicalRules>

```

Figure 6-59 : Extrait des règles pédagogiques du modèle OKEP/AMBRE-ADD.

La description pédagogique du modèle OKEP/AMBRE-ADD contient les propriétés et les règles permettant de générer une interface pour configurer un problème, mais également des propriétés et des règles que nous ne présentons pas ici, permettant de créer des séquences d'activités et de paramétrer l'interface et les fonctionnalités du logiciel.

Le modèle OKEP/AMBRE-ADD contient en plus de la description pédagogique, une description technique permettant de créer les fichiers de configuration du logiciel. Cette description permet de produire, entre autres, une feuille de style dont un extrait est fourni sur la Figure 6-60. Cet extrait montre la feuille de style permettant de créer le fichier de configuration du générateur GENAMBRE (cf. Figure 6-30). Les valeurs des trois paramètres de la Figure 6-58 sont ainsi combinées et traduites pour être interprétées par le générateur (cf. ① sur la Figure 6-60).

```

<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
- <xsl:template match="/">
  <xsl:text>consult('..\..\Commun\themes_fr').</xsl:text>
  <br />
  <xsl:text>appel_generation([[</xsl:text>
  <xsl:call-template name="ClassePb" />
  <xsl:text>','</xsl:text>
  <xsl:call-template name="TermeCompa" />
  <xsl:text>'],'</xsl:text>
  <xsl:call-template name="Theme" />
  <xsl:text>'],'</xsl:text>
  <xsl:call-template name="Objets" />
  <xsl:text>'],'</xsl:text>
  <xsl:call-template name="Acteurs" />
  <xsl:text>'],'</xsl:text>
  <xsl:call-template name="UneSorteObjet" />
  <xsl:text>','</xsl:text>
  <xsl:call-template name="PlusieursSortesObjets" />
  <xsl:text>','</xsl:text>
  <xsl:call-template name="AffecterObjets" />
  <xsl:text>','</xsl:text>
  <xsl:call-template name="AffecterActeurs" />
  ....
  <xsl:text>'],'aperçu).</xsl:text>
</xsl:template>
- <xsl:template name="ClassePb">
- <xsl:for-each select="/params/param">
- <xsl:if test="@prop="classe de probleme">
  - <xsl:choose>
  - <xsl:when test="@valeur="reunion">
  - <xsl:for-each select="/params/param">
  - <xsl:if test="@prop="place de l inconnue">
    - <xsl:choose>
    - <xsl:when test="@valeur="resultat">
      <xsl:text>c_reunion_res</xsl:text>
    </xsl:when>
    - <xsl:when test="@valeur="operande">
      <xsl:text>c_reunion_operande</xsl:text>
    </xsl:when>
    - <xsl:otherwise>
      <xsl:text>c_reunion</xsl:text>
    </xsl:otherwise>
    </xsl:choose>
  </xsl:if>
  </xsl:for-each>
  </xsl:when>
  ....

```

Figure 6-60 : Extrait de la feuille de style créée à partir de la description technique du modèle OKEP/AMBRE-ADD.

E. PROCESSUS OPIKSI

Dans l'approche GEPPETO, le modèle d'activités est utilisé pour créer des activités en passant par une étape de définition de contraintes sur activités (cf. Figure 6-1, page 118). Il est donc nécessaire de disposer d'un processus utilisant d'une part le modèle d'activités pour permettre à l'agent de la personnalisation de définir ses contraintes et d'autre part de se servir de ces contraintes pour générer des activités.

Dans l'approche GEPPETO_s, ce processus est le processus OPIKSI (Outsource Personalisation of an ILE by Knowledge Specific to ILE) qui utilise le modèle OKEP/x pour personnaliser un EIAH x. Ce processus se décompose en quatre étapes (cf. Figure 6-61).

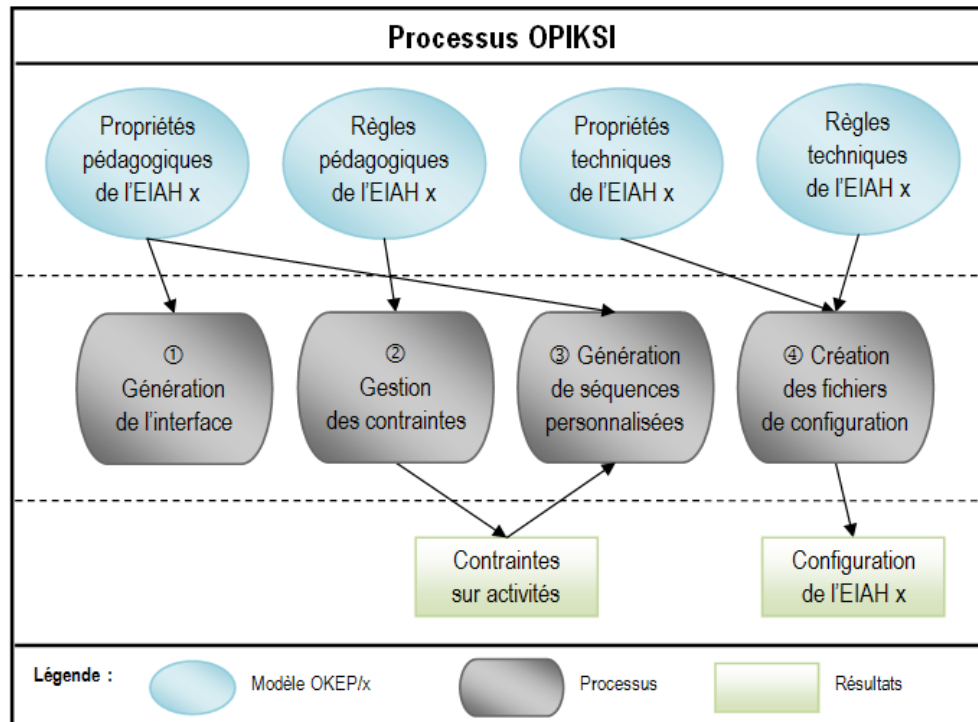


Figure 6-61 : Processus OPIKSI utilisant le modèle OKEP/x pour personnaliser l'EIAH x.

Dans la première étape (cf. ① sur la Figure 6-61), une interface est générée automatiquement à partir des propriétés pédagogiques du modèle OKEP/x.

Dans la seconde étape (cf. ② sur la Figure 6-61), l'enseignant utilise cette interface pour définir des contraintes sur la génération d'activités ou la sélection d'activités déjà existantes, ainsi que sur le choix des paramètres agissant sur les fonctionnalités et l'interface de l'EIAH x. Les règles pédagogiques permettent de gérer dynamiquement l'interface proposée à l'enseignant.

Dans la troisième étape (cf. ③ sur la Figure 6-61), à partir des contraintes spécifiées par l'enseignant et des propriétés pédagogiques du modèle OKEP/x, le système génère des séquences d'activités adaptées aux apprenants et définit les paramètres de personnalisation de l'environnement.

Dans la dernière étape (cf. ④ sur la Figure 6-61), le système utilise la description technique du modèle OKEP/x pour créer les fichiers de configuration propres à l'EIAH x ou pour créer une liste d'instructions à suivre pour utiliser l'interface de configuration de l'EIAH.

6.4.4. BILAN SUR L'ADAPTATION D'EIAH

Dans la section 6.2, nous avons présenté les principes de l'approche GEPPETO. Cette approche est résumée sur la première ligne de la Figure 6-62. Elle contient un méta-modèle d'activités qui est utilisé par un processus de création de modèles pour définir des modèles d'activités papier ou logicielles. Chaque modèle d'activités peut être utilisé par un processus de définition de contraintes pour créer des contraintes sur activités. Ce processus a deux étapes : la proposition d'une interface, adaptée au modèle de l'activité, permettant de contraindre les activités et l'enregistrement de ces contraintes. Un ensemble de contraintes sur activités peut ensuite être utilisé par un processus de génération d'activités pour créer des activités pédagogiques, qu'elles soient papier ou logicielles. Ce processus a également deux étapes : l'utilisation des contraintes pour créer une activité et la mise en page de cette activité.

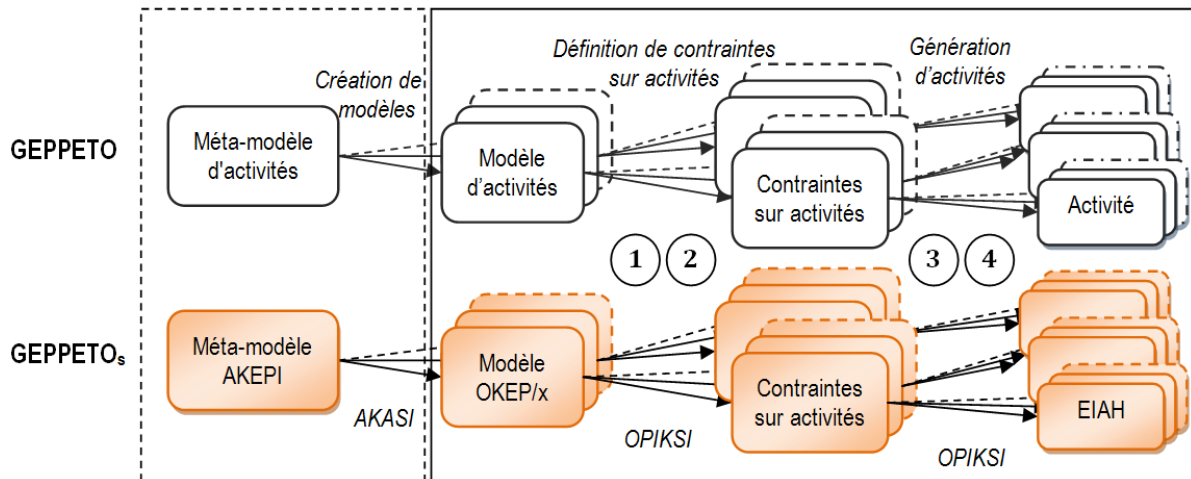


Figure 6-62 : Parallèle entre l'approche GEPPETO et sa déclinaison en GEPPETO_s.

Dans la section 6.4.3, nous avons indiqué comment utiliser l'approche GEPPETO pour personnaliser les activités logicielles, et plus généralement les EIAH. Cette déclinaison de GEPPETO pour les EIAH est appelée GEPPETO_s. Nous soulignons à présent le lien entre les modèles et les processus de l'approche générale et ceux de sa déclinaison.

Le méta-modèle AKEPI correspond au méta-modèle d'activités pour les activités logicielles et plus généralement les EIAH. En effet, personnaliser une activité au sein d'un logiciel requiert de personnaliser aussi l'environnement (interface, fonctionnalités, etc.) dans lequel cette activité se déroule.

À partir du méta-modèle AKEPI, il est possible de définir un modèle d'activités pour chaque EIAH à personnaliser. Ce modèle d'activités pour les EIAH est le modèle OKEP/x, x étant le nom de l'EIAH. Le processus de définition des modèles, le processus AKASI dans ce cas, permet à un expert de définir les connaissances pour un EIAH donné.

Les processus de définition de contraintes et de génération d'activités sont intégrés dans le processus OPIKSI. Dans ce processus, on retrouve bien les quatre étapes composant les deux processus. La première étape correspond à la proposition d'une interface adaptée au modèle de l'activité (cf. ① sur les Figures 6-61 et 6-62). La seconde étape permet de définir des contraintes sur activités, qui correspondent aux contraintes permettant de personnaliser l'EIAH (cf. ② sur les Figures 6-61 et 6-62). La troisième étape crée cette personnalisation en générant ou choisissant des activités dans l'EIAH, en combinant ces activités pour créer des séquences de travail et en définissant les paramètres de l'environnement (interface, fonctionnalités et rétroactions) (cf. ③ sur les Figures 6-61 et 6-62). Enfin, la quatrième étape soit crée ou modifie les fichiers de configuration de l'EIAH, soit crée une liste d'instructions à suivre pour personnaliser l'EIAH via l'interface de configuration de celui-ci ou directement via l'interface du logiciel (cf. ④ sur les Figures 6-61 et 6-62). Ces fichiers, de configuration ou d'instructions, permettent bien de personnaliser des activités pédagogiques.

Les propositions que nous avons faites pour les EIAH respectent donc bien l'approche GEPPETO que nous avons définie.

6.5. CE QU'IL FAUT RETENIR

Dans cette section, nous faisons une synthèse de l'approche GEPPETO et ses deux déclinaisons permettant de personnaliser les activités papier et logicielles. Nous nous appuyons pour cela sur la Figure 6-63.

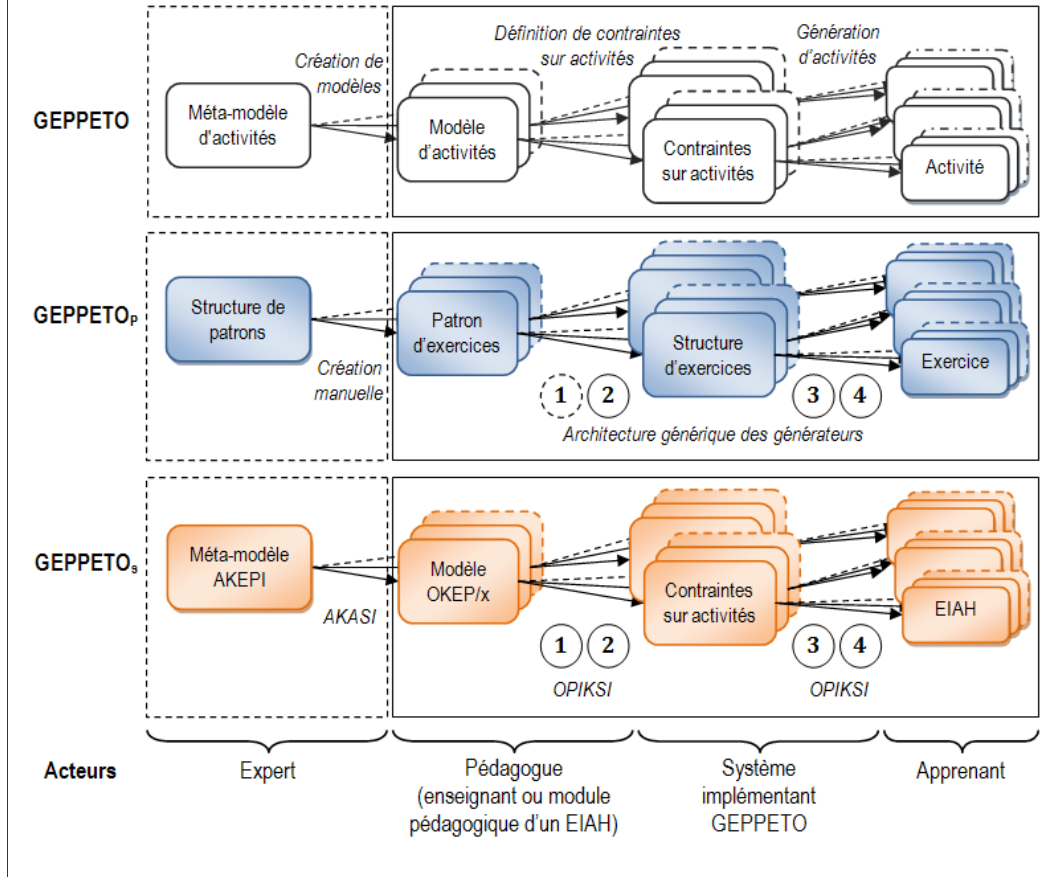


Figure 6-63 : Synthèse de l'approche GEPPETO et ses deux déclinaisons permettant de personnaliser les activités papier et logicielles.

GEPPETO est une approche proposant des modèles et des processus permettant d'adapter les activités pédagogiques proposées aux apprenants en fonction des objectifs d'apprentissage de l'agent mettant en place la personnalisation de l'apprentissage (cf. section 6.2). Cette approche a été déclinée en GEPPETO_p pour la personnalisation des activités papier (cf. section 6.3.3) et en GEPPETO_s pour la personnalisation des EIAH (cf. 6.4.3).

Le méta-modèle de GEPPETO est instancié pour donner une structure de patrons dans GEPPETO_p et pour donner le méta-modèle AKEPI dans GEPPETO_s. La structure de patrons des activités papier est donc le pendant du méta-modèle AKEPI pour les activités logicielles.

Le processus de création des modèles d'activités de GEPPETO est manuel dans GEPPETO_p alors qu'il correspond au processus AKASI dans GEPPETO_s.

Les modèles d'activités de GEPPETO se déclinent en patrons d'exercices dans GEPPETO_P et en modèles OKEP/x dans GEPPETO_S.

Les deux processus GEPPETO utilisant ces modèles d'activités pour adapter les activités sont regroupés dans les générateurs d'exercices de GEPPETO_P et dans le processus OPIKSI de GEPPETO_S. Dans les deux cas, le passage du modèle d'activités à l'activité adaptée se fait en quatre étapes.

La première consiste à proposer une interface permettant de définir des contraintes sur la création des activités. Dans GEPPETO_P, cette interface est associée aux patrons d'exercices et la sélection de l'interface est donc associée à la sélection du générateur correspondant au patron d'exercices. Cette première étape du processus de génération se fait donc en dehors du générateur. À l'inverse, dans GEPPETO_S, l'interface est générée dynamiquement en fonction du modèle OKEP/x lors du processus de génération.

La seconde étape du processus de génération consiste à enregistrer les contraintes sur les activités. La troisième étape utilise les contraintes sur activités pour créer ou sélectionner l'activité adéquate et la dernière étape crée les documents (feuilles d'exercices à imprimer, fichiers de configuration ou listes d'instructions à suivre) permettant de proposer les activités à chaque apprenant. Ces étapes sont équivalentes dans les deux déclinaisons de GEPPETO.

CHAPITRE 7. PROCESSUS DE CRÉATION DE
SÉQUENCES PERSONNALISÉES
D'ACTIVITÉS

PLAN DU CHAPITRE

7.1. Introduction.....	181
7.2. Principe d'exploitation du modèle PERSUA2.....	181
7.3. Architecture du processus de création de séquences personnalisées d'activités	187
7.3.1. Filtrage des règles d'affectation.....	188
7.3.2. Création des listes d'activités à générer.....	189
7.3.3. Génération des activités.....	189
7.3.4. Création des séquences de travail personnalisées.....	190
7.4. Ce qu'il faut retenir	191

PUBLICATION RELATIVE À CE CHAPITRE

[Lefevre et al. 2009c]

Lefevre, M., Jean-Daubias, S., Guin, N. « Personnaliser des séquences de travail à partir de profils d'apprenants ». Poster, 4ème Conférence en Environnements Informatiques pour l'Apprentissage Humain (EIAH'2009), Le Mans, France, 23-26 juin 2009.

7.1. INTRODUCTION

Le modèle PERSUA2 présenté dans la section 4.3 permet à chaque agent de la personnalisation de choisir comment affecter des activités à un apprenant en fonction des buts pédagogiques de l'agent. Pour exploiter ce modèle, nous proposons un processus qui permet d'interpréter le modèle de personnalisation en fonction des profils des apprenants de manière à créer des séquences de travail personnalisées, sur support papier ou logiciel [Lefevre et al. 2009c].

Dans ce chapitre nous illustrons le principe d'exploitation du modèle PERSUA2 à partir d'exemples, puis nous détaillons le processus d'exploitation du modèle PERSUA2.

7.2. PRINCIPE D'EXPLOITATION DU MODÈLE PERSUA2

Pour comprendre le mécanisme permettant d'exploiter le modèle de personnalisation, reprenons le scénario 1 présenté page 27. Dans ce scénario, Alphonse, enseignant de CE2, souhaite s'appuyer sur les informations dont il dispose sur les connaissances de ses élèves pour proposer des séances de remédiation adaptées à chacun des six élèves de son groupe de soutien.

Lors des séances de soutien, il se trouve dans une salle disposant d'une partie classique, avec un tableau et des tables, et d'une partie informatique, avec des ordinateurs, où est installé un logiciel permettant de travailler sur les tables de multiplication. Les activités proposées aux élèves peuvent donc être des activités papier ou logicielles.

Pour préparer ces séances, Alphonse a créé, pour chacun des six élèves, un profil contenant des informations sur leurs compétences en français et en mathématiques. La Figure 7-1 montre les profils de Jules et de Mathilde.

Jules	Commentaire	Souvent absent	Mathilde	Commentaire	-
	Français			Français	
	Orthographe	Maîtrisé		Orthographe	Maîtrisé
	Conjugaison			Conjugaison	
	Passé simple	Non maîtrisé		Passé simple	Maîtrisé
	Présent			Présent	
	Verbes du 1 ^{er} groupe	Maîtrisé		Verbes du 1 ^{er} groupe	Maîtrisé
	Verbes du 2 ^{ème} groupe	Non maîtrisé		Verbes du 2 ^{ème} groupe	Maîtrisé
	Verbes du 3 ^{ème} groupe	-		Verbes du 3 ^{ème} groupe	Partiellement maîtrisé
	Mathématiques			Mathématiques	
	Addition			Addition	
	Connaître ses tables	10		Connaître ses tables	5
	Savoir résoudre un problème additif	6		Savoir résoudre un problème additif	3
Multiplication	8	Multiplication	2		

Figure 7-1 : Profils de Jules et Mathilde.

Pendant ces séances de soutien, Alphonse veut faire travailler les élèves qui ne maîtrisent pas les compétences de conjugaison sur un exercice papier de type P_B17 et ceux qui les maîtrisent partiellement sur deux exercices papier, un de type P_B18a et un de type P_B18b. Cela revient à regarder la valeur de l'élément « Français - Conjugaison » dans le profil de l'élève, élément qui prend pour valeur la moyenne de ses sous-éléments. L'échelle associée à cette valeur est la liste énumérée « non maîtrisé, partiellement maîtrisé, maîtrisé ». Si la valeur de l'élève est « non maîtrisé » alors l'élève aura un exercice papier de type P_B17, si la valeur est « partiellement maîtrisé » alors l'élève aura sur un exercice papier de type P_B18a et un exercice papier de type P_B18b, enfin si la valeur est « maîtrisé », l'élève n'aura pas d'exercice.

Alphonse souhaite également faire travailler les élèves sur les tables d'addition. Il souhaite que les élèves ne maîtrisant pas cette compétence travaillent sur un exercice papier de type P_C24a, que ceux la maîtrisant partiellement travaillent sur un exercice papier de type P_C24b et que ceux la maîtrisant travaille sur un exercice papier de type P_C24c. Cela revient à regarder la valeur de l'élément « Mathématiques – Addition – Connaître ses tables d'addition » dans le profil de l'élève. Cette valeur est définie grâce à un entier allant de 0 à 10. Si la valeur de l'élève est strictement inférieure à 5 alors l'élève aura un exercice papier de type P_C24a, si la valeur est comprise entre 5 inclus et 8 exclu, alors l'élève aura un exercice papier de type P_C24b sinon l'élève aura un exercice papier de type P_C24c.

Alphonse souhaite enfin faire travailler les élèves sur la multiplication en utilisant le logiciel installé dans la salle informatique où il se trouve. Il souhaite que les élèves ne maîtrisant pas la multiplication fassent une activité de type S_A12 sur le logiciel et que les autres fassent une activité de type S_A75 sur le logiciel. Cela revient à regarder la valeur de l'élément « Mathématiques - Multiplication » dans le profil de l'élève. Cette valeur est définie grâce à un entier allant de 0 à 10. Si la valeur de l'élève est strictement inférieure à 5, il aura une activité logicielle de type S_A12 sur le logiciel de conjugaison, dans le cas contraire il aura une activité logicielle de type S_A75 sur le même logiciel.

Pour la prochaine séance de soutien, Alphonse souhaite faire travailler ses élèves en priorité sur la conjugaison, puis sur les tables de multiplication. Il veut de plus que chacun des élèves fasse entre trois et quatre activités.

Cette stratégie de création de séances de soutien personnalisées peut être formalisée grâce au modèle présenté dans la section 4.3. Cette formalisation est fournie dans la Figure 7-2 et peut être schématisée par la Figure 7-3.

$P_{\text{Alphonse}} = (\text{SP}, \text{CU})$ $\text{SP} = \{(\text{RA}_1, ++), (\text{RA}_2, ++), (\text{RA}_3, -), (\text{RA}_4, -), (\text{RA}_5, -), (\text{RA}_6, +)\}$ $\text{RA}_1 = \{\text{CP}_{\text{E1}}, (\text{P_B17})\}$ $\text{RA}_2 = \{\text{CP}_{\text{E2}}, (\text{P_B18a}, \text{P_B18b})\}$ $\text{RA}_3 = \{\text{CP}_{\text{V(N)1}}, (\text{P_C24a})\}$ $\text{RA}_4 = \{\text{CP}_{\text{V(N)2}}, (\text{P_C24b})\}$ $\text{RA}_5 = \{\text{CP}_{\text{V(N)3}}, (\text{P_C24c})\}$ $\text{RA}_6 = \{\text{CP}_{\text{V(N)4}}, (\text{S_A12}), (\text{S_A75})\}$ $\text{CP}_{\text{E1}} = \{\text{« Français - Conjugaison », (« non maîtrisé, partiellement maîtrisé, maîtrisé »), « non maîtrisé, partiellement maîtrisé, maîtrisé », moyenne, CP}_{\text{V(T)1}}\}$ $\text{CP}_{\text{E2}} = \{\text{« Français - Conjugaison », (« non maîtrisé, partiellement maîtrisé, maîtrisé »), « non maîtrisé, partiellement maîtrisé, maîtrisé », moyenne, CP}_{\text{V(T)2}}\}$ $\text{CP}_{\text{V(T)1}} = \{X, (\text{non maîtrisé})\}$ $\text{CP}_{\text{V(T)2}} = \{X, (\text{partiellement maîtrisé})\}$ $\text{CP}_{\text{V(N)1}} = \{\text{« Mathématiques - Addition - Connaître ses tables d'addition », ((0,true),(5,false))\}$ $\text{CP}_{\text{V(N)2}} = \{\text{« Mathématiques - Addition - Connaître ses tables d'addition », ((5,true),(8,false))\}$ $\text{CP}_{\text{V(N)3}} = \{\text{« Mathématiques - Addition - Connaître ses tables d'addition », ((8,true),(10,true))\}$ $\text{CP}_{\text{V(N)4}} = \{\text{« Mathématiques - Multiplication », ((0,true),(5, false))\}$ $\text{CU} = \{\text{LP}, \text{CC}, \emptyset\}$ $\text{LP} = \{\text{Brice}, \text{Chloé}, \text{Jules}, \text{Oscar}, \text{Mathilde}, \text{Victoria}\}$ $\text{CC} = \{\text{N}_a, \text{indifférent}, \emptyset, \emptyset, \emptyset\}$ $\text{N}_a = [3, 4]$

Figure 7-2 : Exemple d'un modèle de personnalisation.

Stratégie pédagogique				
ID	SI	ALORS	SINON	
RA ₁	Conjugaison = [Non maîtrisé]	Activité papier type P_B17		++
RA ₂	Conjugaison = [Partiellement maîtrisé]	Activité papier type P_B18a et B18b		++
RA ₃	Addition- Connaître ses tables = [0, 5[Activité papier type P_C24a		-
RA ₄	Addition- Connaître ses tables = [5, 8[Activité papier type P_C24a		-
RA ₅	Addition- Connaître ses tables = [8, 10]	Activité papier type P_C24a		-
RA ₆	Multiplication = [0, 5[Activité logicielle type S_A12	Activité logicielle type S_A75	+
Contexte d'utilisation				
Liste des élèves		Brice, Chloé, Jules, Oscar, Mathilde, Victoria		
Nombre d'exercices		Min = 3 ; Max = 4		

Figure 7-3 : Schématisation du modèle de personnalisation de la Figure 7-2.

Pour exploiter ce modèle de personnalisation, une première étape consiste à comparer les valeurs contenues dans les profils des élèves avec les conditions des règles d'affectation définies par l'enseignant afin d'obtenir la liste des règles activées par le profil, ainsi que la liste des contraintes sur les activités concernées pour chaque règle.

Illustrons cette exploitation à l'aide du profil de Jules (cf. Figure 7-1).

Pour interpréter les règles d'affectation RA₁ et RA₂, il faut obtenir la valeur de la composante « Français - Conjugaison » en faisant une moyenne des valeurs de ses sous-composantes. Or Jules

ne possède pas de valeur pour l'élément « Français – Conjugaison – Présent – Verbes du 3^{ème} groupe ». Dans ce cas, les deux règles d'affectation vont être retenues.

Pour interpréter les règles d'affectation RA₃, RA₄ et RA₅, il faut exploiter l'élément « Mathématiques – Addition – Connaître ses tables d'addition ». Jules a 10 pour cet élément. Or 10 appartient à l'intervalle [8, 10], qui est l'intervalle d'application de la règle d'affectation RA₅. On retient donc uniquement la règle RA₅.

Pour interpréter la règle RA₆, il faut exploiter l'élément « Mathématiques – Multiplication ». Jules a 8 pour cet élément. Or 8 n'appartient pas à l'intervalle [0, 5]. La règle d'affectation RA₆ ayant deux listes de contraintes sur activités, une pour les élèves validant la condition de la règle et une pour ceux ne la validant pas. On retient donc la règle RA₆ avec la seconde liste de contraintes sur activités.

La synthèse de cette interprétation des règles d'affectation en fonction des valeurs du profil de Jules est fournie sur la Figure 7-4.

Prenons maintenant le profil de Mathilde (cf. Figure 7-1).

Pour interpréter les règles d'affectation RA₁ et RA₂, il faut obtenir la valeur de la composante « Français - Conjugaison » en faisant une moyenne. On fait tout d'abord la moyenne de l'élément « Français - Conjugaison – Présent » qui a pour résultat « maîtrisé ». Puis on fait la moyenne des éléments « Français - Conjugaison – Présent » et « Français - Conjugaison – Passé simple ». Cette moyenne a pour résultat « maîtrisé ». Mathilde a donc la valeur « maîtrisé » pour l'élément « Français - Conjugaison ». On ne retient donc aucune des deux règles d'affectation pour cette élève.

Pour interpréter les règles d'affectation RA₃, RA₄ et RA₅, il faut exploiter l'élément « Mathématiques – Addition – Connaître ses tables d'addition ». Mathilde a une note de 5 pour cet élément et valide donc la condition de la règle d'affectation RA₄.

Pour interpréter la règle d'affectation RA₆, il faut exploiter l'élément « Mathématiques – Multiplication ». Mathilde a une note 2 pour cet élément. On retient donc la règle RA₆ avec la première liste de contraintes sur activités.

La synthèse de cette interprétation des règles d'affectation en fonction des valeurs du profil de Mathilde est fournie sur la Figure 7-5.

Jules	ID	Contraintes sur activités	⌘
	RA ₁	P_B17	++
	RA ₂	P_B18a et P_B18b	++
	RA ₅	P_C24a	-
	RA ₆	S_A75	+

Figure 7-4 : Liste des règles d'affectation activées par le profil de Jules et des contraintes sur activités associées.

Mathilde	ID	Contraintes sur activités	⌘
	RA ₄	P_C24b	-
	RA ₆	S_A12	+

Figure 7-5 : Liste des règles d'affectation activées par le profil de Mathilde et des contraintes sur activités associées.

À partir des deux listes contenant, pour chaque élève, les règles d'affectation activées par leur profil, ainsi que la liste de contraintes sur activités correspondant à chaque règle (cf. Figure 7-4 et Figure 7-5), il est possible de prendre en compte le contexte d'utilisation représenté dans le modèle de personnalisation pour obtenir la liste définitive des activités à générer pour chaque élève.

Prenons la liste des activités à générer pour Jules (cf. Figure 7-4). Le résultat de l'application des règles d'affectation sur son profil donne cinq activités à générer (P_B17, P_B18a, P_B18b, P_C24b, S_A75). Or le contexte d'utilisation spécifie qu'il en faut au maximum quatre. Il faut donc en supprimer une. C'est à ce moment là que le niveau de priorité de la règle d'affectation entre en compte. On va supprimer de la liste l'activité à générer associée à la règle d'affectation de plus faible niveau de priorité, soit la règle RA₅ et l'activité à générer P_C24b. Nous avons donc maintenant une liste d'activités à générer correspondant à la fois au profil de Jules, à la stratégie pédagogique de l'enseignant et compatible avec le contexte d'utilisation (cf. Figure 7-6).

Prenons maintenant la liste des activités à générer pour Mathilde (cf. Figure 7-5). Il y a pour l'instant deux activités à générer (P_C24b et S_A12), or le contexte d'utilisation spécifie qu'il en faut au minimum trois. Il faut donc en ajouter une. On va ajouter à la liste l'activité à générer associée à la règle d'affectation de plus haute priorité, soit la règle RA₆ et l'activité à générer S_A12. Nous avons donc maintenant une liste d'activités à générer correspondant à la fois au profil de Mathilde, à la stratégie pédagogique de l'enseignant et compatible avec le contexte d'utilisation (cf. Figure 7-7).

Jules	Contraintes sur activités	Occurrence
	P_B17	1
	P_B18a	1
	P_B18b	1
	P_C24a	0
	S_A75	1

Mathilde	Contraintes sur activités	Occurrence
	P_C24b	1
	S_A12	2

Figure 7-6 : Liste des activités à générer pour Jules.

Figure 7-7 : Liste des activités à générer pour Mathilde.

Maintenant que nous avons pour chaque élève la liste des activités à générer, il reste à les générer et à créer à partir des résultats de la génération, les feuilles d'exercices à imprimer et les sessions sur le logiciel de conjugaison. Pour les activités papier, il est nécessaire de posséder des générateurs d'exercices, ainsi que des connaissances permettant de composer ces exercices afin de créer une feuille d'exercices à imprimer. Pour les activités sur le logiciel de conjugaison, il est nécessaire de posséder un modèle de ce logiciel permettant d'une part de générer une activité ou de la choisir dans la base d'activités du système en fonction des contraintes spécifiées par l'enseignant, et d'autre part de créer les fichiers de configuration du logiciel permettant de le paramétrer pour chaque élève. Nous n'expliquons pas en détail ces deux points puisqu'ils ont fait chacun l'objet d'une section spécifique (cf. respectivement section 6.3.3 page 127 et section 6.4.3 page 153).

À la fin de l'étape de génération des activités et de création des séquences de travail pour chaque élève, nous obtenons les résultats suivants (cf. Figure 7-8) :

- Pour Jules :
 - une feuille d'exercices à imprimer contenant trois exercices respectant les contraintes P_B17, P_B18a et P_B18b,
 - un fichier de configuration pour le logiciel de conjugaison permettant de faire une activité respectant les contraintes S_A75.
- Pour Mathilde :
 - une feuille d'exercices à imprimer contenant un exercice respectant les contraintes P_C24b,
 - un fichier de configuration pour le logiciel de conjugaison permettant de faire deux activités respectant toutes les deux les contraintes S_A12. Ces activités ne

possèdent pas le même énoncé, mais possèdent des caractéristiques en commun (par exemple, le nombre de table à connaître, des tables précises comme celle de 9).

- Pour l'enseignant :
 - un bilan contenant un descriptif des activités fournies à chacun de ses élèves, ainsi que la correction des activités contenues dans les feuilles d'exercices papier.

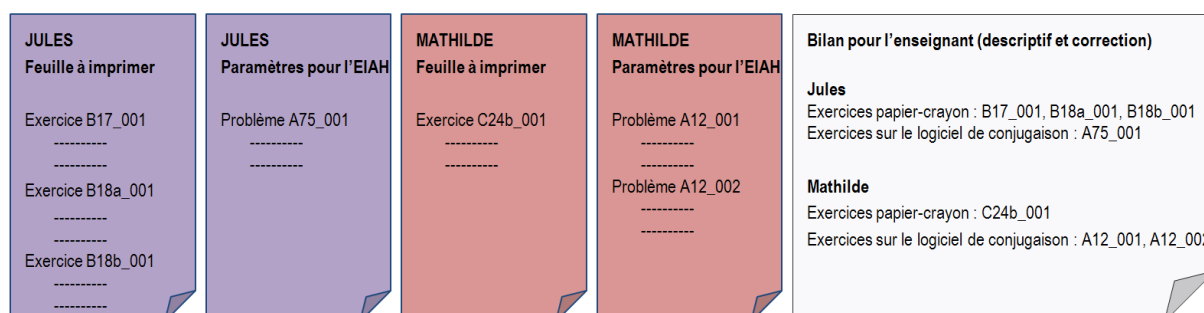


Figure 7-8 : Résultat du processus de création de séquences d'activités personnalisées.

Avant de détailler plus formellement l'architecture du processus de création des séquences d'activités personnalisées, résumons le principe que nous proposons à travers notre modèle de personnalisation et le processus qui lui est associé :

- L'enseignant définit des *règles d'affectation* des activités aux apprenants
 - en contraignant le profil d'apprenant pour en sélectionner une partie, puis en contraignant les valeurs que les profils d'apprenants contiendront pour cette partie (cf. ① sur la Figure 7-9),
 - en contraignant la génération des activités à fournir (cf. ② sur la Figure 7-9).
- L'enseignant hiérarchise ensuite ces règles d'affectation en fonction de leur *niveau de priorité* pour créer sa *stratégie pédagogique* (cf. ③ sur la Figure 7-9).
- L'enseignant définit un *contexte d'utilisation* de sa stratégie pédagogique contenant entre autres la liste des apprenants pour lesquels il souhaite une séquence personnalisée (cf. ④ sur la Figure 7-9).
- Le système filtre les règles d'affectation en fonction du profil de chaque apprenant (cf. ⑤ sur la Figure 7-9).
- Le système filtre les activités à générer en fonction des règles activées par les profils et du contexte d'utilisation (cf. ⑥ sur la Figure 7-9).
- Le système crée les sessions personnalisées en fonction des listes d'activités à générer (cf. ⑦ sur la Figure 7-9).
- Le système fournit autant de séquences personnalisées qu'il y a de profils d'élèves, ainsi qu'un bilan pour l'enseignant (cf. ⑧ sur la Figure 7-9).

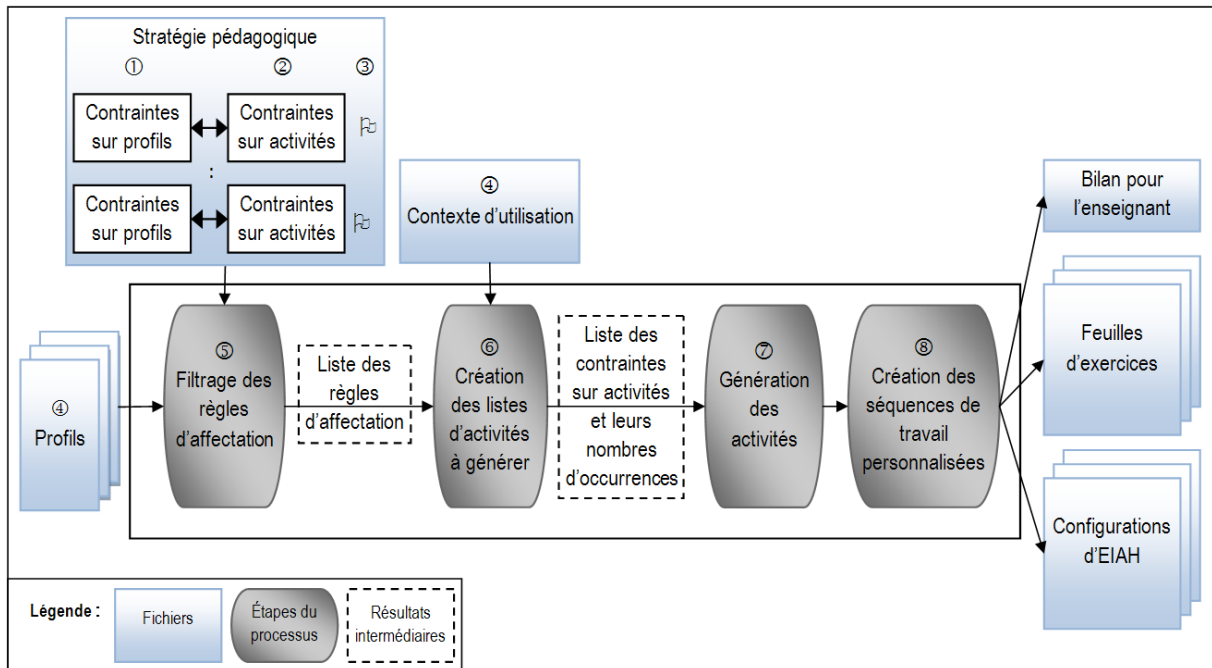


Figure 7-9 : Principe du modèle de personnalisation et de son processus d'exploitation.

7.3. ARCHITECTURE DU PROCESSUS DE CRÉATION DE SÉQUENCES PERSONNALISÉES D'ACTIVITÉS

Nous revenons à présent sur le processus de génération de séquences personnalisées pour détailler son architecture. Ce processus a été mis en œuvre dans le module Adapte [Lefevre et al. 2008b, Lefevre et al. 2009a].

Le processus prend en entrée trois éléments : les profils des apprenants (cf. E1 sur la Figure 7-10), la stratégie pédagogique de l'enseignant (cf. E2 sur la Figure 7-10) et le contexte d'utilisation des séquences d'activités personnalisées (cf. E3 sur la Figure 7-10).

Il fournit en sortie deux types d'éléments : les séquences d'activités adaptées au profil de chaque apprenant (cf. S1 sur la Figure 7-10) et un bilan des propositions à destination de l'enseignant (cf. S2 sur la Figure 7-10).

L'architecture de ce processus est constituée de quatre parties : le filtrage des règles d'affectation (cf. P1 sur la Figure 7-10), la création des listes d'activités à générer (cf. P2 sur la Figure 7-10), la génération des activités (cf. P3 sur la Figure 7-10) et la création de séquences de travail (cf. P4 sur la Figure 7-10). Nous présentons, dans la suite de cette section, chacun de ces constituants.

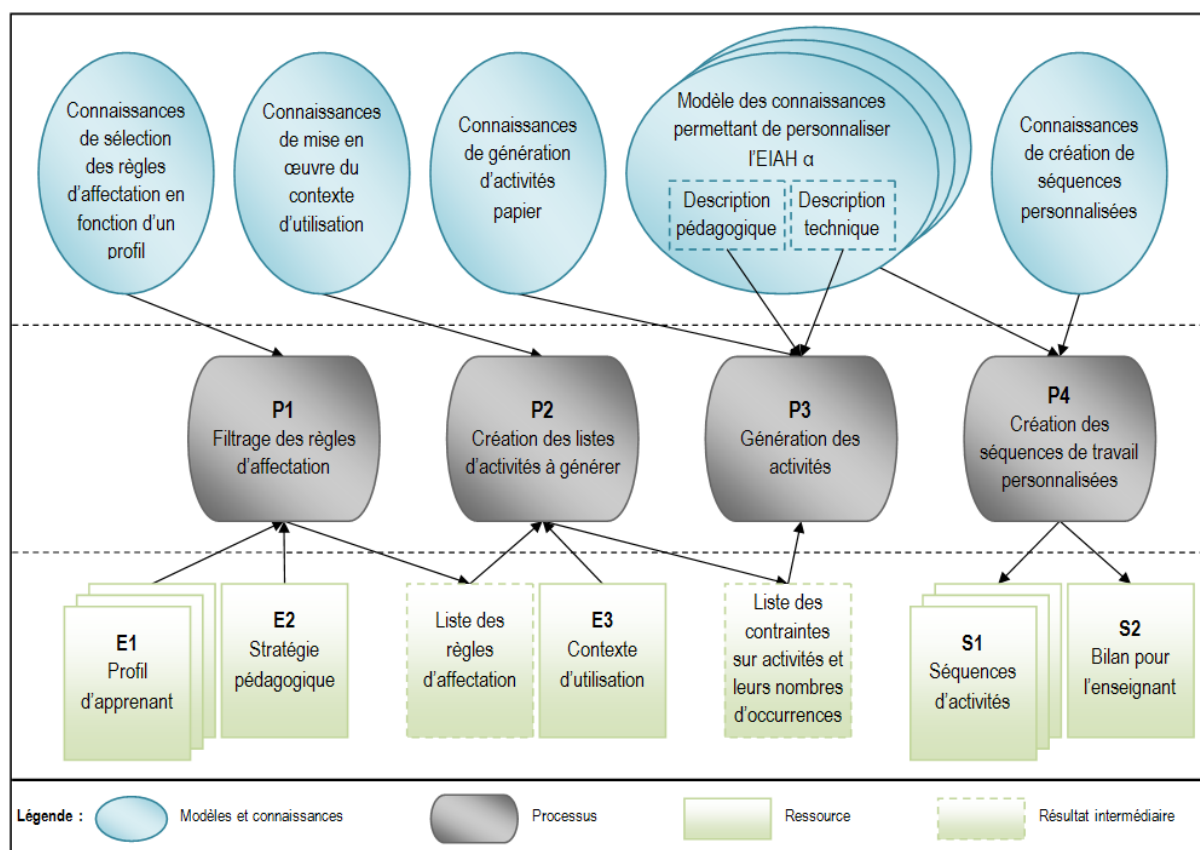


Figure 7-10 : Architecture du processus de génération de séquences d'activités personnalisées.

7.3.1. FILTRAGE DES RÈGLES D'AFFECTION

Le processus de filtrage (cf. P1 sur la Figure 7-10) permet de sélectionner les règles d'affectation dans lesquelles les conditions sont validées par les valeurs contenues dans le profil d'un apprenant.

Les **profils apprenants** sont des profils respectant le formalisme PMDL [Eyssautier-Bavay 2008, Jean-Daubias et al. 2009a] présenté dans la section 5.3. Les profils contiennent des données potentiellement de toutes disciplines et de tous niveaux, issues de profils papier-crayon venant des pratiques en classe des enseignants, mais également de profils établis par des logiciels.

La **stratégie pédagogique** rassemble un ensemble de règles d'affectation. Ces règles d'affectation sont des liens entre des éléments du profil des apprenants et des structures d'activités contenant un ensemble de contraintes permettant de générer ou de choisir une activité. La stratégie pédagogique respecte le formalisme que nous avons proposé dans la section 4.3.

Les **connaissances de sélection des règles d'affectation en fonction d'un profil** permettent d'interpréter les règles d'affectation représentées dans les stratégies pédagogiques en fonction des valeurs contenues dans les profils d'apprenants. Ces connaissances contiennent d'une part des règles permettant de savoir si les valeurs contenues dans le profil respectent les conditions des règles d'affectation des stratégies pédagogiques et d'autre part des formules permettant de convertir les valeurs du profil dans une échelle donnée. Les formules de conversion d'échelles sont fournies en Annexe C page 289.

Le **filtrage des règles d'affectation** prend en entrée la stratégie pédagogique de l'enseignant et un ensemble de profils d'apprenants. Pour chaque profil, les conditions des règles d'affectation vont être évaluées pour savoir si elles sont validées par les valeurs contenues dans le profil. Le cas échéant, les règles d'affectation sont ajoutées à la liste des règles applicables pour l'apprenant. À la fin du filtrage, on obtient autant de listes de règles d'affectation que l'on avait de profils d'apprenants en entrée.

7.3.2. CRÉATION DES LISTES D'ACTIVITÉS À GÉNÉRER

Le processus de création de listes d'activités à générer (cf. P2 sur la Figure 7-10) permet, pour chaque apprenant, de définir les activités qui doivent être générées afin qu'il travaille dans les conditions précisées dans le contexte d'utilisation.

Le **contexte d'utilisation** permet à l'enseignant de préciser des contraintes permettant de « borner » les feuilles d'exercices ou les sessions sur les EIAH, c'est-à-dire donner une durée pour la séance de travail, un nombre maximum d'exercices, etc. Le contexte d'utilisation respecte le formalisme que nous avons proposé dans la section 4.3.

Les **connaissances de mise en œuvre du contexte d'utilisation** permettent d'interpréter les contraintes contenues dans le contexte d'utilisation et de les mettre en relation avec les règles d'affectation concernées par un profil. Elles permettent de déterminer le nombre d'occurrences nécessaire pour chacune des contraintes sur activités contenues dans les règles d'affectation afin de respecter les contraintes du contexte d'utilisation. Ces connaissances représentent des règles permettant de dupliquer ou de supprimer les activités à générer contenues dans les règles d'affectation pour obtenir un nombre d'activités à générer ou une durée de travail correspondant aux contraintes du contexte d'utilisation.

La **création des listes d'activités à générer** prend en entrée les règles d'affectation concernées par un profil, ainsi que la situation pédagogique, et retourne la liste des structures d'activités à générer pour créer la séquence de travail d'un apprenant. Un nombre d'occurrences est associé à ces structures d'activités pour respecter les contraintes de temps, de nombre, etc. de la situation pédagogique. À la fin de cette étape, on obtient autant de listes d'activités à générer que l'on avait de listes de règles d'affectation et donc de profils d'apprenants.

7.3.3. GÉNÉRATION DES ACTIVITÉS

Le processus de génération d'activités (cf. P3 sur la Figure 7-10) permet de choisir ou de générer l'énoncé de l'activité et pour les activités papier sa correction.

Les **connaissances de génération d'activités papier** contiennent huit patrons d'exercices (cf. section 6.3.3.c page 131 et Annexe F page 307), ainsi que les huit générateurs d'exercices auxquels ils sont associés (cf. section 6.3.3.d page 141 et Annexe G page 333).

Un **modèle des connaissances permettant la personnalisation d'un EIAH x** (le modèle OKEP/x) est une instance du méta-modèle AKEPI, méta-modèle permettant d'acquérir les connaissances nécessaires à la personnalisation des EIAH (cf. section 6.4.3.b). Chaque modèle OKEP/x contient la description pédagogique et la description technique de l'EIAH x. La description pédagogique contient la liste des propriétés personnalisables de l'EIAH, ainsi que les règles qui permettent de les gérer. La description technique regroupe toutes les informations nécessaires pour agir concrètement sur le système : localisation du système, du générateur

d'exercices ou de la base d'exercices, place et contenu des fichiers de configuration et règles permettant de remplir ces fichiers.

La **génération des activités** prend en entrée la liste des activités à générer, ainsi que leur nombre d'occurrences. Pour chaque activité à générer, le processus génère le nombre d'activités demandées en vérifiant éventuellement que l'énoncé de chaque occurrence est unique et en s'assurant, dans l'historique de chaque apprenant, que l'activité générée ne lui a pas déjà été proposée. Pour générer les activités papier, le système se sert des générateurs d'exercices contenus dans les connaissances de génération d'activités papier. Pour les activités sur des EIAH externes, le système se sert du générateur contenu dans l'EIAH quand celui-ci existe et est utilisable via des fichiers de configuration, dans le cas contraire, il choisit les activités dans la base de données de l'EIAH.

7.3.4. CRÉATION DES SÉQUENCES DE TRAVAIL PERSONNALISÉES

Le processus de création des séquences de travail personnalisées (cf. P4 sur la Figure 7-10) permet de combiner les activités pour créer des feuilles d'exercices à imprimer ou des séquences sur un EIAH.

Les **connaissances de création de séquences personnalisées** contiennent d'une part les règles permettant de créer des feuilles d'exercices papier à partir des activités générées et d'autres part les règles permettant de créer pour les EIAH externes des séquences de travail valides.

La **création de séquences de travail personnalisées** correspond donc à une mise en forme des activités générées ou sélectionnées dans l'étape précédente pour fournir soit une feuille d'exercices papier à imprimer, soit une séance de travail sur un EIAH, soit les deux. Dans le cas des EIAH, les connaissances techniques des modèles OKEP/x sont nécessaires pour pouvoir créer des fichiers de configuration valides pour chacun des EIAH. En plus des séquences de travail destinées aux apprenants, le processus fournit à l'enseignant un bilan contenant le résumé des séquences proposées, et, dans le cas des activités papier, leur correction.

À la fin du processus de génération de séquences personnalisées, on obtient autant de feuilles d'exercices papier et/ou de configuration d'EIAH que l'on avait de profils d'apprenants. Les séquences créées sont toujours proposées à l'enseignant pour validation ou modification afin de lui laisser jusqu'au bout du processus sa place de pédagogue.

7.4. CE QU'IL FAUT RETENIR

Dans la section 4.3, nous avons présenté le modèle que nous proposons pour permettre une personnalisation unifiée des activités d'apprentissage, le modèle PERSUA2. Pour pouvoir être intégré à un système et surtout exploité par celui-ci, nous lui avons associé un processus permettant à partir de profils d'apprenants décrits selon le formalisme PMDL d'obtenir des séquences de travail adaptées aux profils des apprenants mais surtout correspondant à une stratégie pédagogique donnée.

SYNTHÈSE DE LA PARTIE

Dans les Chapitres 4 et 7, nous avons défini le modèle de personnalisation unifiée associé à un processus permettant de l'exploiter. Ce modèle a pour but de permettre la personnalisation de plusieurs types d'activités, sur différents supports (papier ou logiciel), à travers un processus unique et donc un outil unique. Il peut également être utilisé lors de la conception d'un nouvel EIAH souhaitant proposer une personnalisation des activités qu'il contient. Ce modèle de personnalisation unifiée s'appuie d'une part sur l'utilisation de profils d'apprenants et d'autre part sur l'adaptation des activités pédagogiques aux besoins pédagogiques de l'agent (enseignant ou EIAH) souhaitant la personnalisation.

Pour pouvoir utiliser des profils d'apprenants, nous avons défini dans le Chapitre 5 le modèle cPMDL, modèle des contraintes permettant de sélectionner une partie d'un profil. Ces contraintes sur profils sont formalisées à partir du langage de modélisation PMDL et sont de trois types : des contraintes sur profils portant sur une valeur, des contraintes sur profils portant sur un élément et des contraintes sur profils portant sur un nombre d'occurrences. L'identification de ces trois types de contraintes s'est faite suite à des entretiens avec des enseignants et la formalisation a été proposée en bénéficiant des possibilités offertes par le langage PMDL. Afin de répondre à des besoins non identifiés *a priori*, il est possible d'augmenter le modèle en définissant de nouveaux types de contraintes sur profils.

Pour adapter les activités à un besoin pédagogique donné, nous avons proposé dans le Chapitre 6 l'approche GEPPETO, qui a été déclinée en GEPPETO_P pour la personnalisation des activités sur support papier et en GEPPETO_S pour la personnalisation des activités au sein des EIAH. GEPPETO_P permet ainsi d'adapter les activités contenues dans la typologie des activités papier qui lui est associée. Cette typologie peut être augmentée pour prendre en compte des activités qui n'appartiendraient pas au corpus initial. GEPPETO_S permet d'adapter les activités contenues dans les EIAH et les paramètres de ces EIAH (fonctionnalités, rétroactions, interface). L'approche GEPPETO_S permet d'adapter les EIAH quel que soit leur type (simulateurs, tuteurs intelligents, micromondes, exercices, hypermédias) proposant des activités individuelles d'apprentissage aux apprenants, mais ne prend actuellement pas en compte les activités collaboratives ou collectives. GEPPETO_S est basée sur le méta-modèle AKEPI qui peut être augmenté pour considérer de nouveaux types de paramètres afin de traiter des versants non exploités des EIAH pris en compte ou éventuellement pour considérer de nouveaux types de systèmes. L'approche GEPPETO peut être utilisée au sein du modèle de personnalisation unifiée, mais également en dehors de ce modèle pour fournir des systèmes permettant d'adapter les activités pédagogiques aux besoins des enseignants sans utiliser de profils d'apprenants.

Le modèle de personnalisation unifiée que nous proposons n'a à notre connaissance pas d'équivalent à l'heure actuelle. Il a été défini suite à des entretiens avec des enseignants pour identifier leurs pratiques en matière de personnalisation. S'il n'est pas exhaustif, il possède l'avantage d'être ouvert. Il a en effet été conçu afin d'être extensible et peut ainsi évoluer pour prendre en compte de nouvelles pratiques ou de nouveaux besoins éducatifs. Il devrait donc permettre, à moindre coût, d'intégrer de nouveaux types d'EIAH, comme ceux proposant des situations d'apprentissage collaboratives ou collectives. La prise en compte de tels EIAH est une perspective qui sera discutée dans la conclusion de ce manuscrit.

PARTIE 3. MISE EN OEUVRE & ÉVALUATION

PLAN DE LA PARTIE

Dans cette partie, nous expliquons comment nous avons mis en œuvre nos contributions théoriques dans le logiciel Adapte, puis nous présentons les évaluations que nous avons menées autant sur les contributions théoriques que sur leur mise en œuvre.

Ainsi, dans le Chapitre 8, nous détaillons le fonctionnement du logiciel Adapte avant de décrire son architecture en situant sur celle-ci nos contributions théoriques. Nous présentons ensuite les différents modes d'utilisation d'Adapte. Enfin, nous reprenons chacun des scénarios d'usage présentés en introduction du manuscrit pour indiquer quelle solution Adapte, et donc nos contributions théoriques, apportent pour améliorer ces situations.

Dans le Chapitre 9, nous présentons les évaluations que nous avons effectuées concernant nos contributions théoriques, ainsi que sur leur mise en œuvre. Pour cela, nous présentons notre méthodologie, la grille d'analyse que nous avons retenue, puis nous décrivons les expérimentations menées dans le cadre de la thèse. Nous en faisons ensuite une synthèse, avant de conclure sur les évaluations qu'il nous semble pertinent de conduire pour achever la validation de ce travail.

CHAPITRE 8. ADAPTE, OU COMMENT AIDER
L'ENSEIGNANT À PROPOSER DES
ACTIVITÉS PERSONNALISÉES À CHACUN
DE SES APPRENANTS

PLAN DU CHAPITRE

8.1.	Introduction.....	201
8.2.	Fonctionnement du point de vue des utilisateurs	201
8.2.1.	Intégration d'un nouvel EIAH	202
8.2.2.	Création de séquences de travail personnalisées.....	204
a.	Création de stratégies pédagogiques.....	205
a.1.	Création de structures d'activités	205
a.2.	Création de contraintes sur profils.....	208
a.3.	Création de règles d'affectation d'activités aux apprenants	211
b.	Spécification du contexte d'utilisation d'une stratégie pédagogique.....	213
c.	Création de séquences personnalisées d'activités	215
8.2.3.	Conversion d'activités vers une norme pédagogique	217
8.3.	Architecture d'Adapte.....	217
8.3.1.	Processus pour l'intégration d'un nouvel EIAH	218
8.3.2.	Processus pour la création de séquences de travail personnalisées.....	218
8.3.3.	Place de nos contributions théoriques dans Adapte	222
8.4.	Modes d'utilisation d'Adapte	223
8.4.1.	Personnalisation des activités : avec intervention de l'enseignant et exploitation de profils d'apprenants.....	223
8.4.2.	Personnalisation des activités : sans intervention de l'enseignant et exploitation de profils d'apprenants.....	224
8.4.3.	Personnalisation des activités : avec intervention de l'enseignant sans exploitation de profils d'apprenants.....	225
8.5.	Retour sur les scénarios d'usage.....	226

PUBLICATIONS RELATIVES À CE CHAPITRE

- [Lefevre et al. 2007b] Lefevre, M., Jean-Daubias, S., Guin, N. « Adapte, un module pour proposer des activités personnalisées », Rapport de recherche RR-LIRIS-2007-027, LIRIS, Villeurbanne, France, 2007.
- [Lefevre 2008] Lefevre, M. « Adapte, un outil pour proposer des activités personnalisées au sein du projet PERLEA ». Poster, 2nde Rencontres Jeunes Chercheur en Environnements Informatiques pour l'Apprentissage Humain (RJC EIAH 2008), Lille, France, pp. 145-146, 15-16 mai 2008.
- [Lefevre et al. 2008a] Lefevre, M., Guin, N., Jean-Daubias, S. « Adapte, a tool for the teacher to personalize activities », Rapport de recherche RR-LIRIS-2008-008 (version longue de l'article d'ITS 2008), LIRIS, Villeurbanne, France, 2008.
- [Lefevre et al. 2008b] Lefevre, M., Guin, N., Jean-Daubias, S. « Adapte, a tool for the teacher to personalize activities ». 9th International Conference on Intelligent Tutoring Systems (ITS'2008), Montréal, Canada, ISBN 978-3-540-69130-3, pp. 699-701, 23-27 juin 2008.
- [Jean-Daubias et al. 2009d] Jean-Daubias, S., Lefevre, M., Guin, N. « Adapte, un outil générique pour proposer des activités pédagogiques personnalisées ». Workshop Prise en Compte de l'Utilisateur dans les Systèmes d'Information (PeCUSI), INFORSID 2009, Toulouse, France, pp. 51-62, 26 mai 2009.
- [Lefevre et al. 2009a] Lefevre, M., Cordier, A., Jean-Daubias, S., Guin, N. « A Teacher-dedicated Tool Supporting Personalization of Activities ». ED-MEDIA 2009 - World Conference on Educational Multimedia, Hypermedia & Telecommunications, Honolulu, Hawaii, pp. 1136-1141, 22-26 juin 2009.

8.1. INTRODUCTION

Le logiciel Adapte s'intègre dans l'environnement EPROFILEA associé au projet PERLEA [PERLEA 2002, Jean-Daubias 2003]. Le rôle de cet environnement est d'assister l'enseignant dans sa gestion de profils créés par l'enseignant lui-même ou issus d'EIAH. Ces profils peuvent porter sur toutes les disciplines et tous les niveaux, scolaires ou universitaires. EPROFILEA comporte deux parties : la constitution de profils conformes à l'environnement et leur exploitation. Le logiciel Adapte est une des exploitations possibles des profils : il permet de fournir à chaque apprenant des activités adaptées à son profil tout en respectant les choix pédagogiques de son enseignant. Ces activités peuvent être des activités papier-crayon proposées par le système ou des activités logicielles gérées par un autre EIAH [Lefevre et al. 2009a].

Dans ce chapitre, nous présentons le fonctionnement du logiciel Adapte du point de vue des utilisateurs, puis nous présentons son architecture et les processus nécessaires à son fonctionnement en faisant le lien avec nos contributions théoriques. Nous décrivons ensuite les différents modes d'utilisation du logiciel avant de voir comment celui-ci peut servir dans les scénarios qui ont été présentés dans l'introduction de ce manuscrit.

8.2. FONCTIONNEMENT DU POINT DE VUE DES UTILISATEURS

Le logiciel Adapte comporte trois parties (cf. Figure 8-1) : une partie « Intégration d'un nouvel EIAH » dans laquelle un expert initialise les bases de connaissances relatives aux EIAH à personnaliser, une partie « Création de séquences de travail personnalisées » qui correspond à l'utilisation courante du logiciel, et une partie « Conversion vers une norme pédagogique » qui permet d'exporter les activités créées dans Adapte vers une norme donnée dans un but d'interopérabilité. Nous présentons dans cette section chacune de ces parties.

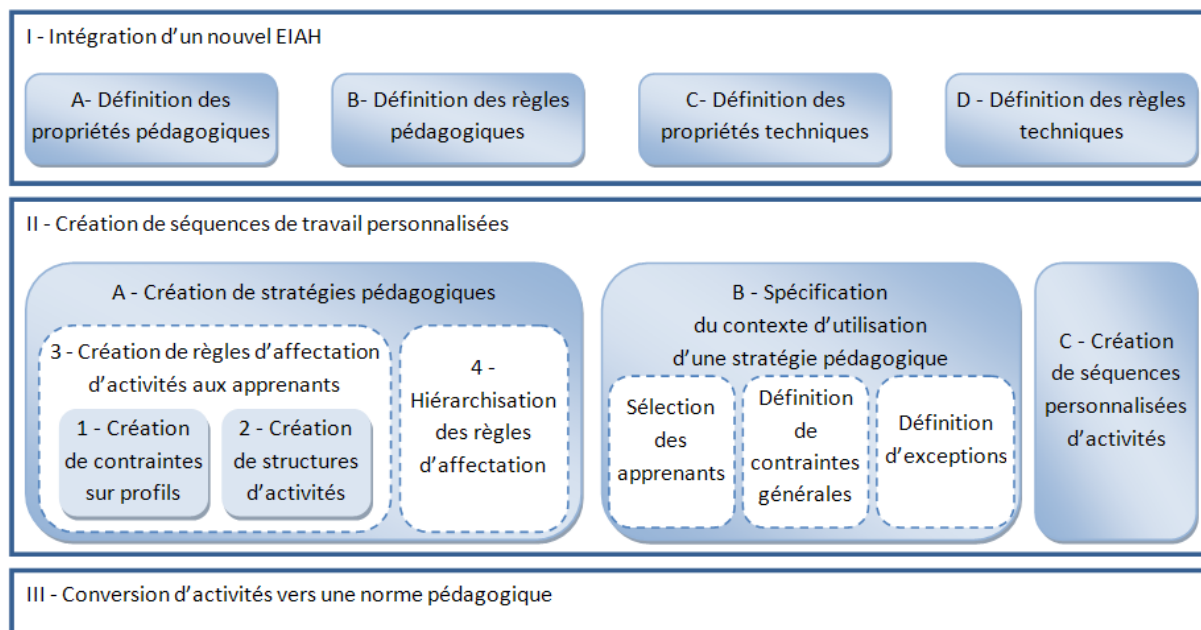


Figure 8-1 : Fonctionnement du logiciel Adapte.

8.2.1. INTÉGRATION D'UN NOUVEL EIAH

La partie « Intégration d'un nouvel EIAH » du logiciel Adapte est utilisée par un expert qui peut être le correspondant informatique de l'institution scolaire utilisant Adapte, un concepteur d'Adapte ou un concepteur du nouvel EIAH à personnaliser. Cette partie permet, pour chaque EIAH, d'intégrer les connaissances techniques et pédagogiques nécessaires. Cette partie est nécessaire pour qu'Adapte puisse personnaliser un EIAH mais elle n'est utilisée que dans la phase d'initialisation du logiciel.

Pour permettre à l'expert de fournir les connaissances utiles à la personnalisation d'un EIAH, Adapte utilise le méta-modèle AKEPI, un méta-modèle pour l'acquisition des connaissances permettant la personnalisation des EIAH (cf. section 6.4.3.b). Ce méta-modèle contient quatre parties sur lesquelles la partie « Intégration d'un nouvel EIAH » s'appuie afin de définir les modèles OKEP (cf. I sur la Figure 8-1). Chaque modèle OKEP/x est un modèle des connaissances opérationnelles permettant la personnalisation de l'EIAH x.

Figure 8-2 : Écran d'Adapte permettant de définir les propriétés de la description pédagogique du modèle OKEP/x d'un EIAH x.

Dans la première étape de la partie « Intégration d'un nouvel EIAH » (cf. I-A sur la Figure 8-1), l'expert définit les propriétés de la description pédagogique de l'EIAH comme défini dans le méta-modèle AKEPI (cf. Figure 6-40, page 161). Il définit pour cela le contenu pédagogique de l'EIAH, et si l'EIAH permet leur personnalisation, l'organisation pédagogique (comment combiner des activités pour former des séquences de travail), les fonctionnalités de l'EIAH, les rétroactions proposées à l'apprenant et l'interface du logiciel (langue, couleur, police, etc.). La Figure 8-2 présente l'écran du logiciel Adapte permettant de définir cette description pédagogique du modèle OKEP. Sur cet écran, un expert a décrit les activités contenues dans le

logiciel ABALECT⁴ [Chevé 2005]. Dans ce logiciel, toutes les activités sont de type « Exercice » (cf. ① sur Figure 8-2) et peuvent être paramétrées à l'aide de quatre propriétés (cf. ② sur la Figure 8-2). Ces quatre propriétés ont toutes le même format : un titre (cf. ① sur Figure 8-3), un identifiant unique (cf. ② sur Figure 8-3), éventuellement une liste de compétences associées (cf. ③ sur Figure 8-3), la liste énumérée des valeurs possibles (cf. ④ sur Figure 8-3) et éventuellement une catégorie ou des sous-catégories (cf. ⑤ sur Figure 8-3). Les paramètres P001 et P002 (cf. ② sur la Figure 8-2) permettent de décrire le texte sur lequel les élèves vont travailler, c'est pourquoi la catégorie à laquelle ils sont rattachés est « Choix du texte ». Les propriétés P003 et P004 permettent de décrire le ou les exercices à faire sur le texte, c'est pourquoi la catégorie à laquelle ils sont rattachés est « Choix de l'exercice ».

Figure 8-3 : Écran d'Adapte permettant de définir un paramètre du modèle OKEP/x d'un EIAH x.

Dans la seconde étape (cf. I-B sur la Figure 8-1), l'expert définit les règles permettant de contraindre les propriétés de la description pédagogique du modèle OKEP. Il utilise pour cela des règles « si-alors » respectant les formats proposés dans la section 6.4.3.b (cf. Figure 6-51 page 165). La Figure 8-4 présente l'écran d'Adapte permettant de définir ces règles. Sur cet exemple, concernant toujours le logiciel ABALECT, l'expert a défini l'ensemble des règles permettant de contraindre les propriétés pédagogiques d'OKEP/ABALECT. La règle affichée à l'écran indique que si le paramètre P001 a pour valeur « CE1 », alors le domaine de valeurs du paramètre P002 doit être réduit à la liste des textes sélectionnés.

⁴ Le logiciel ABALECT est présenté dans la section 6.4.3.a, page 148, et le modèle OKEP/ABALECT est fourni en Annexe I, page 292, et en partie expliqué dans la section 6.4.3.d.1, page 161 .

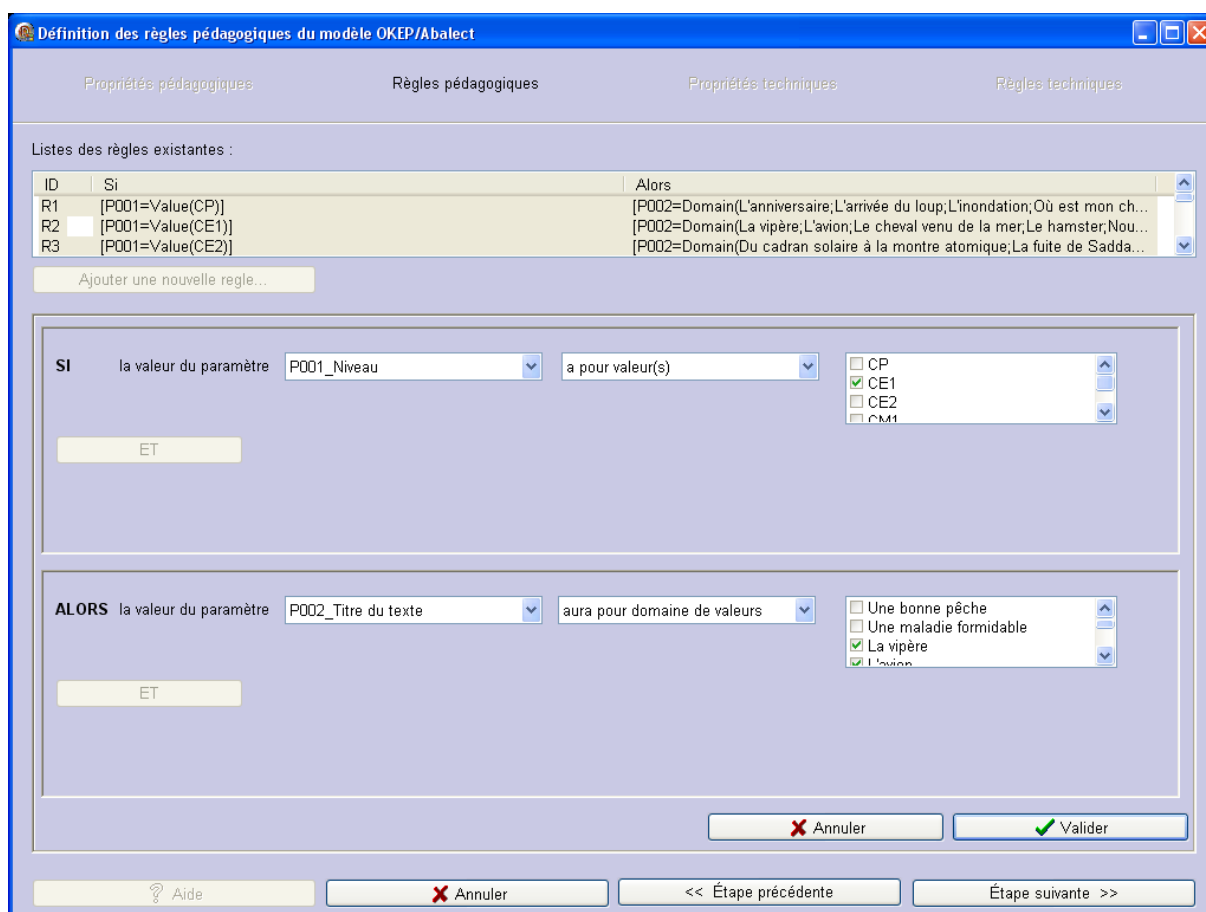


Figure 8-4 : Écran d'Adapte permettant de définir les règles de la description pédagogique du modèle OKEP/x d'un EIAH x.

Cette description pédagogique, contenant l'ensemble des propriétés personnalisables de l'EIAH ainsi que les règles permettant de contraindre l'utilisation de ces propriétés, est mémorisée dans le modèle OKEP/x de l'EIAH x décrit.

Dans les troisième et quatrième étapes (cf. I-C et I-D sur la Figure 8-1), toujours en respectant le même méta-modèle, deux écrans permettent de définir les propriétés et les règles contenues dans la description technique du modèle OKEP de l'EIAH (cf. section 6.4.3.b.2, page 165). Le premier écran permet de lister les dossiers, les fichiers et les menus utiles à la personnalisation et le second écran permet d'indiquer comment créer ou compléter les fichiers de configuration et/ou d'indiquer quels items des menus l'utilisateur de l'EIAH devra utiliser pour personnaliser l'EIAH.

Cette partie d'Adapte instancie le processus AKASI, présenté dans la section 6.4.3.c page 167, qui utilise le méta-modèle AKEPI pour créer un modèle OKEP/x propre à l'EIAH x. Un exemple de modèle OKEP issu de cette partie « Intégration d'un nouvel EIAH » est fourni en Annexe I, page 349.

8.2.2. CRÉATION DE SÉQUENCES DE TRAVAIL PERSONNALISÉES

La partie du logiciel dédiée à la création de séquences de travail personnalisées permet d'obtenir des activités papier ou logicielles. Elle peut être utilisée selon plusieurs modes, détaillés dans la section 8.4, dans lesquels l'enseignant intervient à des degrés divers (allant de l'utilisation du logiciel exclusivement par un enseignant à l'utilisation du logiciel sans enseignant). Nous

détaillons ici l'usage le plus courant : la création de séquences d'activités à partir du profil des apprenants et selon les intentions pédagogiques de l'enseignant. Ce mode d'utilisation nécessite trois étapes (cf. II sur la Figure 8-1) : la création par l'enseignant de ses stratégies pédagogiques, la spécification par l'enseignant du contexte d'utilisation et la création par le système de séquences d'activités personnalisées.

A. CRÉATION DE STRATÉGIES PÉDAGOGIQUES

Créer une stratégie pédagogique consiste à décrire la manière dont on souhaite que les activités soient affectées aux apprenants. L'affectation d'une activité à un apprenant est une tâche difficile à modéliser puisqu'il convient de capter l'expertise des enseignants et de la représenter en tant que connaissance dans le système d'assistance. C'est sur la base d'un recueil d'expertises auprès d'enseignants que nous avons retenu des règles permettant à un enseignant d'affecter un type d'exercices à un élément de profil. Ces règles ont été formalisées dans le modèle de personnalisation présenté dans la section 4.3. Ce modèle a ensuite été mise en œuvre dans le logiciel Adapte. Ainsi, pour créer une stratégie pédagogique dans Adapte, un utilisateur crée des structures d'activités comportant un ensemble de contraintes permettant de générer ou de sélectionner une activité papier ou logicielle (cf. II-A-1 sur la Figure 8-1). Après avoir créé une structure d'activités, il définit des contraintes sur la structure de profils décrivant le profil des apprenants (cf. II-A-2 sur la Figure 8-1). Le lien entre ces deux types de contraintes est appelé une règle d'affectation d'activités aux apprenants (cf. II-A-a3 sur la Figure 8-1). C'est l'ensemble de ces règles d'affectation et leur hiérarchisation en fonction de leur priorité qui constitue la stratégie pédagogique de l'enseignant (cf. II-A-4 sur la Figure 8-1). Nous détaillons maintenant chacune de ces étapes.

A.1. CRÉATION DE STRUCTURES D'ACTIVITÉS

La création de structures d'activités permet à l'utilisateur de définir des activités, papier ou logicielles, correspondant à ses habitudes de travail.

En ce qui concerne la création d'activités papier, l'utilisateur sélectionne un type d'exercices (Table de conjugaison par exemple) qui est associé à un patron d'exercices (Tableau à double entrée dans ce cas). Adapte propose les huit patrons d'exercices (cf. **A** à **H** sur la Figure 8-5) présentés dans la section 6.3.3.c (cf. A à H sur la Figure 6-2) qui permettent de créer 15 types d'exercices (cf. A1 à H sur les Figures 6-2 et 8-5). À partir du type d'exercices choisi, l'utilisateur peut définir les contraintes qu'il souhaite afin que le système génère des exercices qui lui conviennent. Pour cela, le système lui propose l'écran de définition de contraintes adapté au type d'exercices sélectionné. Un exemple d'écran permettant cette définition de contraintes est fourni sur la Figure 8-6.

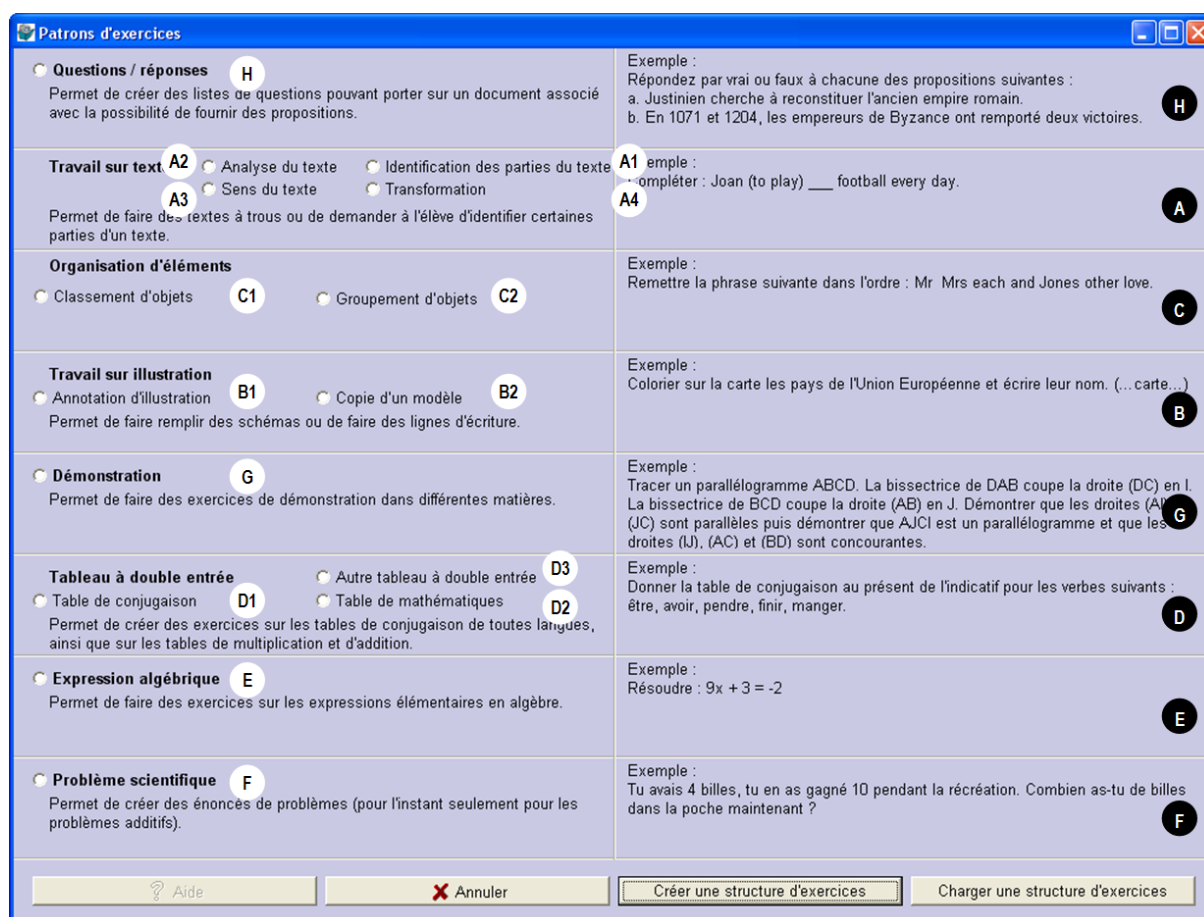


Figure 8-5 : Écran de choix d'un patron d'exercices dans Adapte.

La Figure 8-6 est un exemple d'écran permettant de spécifier des contraintes sur activités afin de créer une structure d'activités. Sur cet écran, associé au générateur d'exercices permettant de créer, entre autres, des tables de conjugaison⁵, on peut voir que l'utilisateur choisit une consigne parmi celles prédéfinies dans le logiciel (cf. ① sur la Figure 8-6). Il précise ensuite que l'exercice doit porter sur la conjugaison du présent (cf. ② sur la Figure 8-6) pour la langue anglaise (cf. ③ sur la Figure 8-6). Concernant le choix des verbes, il ne spécifie pas de contrainte particulière (cf. ④ sur la Figure 8-6), les verbes seront donc choisis aléatoirement dans la base de verbes. L'utilisateur précise également que l'exercice porte sur la conjugaison des verbes uniquement aux trois personnes du singulier et à la première et troisième personne du pluriel (cf. ⑤ sur la Figure 8-6). Il définit ensuite trois verbes prioritaires (cf. ⑥ sur la Figure 8-6). Enfin, il termine cette création de structure d'activités en indiquant que l'exercice contiendra cinq verbes à conjuguer dont deux parmi les deux verbes prioritaires qu'il a choisis (cf. ⑦ sur la Figure 8-6).

⁵ Le patron d'exercices associé à ce générateur est fourni en Annexe F page 252 et la présentation du générateur est fournie en Annexe G page 280.

Figure 8-6 : Écran d'Adapte permettant de définir des contraintes sur le patron opérationnel « Table de conjugaison » du générateur « Tableau à double entrée ».

En ce qui concerne la création d'activités au sein des EIAH, l'utilisateur définit des contraintes sur activités pouvant porter sur la génération des activités quand l'EIAH à personnaliser contient un générateur, et/ou sur la sélection des activités dans les bases de données du système. Il peut aussi définir des contraintes sur les fonctionnalités, les rétroactions ou l'interface du logiciel. Pour cela, le système génère dynamiquement une interface en fonction du modèle OKEP de l'EIAH concerné. Sur l'exemple de la Figure 8-7, on peut voir l'écran généré pour le logiciel ABALECT [Chevé 2005] à partir du modèle OKEP/ ABALECT⁶ issu de la partie « Intégration d'un nouvel EIAH » (cf. section 8.2.1). Sur cette figure, on peut voir que l'utilisateur choisit de contraindre une activité de type « Exercice » (cf. ① sur la Figure 8-7). Pour cela, il stipule que le texte doit être destiné aux élèves de CE1 (cf. ② sur la Figure 8-7). Il ne sélectionne pas un texte particulier (cf. ③ sur la Figure 8-7), c'est donc le système qui fera ce choix parmi les textes respectant la première contrainte (cf. ④ sur la Figure 8-7). L'utilisateur demande ensuite à ce que le type d'exercices concerne la compréhension du texte (cf. ⑤ sur la Figure 8-7). Adapte réduit donc le domaine de valeurs de la propriété concernée : les textes à sélectionner ne sont plus qu'au nombre de quatre (cf. ⑥ sur la Figure 8-7).

⁶ Le modèle OKEP/ABALECT est fourni en Annexe I, page 292, et en partie expliqué dans la section 6.4.3.d.1, page 161 .

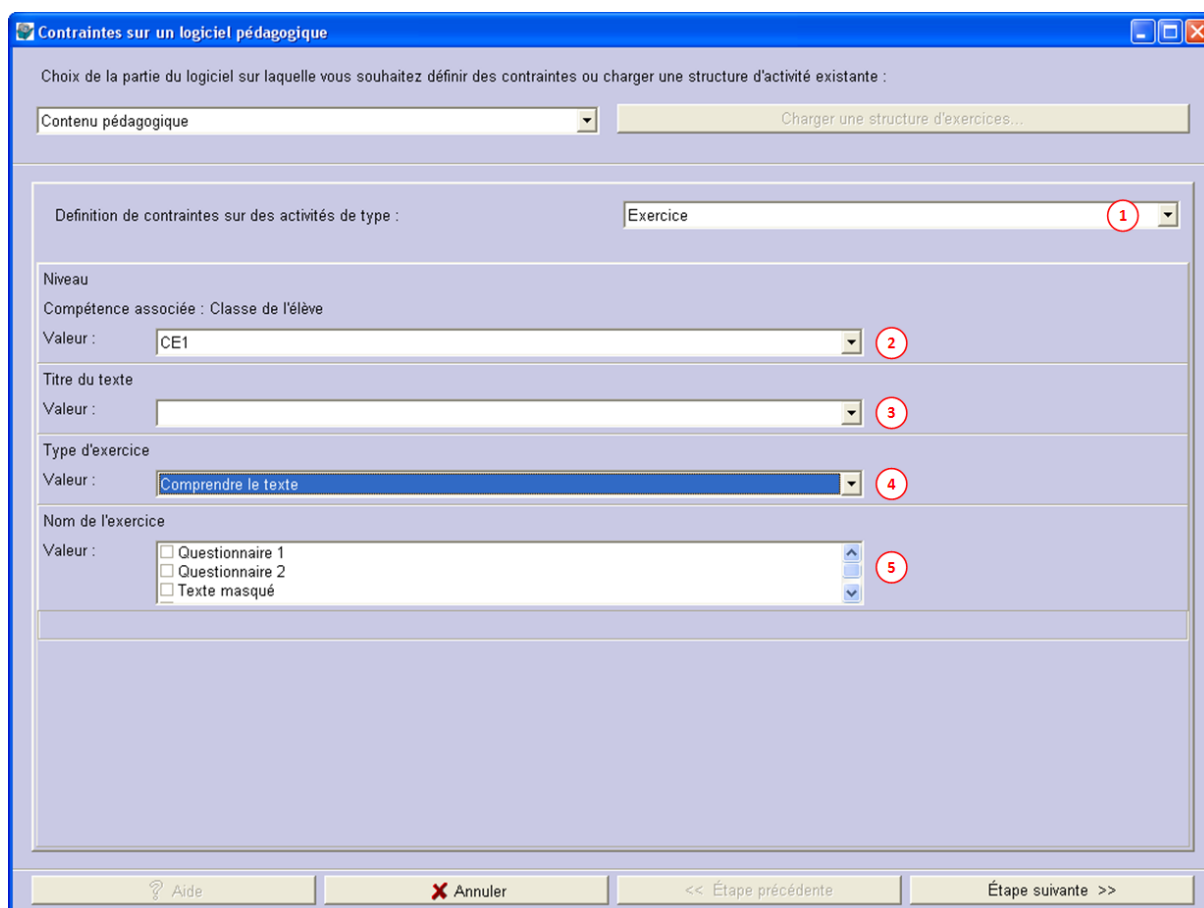


Figure 8-7 : Écran d'Adapte permettant de définir des contraintes sur une activité logicielle.

Toutes ces contraintes, que ce soit sur une activité papier ou une activité logicielle, sont mémorisées dans une structure d'activités. Afin de faciliter la réutilisation des structures d'activités, elles sont enrichies par des métadonnées (cf. section 8.3.2 page 218), proposées par le système et complétées par l'utilisateur.

A.2. CRÉATION DE CONTRAINTES SUR PROFILS

La création de contraintes sur profils permet à l'utilisateur de préciser quels éléments du profil seront discriminants dans la personnalisation des séquences d'activités. Selon la valeur associée à ces éléments, des structures d'activités différentes permettront de faire travailler les apprenants concernés sur des activités différentes.

Les contraintes sur profils respectent les modèles de contraintes présentés dans la section 5.4 : contraintes sur profils portant sur une valeur, contraintes sur profils portant sur un élément et contraintes sur profils portant sur un nombre d'occurrences. Dans le logiciel, seules les contraintes portant sur valeur ou sur élément sont mises en œuvre.

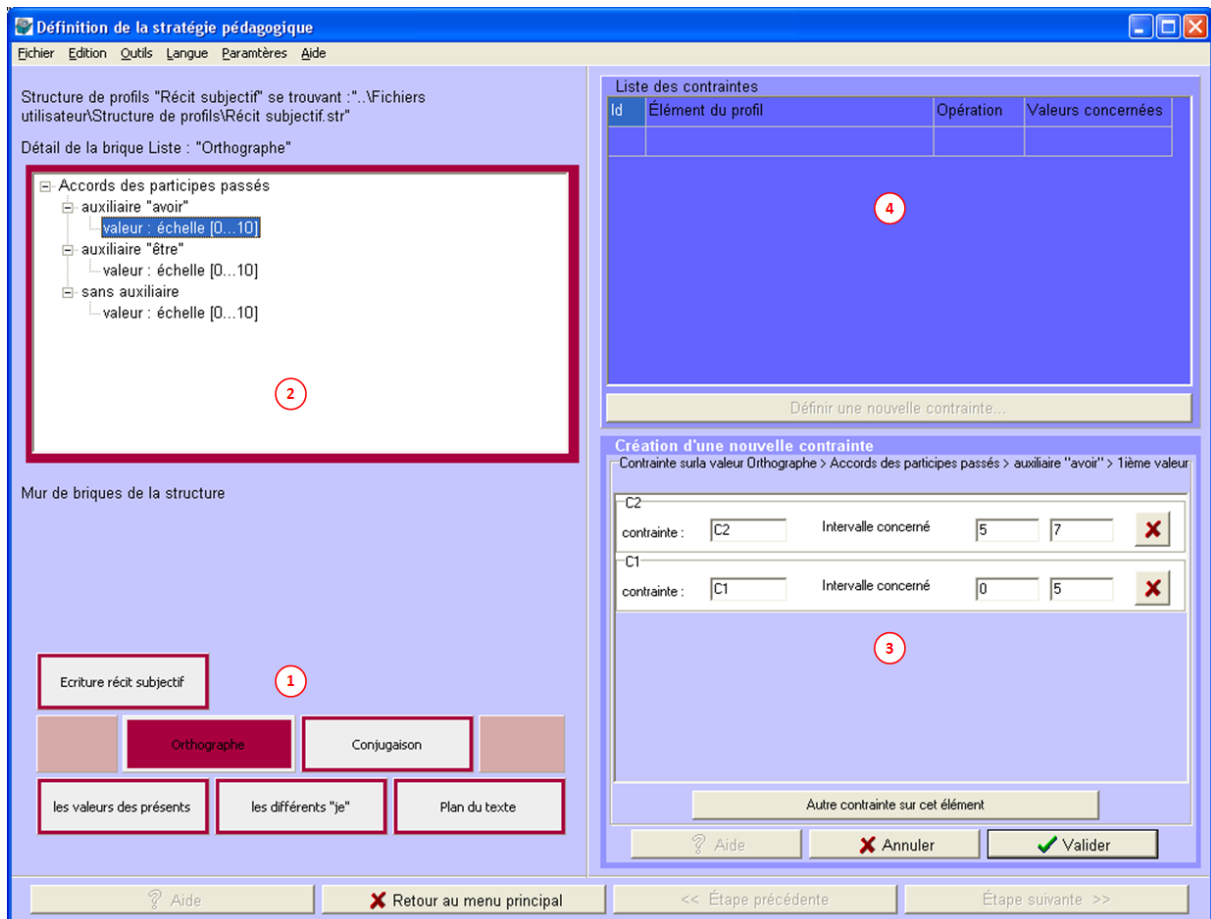


Figure 8-8 : Écran d'Adapte permettant de définir des contraintes sur profils portant sur une valeur numérique.

Les contraintes sur profils portant sur une valeur peuvent porter sur des valeurs numériques, textuelles ou sur des entiers positifs (cf. section 5.4.1).

La Figure 8-8 montre l'exemple de la définition d'une contrainte sur une valeur numérique. La partie gauche de l'écran affiche la structure d'un profil⁷ en utilisant la métaphore du mur de briques (cf. ① sur la Figure 8-8), où chaque brique correspond à un élément de la structure de profils. Lorsque qu'une brique est sélectionnée, son contenu s'affiche en haut à gauche de l'écran (cf. ② sur la Figure 8-8). Sur cet exemple, l'utilisateur a sélectionné la brique « Orthographe » qui contient une liste hiérarchique de composantes. Le premier niveau de la brique contient une composante « Accords des participes passés » qui est décomposée en trois sous-composantes : « auxiliaire avoir », « auxiliaire être » et « sans auxiliaire ». Chaque composante de niveau le plus bas contiendra, dans le profil des apprenants, une valeur respectant l'échelle nommée [0..10] correspondant à une note numérique comprise entre 0 et 10.

Pour définir une contrainte sur la valeur numérique de la sous-composante « auxiliaire avoir », l'utilisateur a sélectionné cet élément dans la partie gauche de l'écran (cf. ② sur la Figure 8-8), et a ensuite défini des intervalles de valeurs (cf. ③ sur la Figure 8-8). Chaque intervalle correspond à une contrainte sur profils différente pour ce même élément. Ici, l'utilisateur a sélectionné l'intervalle [0..5] pour la contrainte C1 et l'intervalle [5..7] pour la contrainte C2. L'utilisateur peut ainsi définir autant d'intervalles différents qu'il le souhaite sur cet élément. Une fois les contraintes C_i ainsi créées, elles sont ajoutées à la liste des contraintes (cf. ④ sur la Figure 8-8)

⁷ Cette structure de profils a été créé par Bérengère, enseignante de français au collège, lors de la mise à l'essai du logiciel Adapte.

sur laquelle l'utilisateur s'appuiera pour créer des règles d'affectation des activités aux apprenants.

Les contraintes sur un entier positif se créent de la même façon avec la même interface.

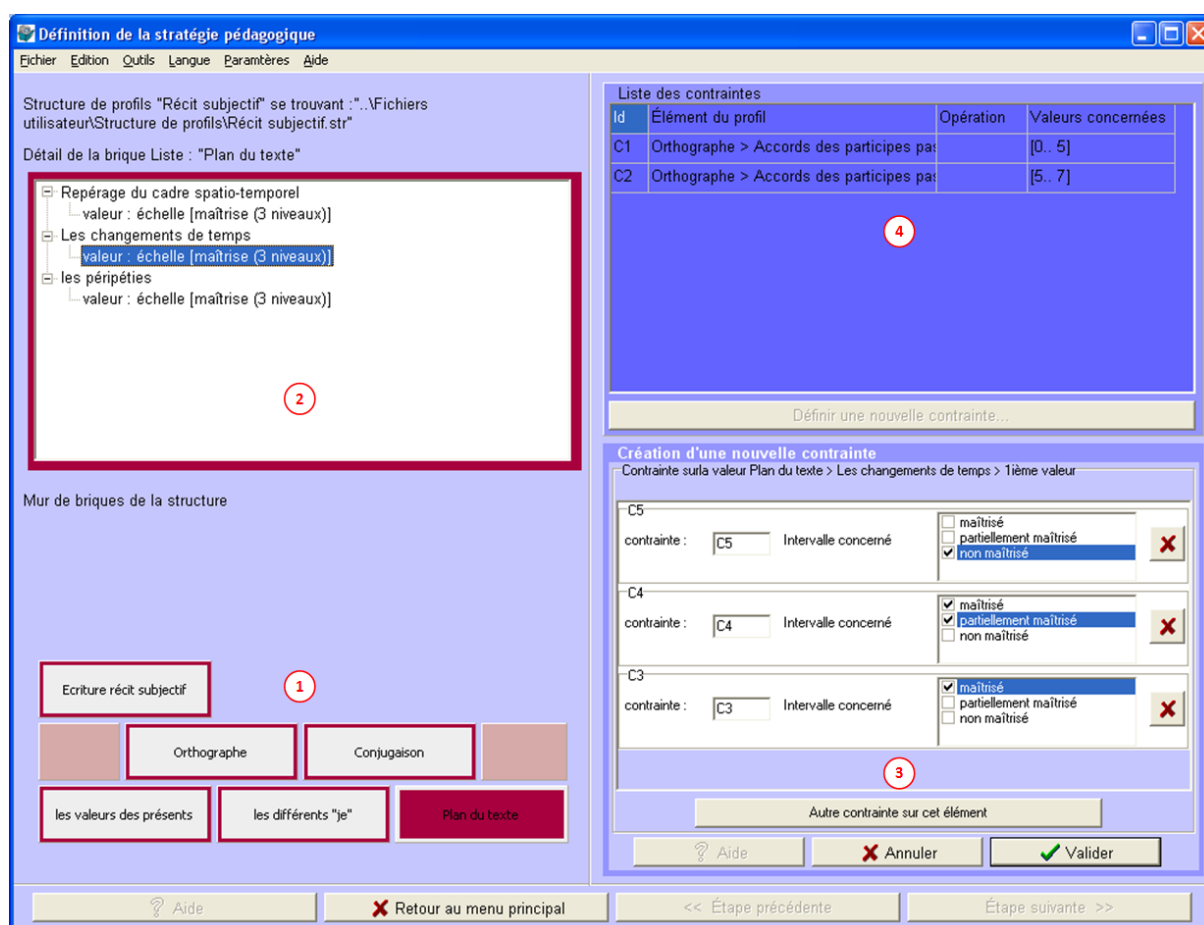


Figure 8-9 : Écran d'Adapte permettant de définir des contraintes sur profils portant sur une valeur textuelle.

La Figure 8-9 montre un exemple de définition d'une contrainte sur profils portant sur une valeur textuelle. Sur la partie gauche de l'écran, on retrouve le mur de briques (cf. ① sur la Figure 8-9). Sur cet exemple, l'utilisateur a sélectionné la brique « Plan du texte » qui contient une liste hiérarchique de composantes. Chaque composante de niveau le plus bas contiendra, dans le profil des apprenants, une valeur respectant l'échelle nommée [maîtrisé (3 niveaux)], qui correspond à la liste énumérée « non maîtrisé, partiellement maîtrisé, maîtrisé ».

Pour définir une contrainte sur la valeur textuelle de la sous-composante « Les changements de temps », l'utilisateur a sélectionné cet élément dans la partie gauche de l'écran (cf. ② sur la Figure 8-9), et a ensuite défini des ensembles de valeurs (cf. ③ sur la Figure 8-9). Chaque ensemble, contenant une ou plusieurs valeurs de la liste énumérée de l'échelle [maîtrisé (3 niveaux)], correspond à une contrainte sur profils portant sur une valeur. Ici, l'utilisateur a sélectionné la valeur « maîtrisé » pour la contrainte C3, les valeurs « partiellement maîtrisé, maîtrisé » pour la contrainte C4 et la valeur « non maîtrisé » pour la contrainte C5. Une fois les contraintes C_i créées, elles sont ajoutées à la liste des contraintes (cf. ④ sur la Figure 8-9).

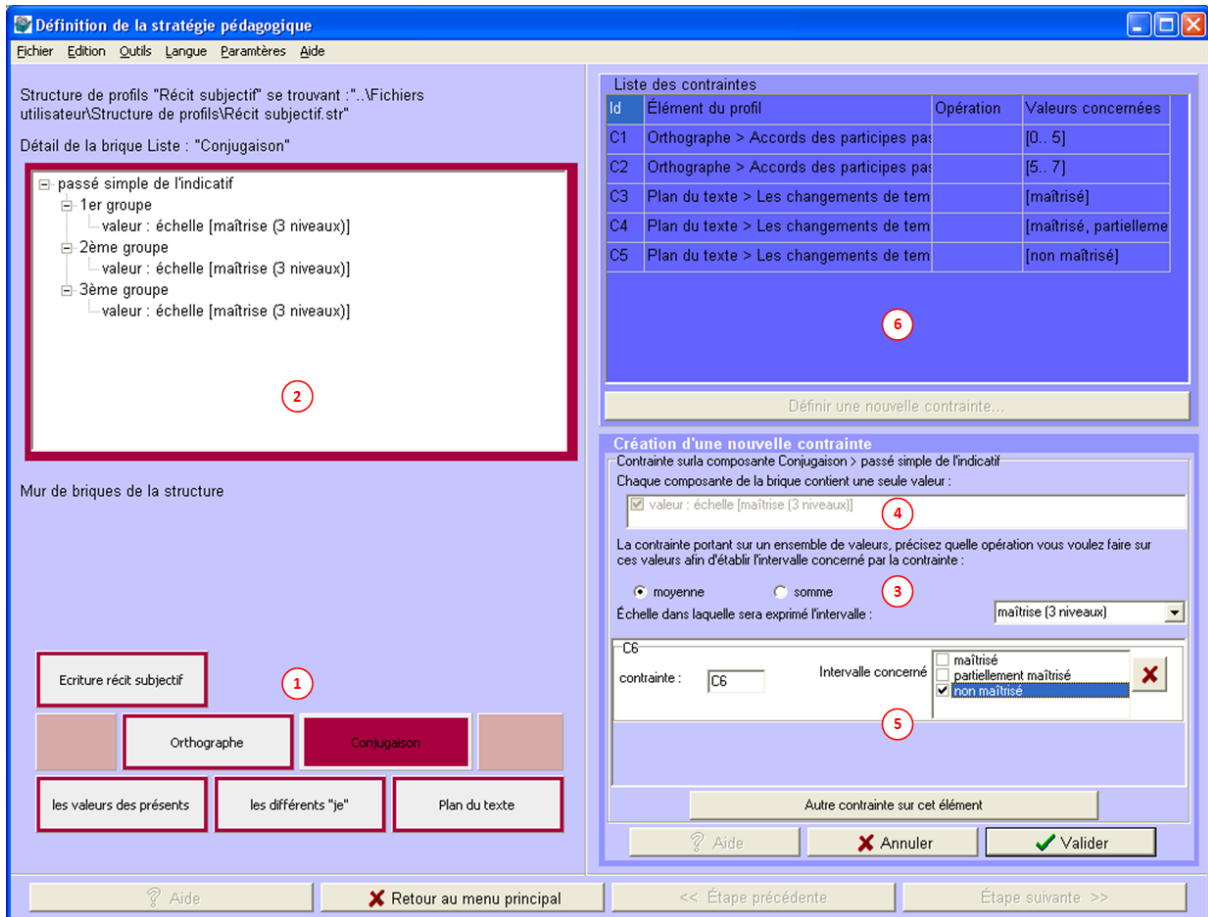


Figure 8-10 : Écran d'Adapte permettant de définir des contraintes sur profils portant sur un élément.

La Figure 8-10 donne un exemple de définition d'une contrainte sur un élément du profil. Sur la partie gauche de l'écran, on retrouve le mur de briques (cf. ① sur la Figure 8-10). Sur cet exemple, l'utilisateur a sélectionné la brique « Conjugaison ». Chaque composante de niveau le plus bas contiendra une valeur dans le profil des apprenants. Cette valeur sera donnée selon l'échelle [maîtrisé (3 niveaux)].

Pour définir une contrainte sur la composante « passé simple de l'indicatif », l'utilisateur a sélectionné cet élément dans la partie gauche de l'écran (cf. ② sur la Figure 8-10). Cet élément contenant trois sous-composantes, l'utilisateur doit préciser la façon de combiner ces valeurs (somme ou moyenne (cf. ③ sur la Figure 8-10)). Si les sous-composantes avaient eu plusieurs valeurs, l'utilisateur aurait également dû préciser quelles valeurs il souhaitait prendre en compte dans le calcul (par exemple : seulement les premières valeurs ou toutes les valeurs (cf. ④ sur la Figure 8-10)). L'utilisateur définit ensuite des ensembles de valeurs pour le résultat de la moyenne des valeurs de chaque sous-composante (cf. ⑤ sur la Figure 8-10). Chaque ensemble, contenant une ou plusieurs valeurs de la liste énumérée de l'échelle [maîtrisé (3 niveaux)], correspond à une contrainte sur profils portant sur un élément. Une fois les contraintes C_i créées, elles sont ajoutées à la liste des contraintes (cf. ⑥ sur la Figure 8-10).

A.3. CRÉATION DE RÈGLES D'AFFECTATION D'ACTIVITÉS AUX APPRENANTS

La création de règles d'affectation d'activités aux apprenants permet à l'utilisateur de faire un lien entre des structures d'activités et des contraintes sur le profil des apprenants.

Une règle d'affectation d'activités aux apprenants est donc définie comme le lien entre d'une part des contraintes permettant de sélectionner une partie du profil et d'autre part une ou

plusieurs structures d'activités utilisées pour adapter les activités que l'on fournira aux apprenants dont le profil respecte les contraintes. Une règle d'affectation a le format d'une règle « si-alors-sinon », le « sinon » étant optionnel.

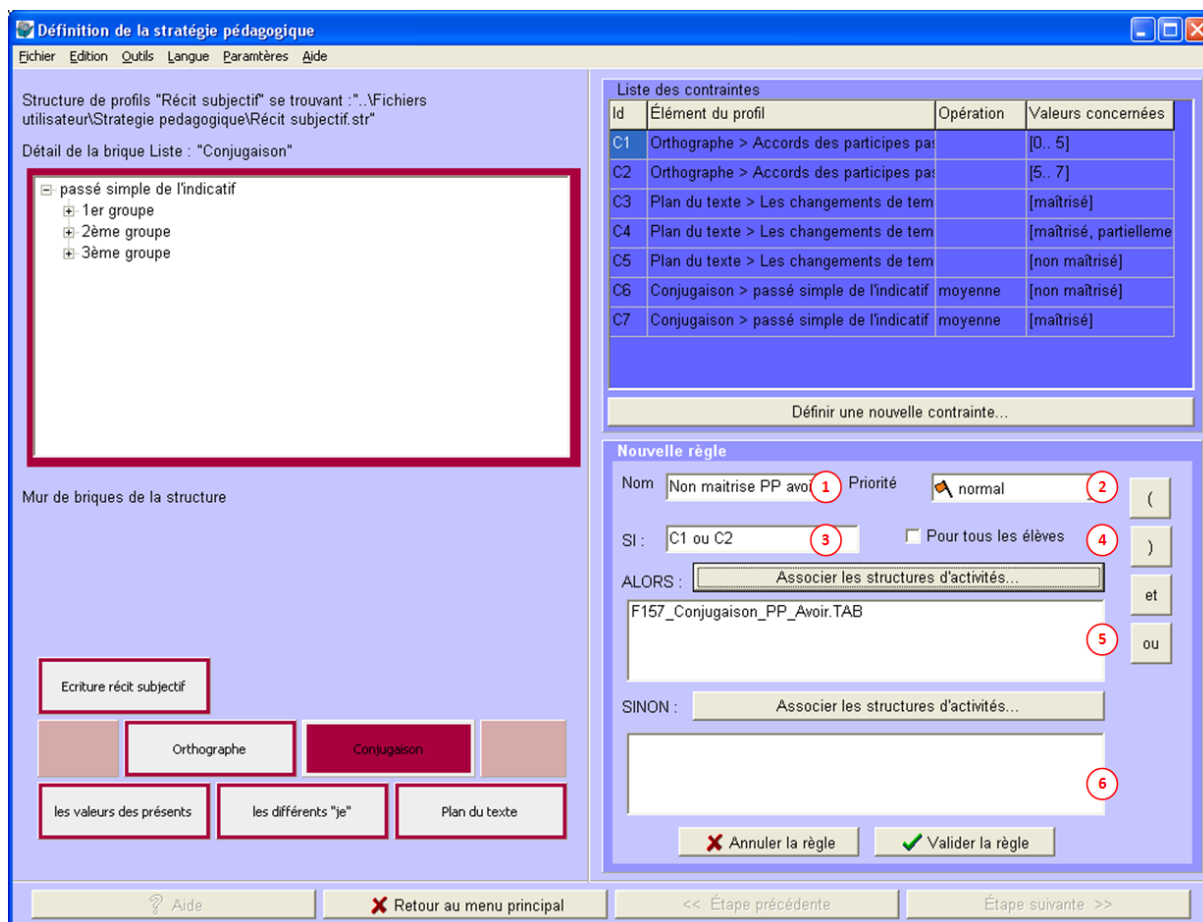


Figure 8-11 : Écran d'Adapte permettant de définir des règles d'affectation des activités aux apprenants.

La Figure 8-11 présente un exemple de définition d'une règle d'affectation des activités aux apprenants. L'utilisateur définit tout d'abord un nom pour la règle (cf. ① sur la Figure 8-11) et un niveau de priorité (cf. ② sur la Figure 8-11). Ce niveau de priorité sert dans les cas où le système a le choix entre deux activités à affecter à l'apprenant. Adapte fournira en priorité les activités associées aux règles ayant un haut niveau de priorité. L'utilisateur crée ensuite la condition de la règle soit en se servant des contraintes qu'il a définies sur la structure de profils (cf. ③ sur la Figure 8-11), soit en cochant la case indiquant que cette règle doit s'appliquer à tous les apprenants (cf. ④ sur la Figure 8-11). Dans le cas où il utilise les contraintes sur profil, l'utilisateur peut les combiner à l'aide des opérateurs « et, ou, non » et de parenthèses, comme le précise le modèle présenté dans la section 4.3. Il définit enfin les structures d'activités associées à cette condition (cf. ⑤ et ⑥ sur la Figure 8-11).

Une fois la règle d'affectation validée, elle est ajoutée à la liste des règles contenues dans la stratégie pédagogique (cf. ① sur la Figure 8-12).

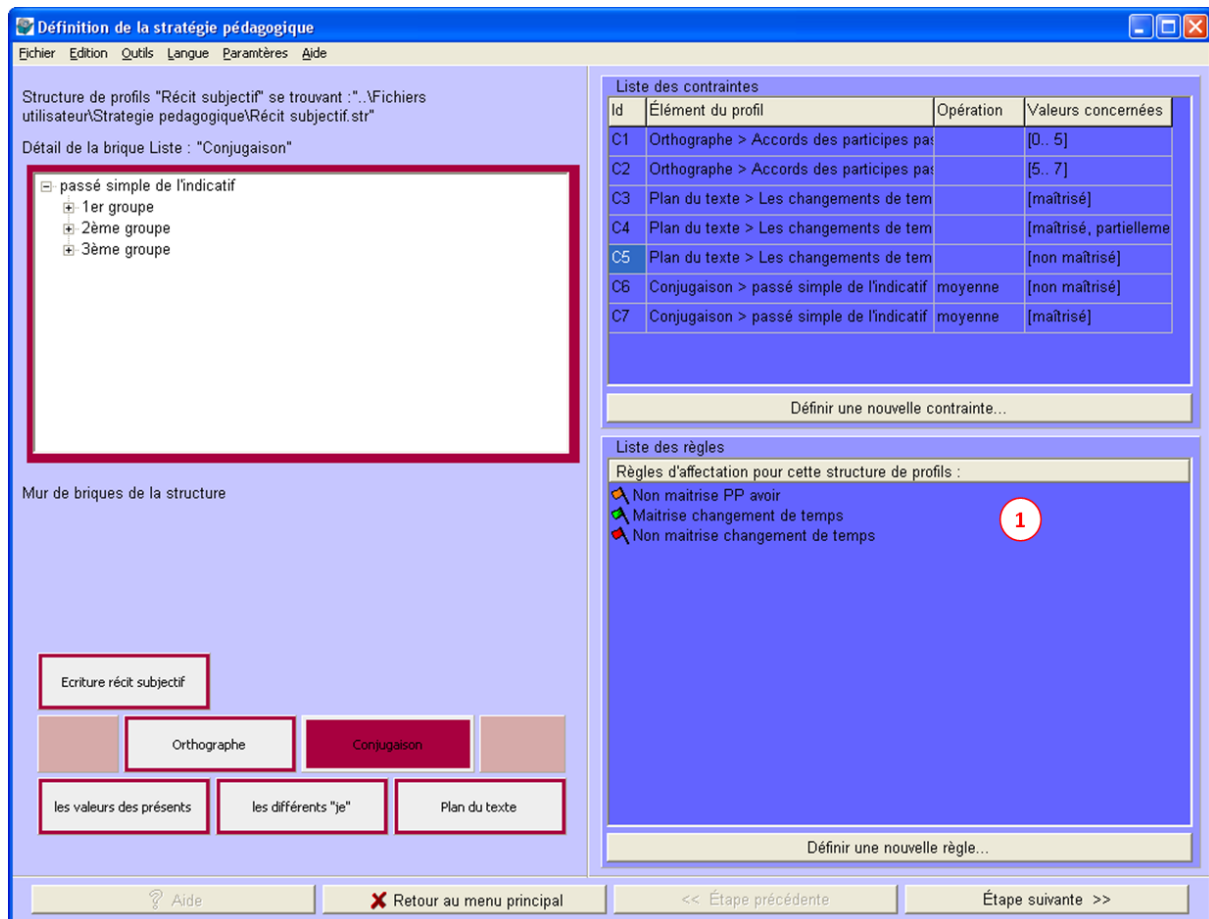


Figure 8-12 : Écran d'Adapte permettant de définir une stratégie pédagogique.

B. SPÉCIFICATION DU CONTEXTE D'UTILISATION D'UNE STRATÉGIE PÉDAGOGIQUE

Spécifier le contexte d'utilisation d'une stratégie pédagogique se fait en décrivant les contraintes d'utilisation des séquences de travail générées (durée de la séquence, nombre d'exercices, etc.) et en fournissant les profils des élèves concernés par les séquences de travail. Les différentes propriétés caractérisant un contexte d'utilisation ont été formalisées dans le modèle de personnalisation présenté dans la section 4.3.

Dans Adapte, la spécification du contexte d'utilisation est réalisée par l'utilisateur en trois étapes (cf. IIb sur la Figure 8-1, page 201). Tout d'abord, l'utilisateur fournit les profils des apprenants pour lesquels il souhaite obtenir des activités personnalisées. Ensuite, il définit des contraintes générales sur les feuilles d'exercices ou les sessions sur les EIAH. Enfin, il peut définir des exceptions concernant les séquences de travail de certains apprenants.

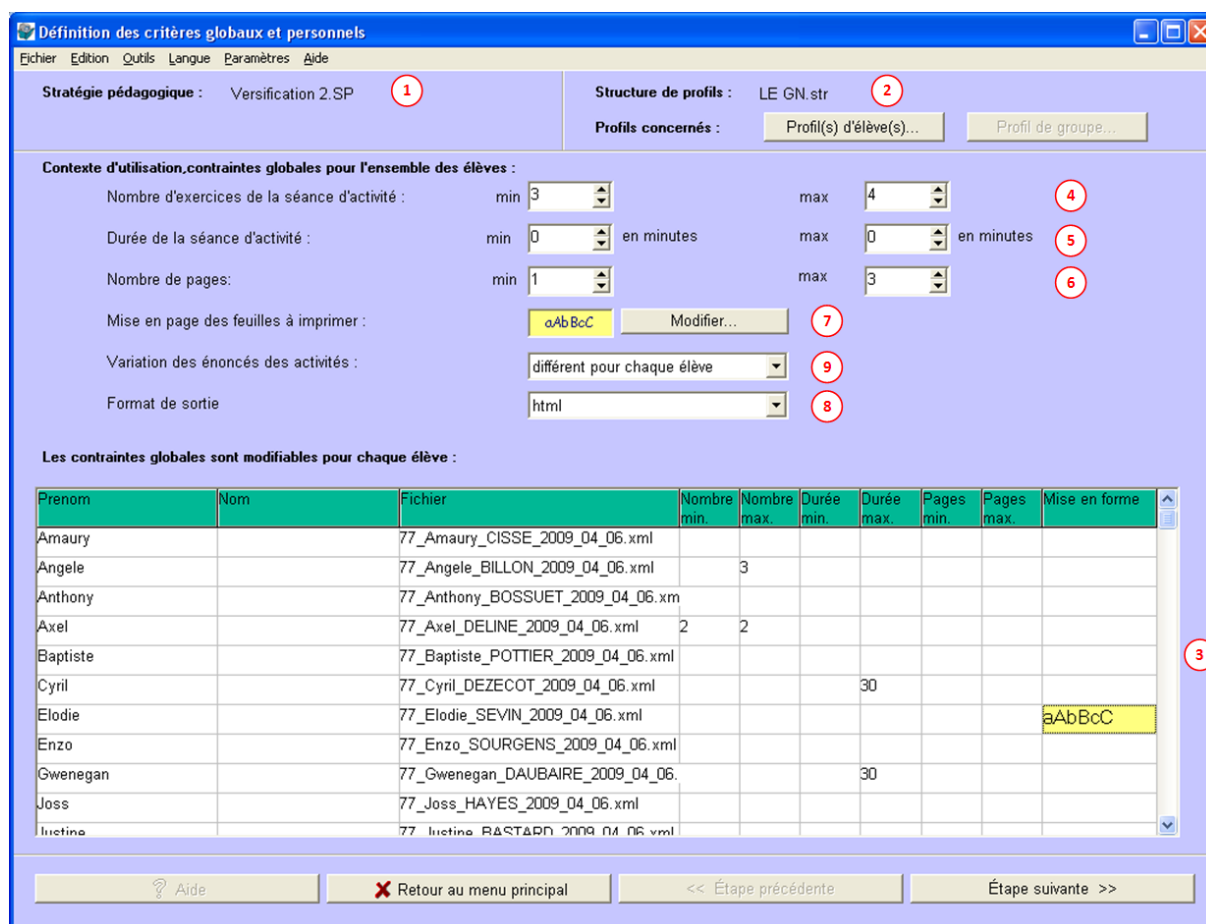


Figure 8-13 : Écran d'Adapte permettant de définir un contexte d'utilisation pour une stratégie pédagogique donnée.

La Figure 8-13 montre l'écran d'Adapte permettant de définir un contexte d'utilisation pour une stratégie pédagogique. Cette stratégie pédagogique (cf. ① sur la Figure 8-13) est elle-même définie pour une structure de profils donnée (cf. ② sur la Figure 8-13). La structure de profils contraint les profils d'élèves ou de groupes pouvant être choisis par l'utilisateur afin de ne pouvoir prendre en compte que des profils pertinents vis-à-vis de la stratégie pédagogique. Une fois les profils choisis (cf. ③ sur la Figure 8-13), l'utilisateur peut définir des contraintes générales sur la création des séquences d'activités. Il peut tout d'abord borner le nombre d'exercices contenus dans la feuille d'exercices papier ou dans la séance d'activités sur un EIAH (cf. ④ sur la Figure 8-13). Il peut de la même façon contraindre la durée de la séance de travail en lui attribuant une durée minimum et/ou une durée maximum (cf. ⑤ sur la Figure 8-13). Pour les feuilles d'exercices papier, il peut limiter le nombre de pages (cf. ⑥ sur la Figure 8-13). Cette dernière limitation dépend de la mise en page des feuilles à imprimer (style de la police d'écriture, taille, etc.) (cf. ⑦ sur la Figure 8-13) et du format de sortie des feuilles à imprimer (HTML, RTF...) (cf. ⑧ sur la Figure 8-13). Enfin, il peut spécifier que lors de la création des activités, papier ou logicielles, si plusieurs apprenants doivent avoir une activité issue de la même structure d'activités, les énoncés devront être identiques pour tous les élèves ou au contraire différents pour tous les élèves (cf. ⑨ sur la Figure 8-13). Une fois ces contraintes générales définies, l'utilisateur peut définir des exceptions pour certains de ses élèves. Pour cela, il précise dans le tableau contenant la liste des élèves ces exceptions (cf. ⑩ sur la Figure 8-13). Sur cet exemple, il demande ainsi à ce que les séquences de travail contiennent de trois à quatre activités pour tous les apprenants (cf. ④ sur la Figure 8-13), sauf pour Angele qui aura au plus trois activités, et pour Axel qui n'en aura que deux.

C. CRÉATION DE SÉQUENCES PERSONNALISÉES D'ACTIVITÉS

Créer une séquence personnalisée d'activités consiste à fournir un ensemble d'activités, qu'elles soient papier ou logicielles. La création des activités nécessaires se fait automatiquement par le système d'une part en fonction du profil des apprenants, de la stratégie pédagogique et du contexte d'utilisation définis par l'utilisateur et d'autre part, en s'appuyant sur les connaissances relatives à la création d'une feuille d'exercices papier et/ou à la création d'une session sur un EIAH. Une fois qu'Adapte a proposé pour chaque apprenant une séquence personnalisée d'activités, l'enseignant peut valider ou modifier les choix du système.

Pour les activités papier-crayon, Adapte fournit une feuille d'exercices propre au profil de chaque apprenant en générant les exercices contenus dans la feuille. Il fournit de plus, à l'utilisateur, la correction des exercices. Un exemple de séquence personnalisée d'activités papier proposée par Adapte est présenté sur la Figure 8-14.

Feuille d'exercices personnalisée pour Ophélie	Correction de la feuille d'exercices personnalisée pour Ophélie
<p>Exercice 1 :</p> <p>Conjugez les verbes suivants à l'indicatif présent et pour les 1ère, 2ème personnes du pluriel : Verbes : voir, jeter.</p> <p>*****</p>	<p>Correction 1 :</p> <p>voir : nous voyons, vous voyez jeter : nous jetons, vous jetez</p> <p>*****</p>
<p>Exercice 2 :</p> <p>Ce matin, Damien avait 42 fleurs. Il en a 32 maintenant. Damien a-t-il donné ou cueilli des fleurs ? Combien ?</p> <p>*****</p>	<p>Correction 2 :</p> <p>Damien a donné 10 fleurs</p> <p>*****</p>
<p>Exercice 3 :</p> <p>Ce matin, Romain avait des jonquilles. Il en a donné 14 à midi. Il en a 11 maintenant. Quel était le nombre de jonquilles de Romain au début ?</p> <p>*****</p>	<p>Correction 3 :</p> <p>Romain avait 25 jonquilles</p> <p>*****</p>
<p>Exercice 4 :</p> <p>Conjugez le verbe suivant au conditionnel présent pour la 3ème personne du singulier : Verbe : finir.</p> <p>*****</p>	<p>Correction 4 :</p> <p>finir : il finirait</p> <p>*****</p>

Figure 8-14 : Séquence personnalisée d'activités papier générée par Adapte.

Pour les activités logicielles, Adapte définit une session sur l'EIAH propre au profil de chaque apprenant. Les activités contenues dans la session sont issues de générateurs compris dans l'EIAH ou de bases de données de l'EIAH. Trois cas de figure se présentent :

- l'EIAH est paramétrable par des fichiers de configuration : Adapte gère ces fichiers ;
- l'EIAH est paramétrable uniquement par une interface enseignant ou administrateur : Adapte génère pour l'enseignant une feuille d'instructions indiquant les choix à effectuer sur l'interface ;
- l'EIAH est paramétrable uniquement par l'élève lors de son utilisation : Adapte génère une feuille d'instructions contenant la liste des exercices que l'apprenant doit effectuer.

Pour certains EIAH, ces cas peuvent se combiner. C'est le cas par exemple du logiciel ABALECT [Chevé 2005] pour lequel Adapte génère une liste d'instructions à suivre par l'apprenant (cf. fichiers HTML de la partie droite de la Figure 8-15, dont un est affiché dans la partie gauche) et des fichiers de configuration (cf. fichiers « .bil » de la partie droite de la Figure 8-15).

<p>Feuille d'exercices personnalisée pour Nadège B.</p> <p>Selectionner le menu Elève. Choisissez votre nom dans la liste. Pour le texte "Du cadran solaire à la montre atomique", faites les exercices suivants : "Texte masqué".</p>	B_Nadège.bil	0 Ko	Fichier BIL
	B_Mélina.bil	0 Ko	Fichier BIL
	G_Kévin.bil	0 Ko	Fichier BIL
	L_Lise.bil	0 Ko	Fichier BIL
	M_Ophélie.bil	0 Ko	Fichier BIL
	R_Marie.bil	0 Ko	Fichier BIL
	Versification_2009_04_09_B_Nadège.HTML	1 Ko	Firefox Document
	Versification_2009_04_09_B_Mélina.HTML	1 Ko	Firefox Document
	Versification_2009_04_09_G_Kévin.HTML	1 Ko	Firefox Document
	Versification_2009_04_09_L_Lise.HTML	1 Ko	Firefox Document

Figure 8-15 : Séquences personnalisées d'activités logicielles proposées par Adapte composées d'une liste d'instructions pour l'apprenant et d'un fichier de configuration pour ABALECT.

La Figure 8-16 montre l'écran d'Adapte permettant à l'utilisateur de valider, modifier ou supprimer les séquences personnalisées d'activités proposées. La modification d'une séquence d'activités peut se faire en ajoutant ou supprimant une activité donnée, en demandant un autre énoncé pour une activité donnée, ainsi qu'en changeant l'ordre des activités, soit manuellement, soit grâce à des fonctions de tris.

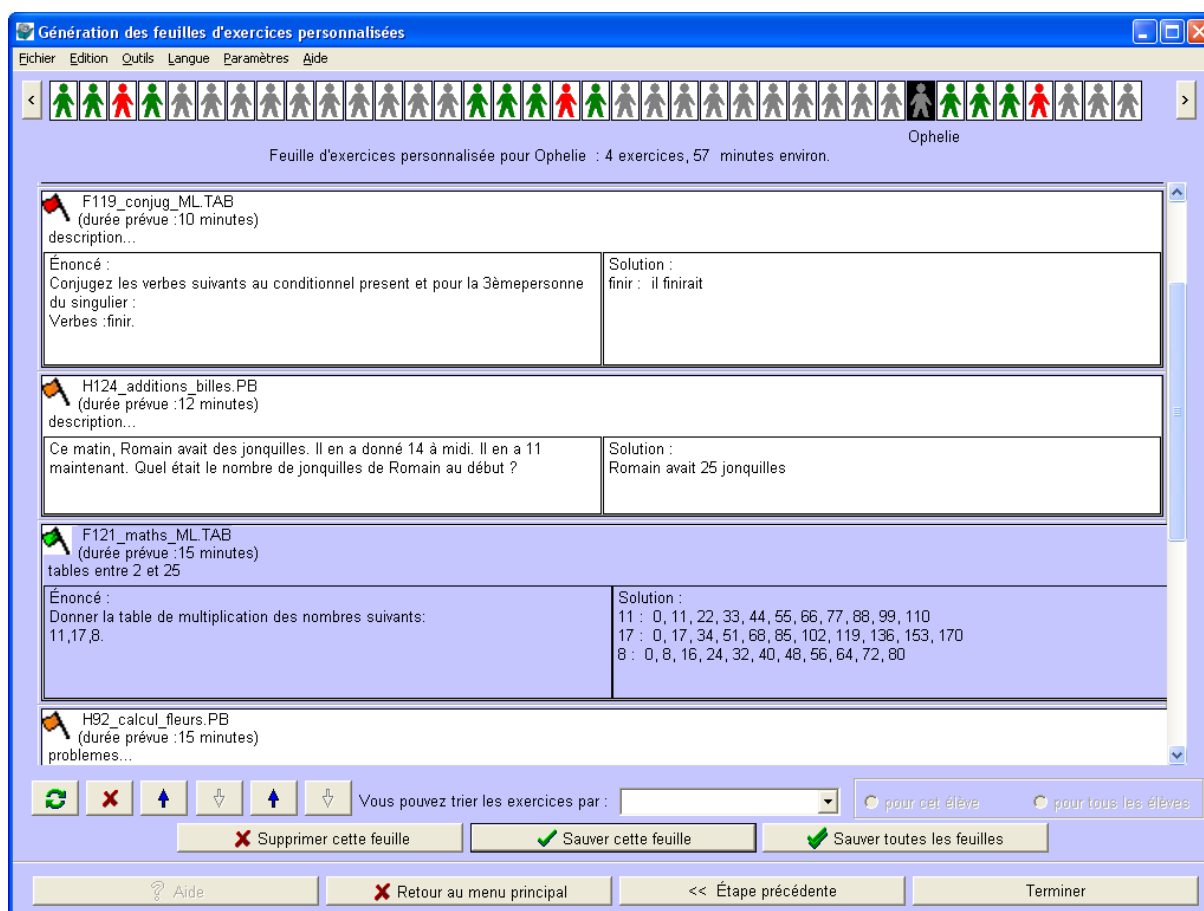


Figure 8-16 : Écran de validation des séquences de travail personnalisées dans Adapte.

8.2.3. CONVERSION D'ACTIVITÉS VERS UNE NORME PÉDAGOGIQUE

Lorsqu'Adapte génère des activités papier, nous avons fait le choix de faire les sorties dans un format spécifique pour ne pas dépendre d'une norme donnée. La conversion vers une norme pédagogique est une fonctionnalité optionnelle dans le logiciel Adapte qui permet de convertir les activités papier générées vers une norme pédagogique donnée (par exemple SCORM, LOM...) afin de permettre leur échange avec d'autres systèmes.

8.3. ARCHITECTURE D'ADAPTE

Dans la section précédente, nous avons présenté le fonctionnement du logiciel Adapte pour acquérir les connaissances nécessaires à la personnalisation des EIAH, via l'étape d'intégration d'un nouvel EIAH, pour obtenir des séquences de travail personnalisées dans son mode d'utilisation classique et pour convertir des activités vers une norme pédagogique. Pour permettre ces différentes étapes, le logiciel Adapte instrumente plusieurs processus : ceux permettant d'acquérir les connaissances nécessaires à la personnalisation des activités logicielles (cf. ① sur la Figure 8-17), ceux se servant, entre autres, de ces connaissances pour permettre la création de stratégies pédagogiques et de leur contexte d'utilisation (cf. ② sur la Figure 8-17), ceux utilisant les stratégies pédagogiques pour générer des séquences de travail personnalisées (cf. ③ sur la Figure 8-17) et celui permettant de convertir des activités papier dans une norme pédagogique donnée.

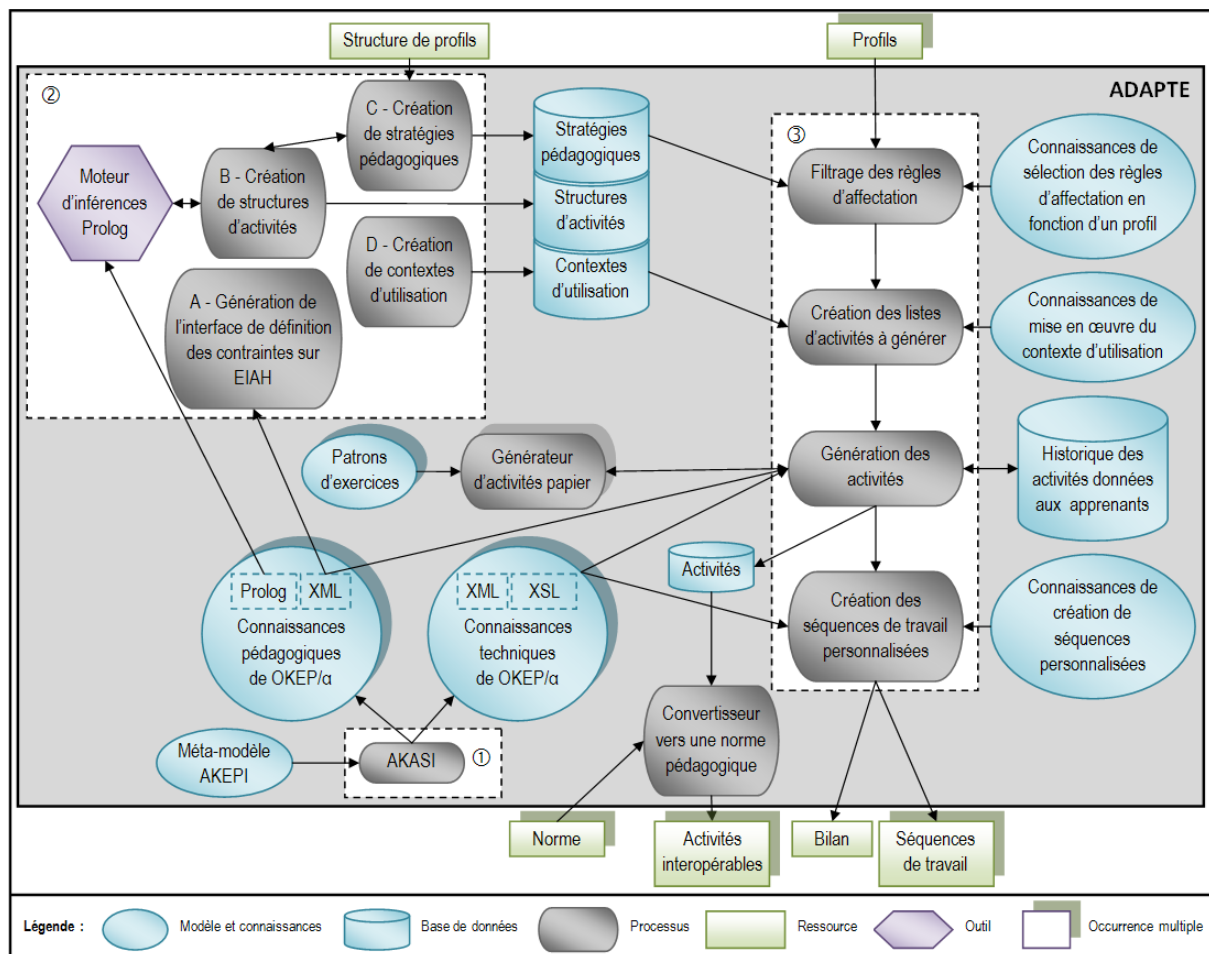


Figure 8-17 : Architecture du logiciel Adapte.

Nous détaillons à présent les processus de chacun des cadres en pointillés sur l'architecture du logiciel Adapte (cf. Figure 8-17) avant de faire le lien entre nos contributions théoriques et cette architecture.

8.3.1. PROCESSUS POUR L'INTÉGRATION D'UN NOUVEL EIAH

La partie « Intégration d'un nouvel EIAH » utilise le processus AKASI (cf. ① sur la Figure 8-17) présenté dans la section 6.4.3.c page 167 et dont les détails techniques sont fournis en Annexe J page 361. Ce processus permet de créer, pour chaque EIAH x à personnaliser, un modèle OKEP/ x . Chaque modèle OKEP/ x respecte le méta-modèle AKEPI (cf. section 6.4.3.b page 160), représenté par un XML Schema (fichier XSD). Un modèle OKEP/ x représente les connaissances pédagogiques et techniques relatives à la personnalisation d'un EIAH x . Les connaissances pédagogiques du modèle OKEP/ x regroupent l'ensemble des propriétés paramétrables de l'EIAH x et les règles permettant de contraindre l'utilisation de ces propriétés. Il est créé sous deux formes : un modèle XML et son équivalent Prolog. Les connaissances techniques du modèle OKEP/ x regroupent la description de l'ensemble des fichiers de configuration, des menus et des exécutable utiles pour paramétrer l'EIAH x . Ces connaissances sont traduites en feuilles de style XSL permettant de convertir les contraintes de personnalisation définies par l'utilisateur d'Adapte en configuration de l'EIAH x . L'ensemble des connaissances représentées dans un modèle OKEP/ x est fourni par le concepteur ou par un expert de l'EIAH x . Un exemple de modèle OKEP défini pour le logiciel ABALECT [Chevé 2005] est proposé en Annexe I, page 349.

8.3.2. PROCESSUS POUR LA CRÉATION DE SÉQUENCES DE TRAVAIL PERSONNALISÉES

La partie « Création de séquences de travail personnalisées » se décompose en deux étapes : la définition par l'utilisateur de contraintes pour personnaliser les séquences de travail (stratégies pédagogiques et contexte d'utilisation) (cf. ② sur la Figure 8-17) et la génération par Adapte de ces séquences de travail (cf. ③ sur la Figure 8-17).

Pour permettre à l'utilisateur de définir ses contraintes pour personnaliser les séquences de travail, le logiciel Adapte s'appuie sur le modèle de personnalisation proposé dans la section 4.3 page 84. Adapte possède pour cela plusieurs interfaces permettant de contraindre les profils, de contraindre les activités (papier ou logicielles), de faire le lien entre les deux types de contraintes afin de créer une stratégie pédagogique, et enfin de définir un contexte d'utilisation pour une stratégie pédagogique donnée. Ces interfaces sont toutes prédéfinies, à l'exception de celle permettant de contraindre les activités logicielles. En effet, pour les activités papier, l'interface proposée est prédéfinie et se compose de quinze écrans (cf. un exemple d'écran sur la Figure 8-6), chaque écran étant associé à un des patrons opérationnels d'exercices de l'approche GEPPETO_P (cf. section 6.3.3.c page 131 et Annexe F page 307), tandis que pour les activités logicielles, l'interface est générée automatiquement en fonction de l'EIAH à personnaliser. Adapte utilise pour cela le processus 2-A de la Figure 8-17, qui se sert de la description des paramètres utiles à la personnalisation de l'EIAH définis dans le modèle OKEP. Cette description, stockée sous forme d'un fichier XML, contient le nom des paramètres, mais également les différentes valeurs qu'ils peuvent prendre et la façon de les regrouper (en utilisant la catégorie, voire les sous-catégories auxquelles sont reliés les paramètres). Pour mettre à jour l'interface à chaque action de l'utilisateur (choix d'une valeur pour un paramètre), Adapte utilise un moteur d'inférence Prolog. Ce moteur prend en entrée d'une part la liste des paramètres définis dans le modèle OKEP, la valeur par défaut ou la valeur fixée par l'utilisateur de chaque paramètre, et le

fait qu'un paramètre doit être accessible ou non à l'utilisateur, et d'autre part, les règles contraignant les propriétés du modèle OKEP.

L'ensemble des contraintes faites par l'utilisateur sur une activité est appelé *structure d'activités*. Adapte sauvegarde les structures d'activités dans des fichiers XML (cf. processus 2-B de la Figure 8-17). Chaque structure d'activités contient deux parties : une description générale à l'aide de métadonnées et la liste des contraintes permettant de créer des activités. Un exemple de fichier XML contenant une structure d'activités a été proposé Figure 6-11 page 135. Les métadonnées relatives aux structures d'activités sont décrites Figure 8-18 avec, pour chacune, l'entité qui remplit la métadonnée et son caractère optionnel ou obligatoire.

Métadonnées	Contenu	Entité	Obligatoire
Id	Référence unique.	Système	X
Createur	Personne qui a défini les contraintes présentes dans la structure.	Système	X
Date_creation	Date à laquelle la structure a été créée.	Système	X
Date_modification	Date de la dernière modification de la structure.	Système	X
Langue	Langue de création de la structure et donc des activités issues de la structure.	Système	X
Support	Indique si la structure permet de créer des activités papier ou logicielles.	Système	X
Statut	Précise le caractère public ou privé de la structure.	Utilisateur	X (par défaut <i>public</i>)
Nom	Nom donné à la structure par son créateur.	Utilisateur	X
Niveau	Niveau d'études auquel les activités issues de la structure sont destinées.	Utilisateur	X
Durée	Temps estimé de travail sur une activité issue de la structure par les apprenants.	Utilisateur	X
Discipline	Discipline à laquelle les activités issues de la structure sont destinées, par exemple Mathématiques.	Utilisateur	X
Matière	Précision de la discipline à laquelle les activités sont destinées, par exemple Addition.	Utilisateur	
Descriptif	Descriptif des contraintes contenues dans la structure et/ou du type d'activités créées à partir de cette structure.	Utilisateur	
Objectifs	Objectifs pédagogiques visés par la structure d'activités.	Utilisateur	
Mots_clefs	Liste de mots permettant de retrouver la structure d'activités lors d'une prochaine utilisation.	Utilisateur	

Figure 8-18 : Métadonnées d'une structure d'activités.

Les structures d'activités sont ensuite utilisées pour créer des stratégies pédagogiques. Ainsi, le processus 2-C de la Figure 8-17 utilise d'une part la structure de profils des apprenants, fournie dans un fichier XML, pour permettre à l'utilisateur de choisir les parties du profil pour lesquelles des activités vont être attribuées aux apprenants, et d'autre part les structures d'activités spécifiant les activités à attribuer.

Les stratégies pédagogiques sont également stockées dans des fichiers XML. Chacune contient trois parties : l'ensemble des métadonnées décrivant la stratégie pédagogique, la liste des critères permettant de sélectionner des parties du profil des apprenants et la liste des règles d'affectation des activités aux apprenants. Les métadonnées relatives aux structures d'activités sont décrites Figure 8-19 avec, pour chacune, l'entité qui remplit la métadonnée et son caractère optionnel ou obligatoire.

Métadonnées	Contenu	Entité	Obligatoire
Id	Référence unique.	Système	X
Createur	Personne qui a défini la stratégie pédagogique.	Système	X
Date_creation	Date à laquelle la stratégie a été créée.	Système	X
Date_modification	Date de la dernière modification de la stratégie.	Système	X
Statut	Précise le caractère public ou privé de la stratégie.	Système	X (par défaut <i>public</i>)
Structure_profils	Structure de profils associée à la stratégie pédagogique.	Système	X
Nom	Nom donné à la stratégie pédagogique par son créateur.	Utilisateur	X
Date_utilisation	Date d'utilisation prévue par le créateur, par exemple premier semestre.	Utilisateur	
Niveau	Niveau d'études des apprenants.	Utilisateur	
Discipline	Discipline travaillée	Utilisateur	
Descriptif	Texte décrivant le contenu de la stratégie pédagogique.	Utilisateur	
Objectifs	Objectifs pédagogiques visés en créant la stratégie.	Utilisateur	
Mots_clefs	Liste de mots permettant de retrouver la stratégie lors d'une prochaine utilisation.	Utilisateur	

Figure 8-19 : Métadonnées d'une stratégie pédagogique.

Pour pouvoir être utilisées par Adapte, les stratégies pédagogiques sont associées à des contextes d'utilisation qui sont eux-aussi stockés dans des fichiers XML. Comme les précédents, ces fichiers contiennent un ensemble de métadonnées dont la description est fournie Figure 8-20.

Métadonnées	Contenu	Entité	Obligatoire
Id	Référence unique.	Système	X
Createur	Personne qui a défini les contraintes du contexte d'utilisation.	Système	X
Date_creation	Date à laquelle le contexte a été défini.	Système	X
Date_modification	Date de la dernière modification du contexte.	Système	X
Statut	Précise le caractère public ou privé du contexte.	Système	X (par défaut <i>public</i>)
Strategie_pedagogique	Stratégie pédagogique associée au contexte d'utilisation.	Système	X
Nom	Nom donné au contexte par son créateur.	Utilisateur	X
Public	Public visé par le contexte, par exemple élève de CE1.	Utilisateur	
Descriptif	Descriptif du contexte d'utilisation.	Utilisateur	
Objectifs	Objectifs pédagogiques visés par la mise en œuvre de la stratégie pédagogique dans le contexte créé.	Utilisateur	
Mots_clefs	Liste de mots permettant de retrouver le contexte lors d'une prochaine utilisation.	Utilisateur	

Figure 8-20 : Métadonnées d'un contexte d'utilisation.

Pour générer des séquences de travail personnalisées, Adapte utilise le processus décomposé dans le ③ de la Figure 8-17. Ce processus est décrit dans la section 7.3 page 187, nous ne redétaillons donc pas ses différentes étapes ici. Pour rappel, ce processus prend en entrée n profils d'apprenants, chacun représenté dans un fichier XML, et fournit en sortie n séquences de travail personnalisées, pour les activités papier, leurs corrections et un bilan pour l'enseignant. Les séquences de travail sur papier sont des feuilles d'exercices à imprimer, qui sont actuellement disponibles en format RTF ou HTML. Les séquences de travail sur les EIAH comprennent les dossiers et fichiers de configuration nécessaires à la paramétrisation de celui-ci, et éventuellement une suite d'instructions à imprimer que l'enseignant ou l'apprenant devra suivre pour paramétrer l'EIAH. Le bilan pour l'enseignant est présenté à l'interface.

Ce processus de génération des séquences de travail personnalisées utilise plusieurs bases de connaissances :

- les patrons d'exercices papier : présentés dans la section 6.3.3.c page 131 et en Annexe F page 307, ils sont définis en XML Schema (XSD). Ces modèles sont purement théoriques, ils ne servent pas concrètement dans l'utilisation du logiciel, mais ont servi à implémenter le logiciel. Ils contiennent les différents éléments d'une structure d'exercices associée à un patron ;
- les modèles OKEP : présentés dans la section 6.4.3.d page 168, ils sont définis en XML en respectant le méta-modèle AKEPI (cf. section 6.4.3.b page 160). Dans le but de convertir les contraintes des utilisateurs, sauvegardées dans un fichier XML, en fichier de configuration ou en liste d'instructions, les connaissances techniques des modèles OKEP sont traduites en feuilles de style XSL ;

- les connaissances de sélection des règles d'affectation en fonction d'un profil d'apprenant : présentées dans la section 7.3.1 page 188 , ces connaissances sont incluses dans le code du logiciel Adapte ;
- les connaissances de mise en œuvre du contexte pédagogique : présentées dans la section 7.3.2 page 189, ces connaissances sont incluses dans le code du logiciel Adapte ;
- les connaissances de création de séquences personnalisées : présentées dans la section 7.3.4 page 190, ces connaissances sont incluses dans le code du logiciel Adapte.

Ce processus utilise également un historique des activités fournies aux apprenants, sous forme de fichiers XML, permettant au système de ne pas fournir deux fois la même activité à un apprenant.

8.3.3. PLACE DE NOS CONTRIBUTIONS THÉORIQUES DANS ADAPTE

Dans la partie 2 de ce manuscrit, nous avons détaillé les contributions théoriques faites pendant cette thèse. Ces contributions ont été mises en œuvre dans le logiciel Adapte. La Figure 8-21 montre dans quelle mesure elles sont intégrées dans l'architecture du logiciel.

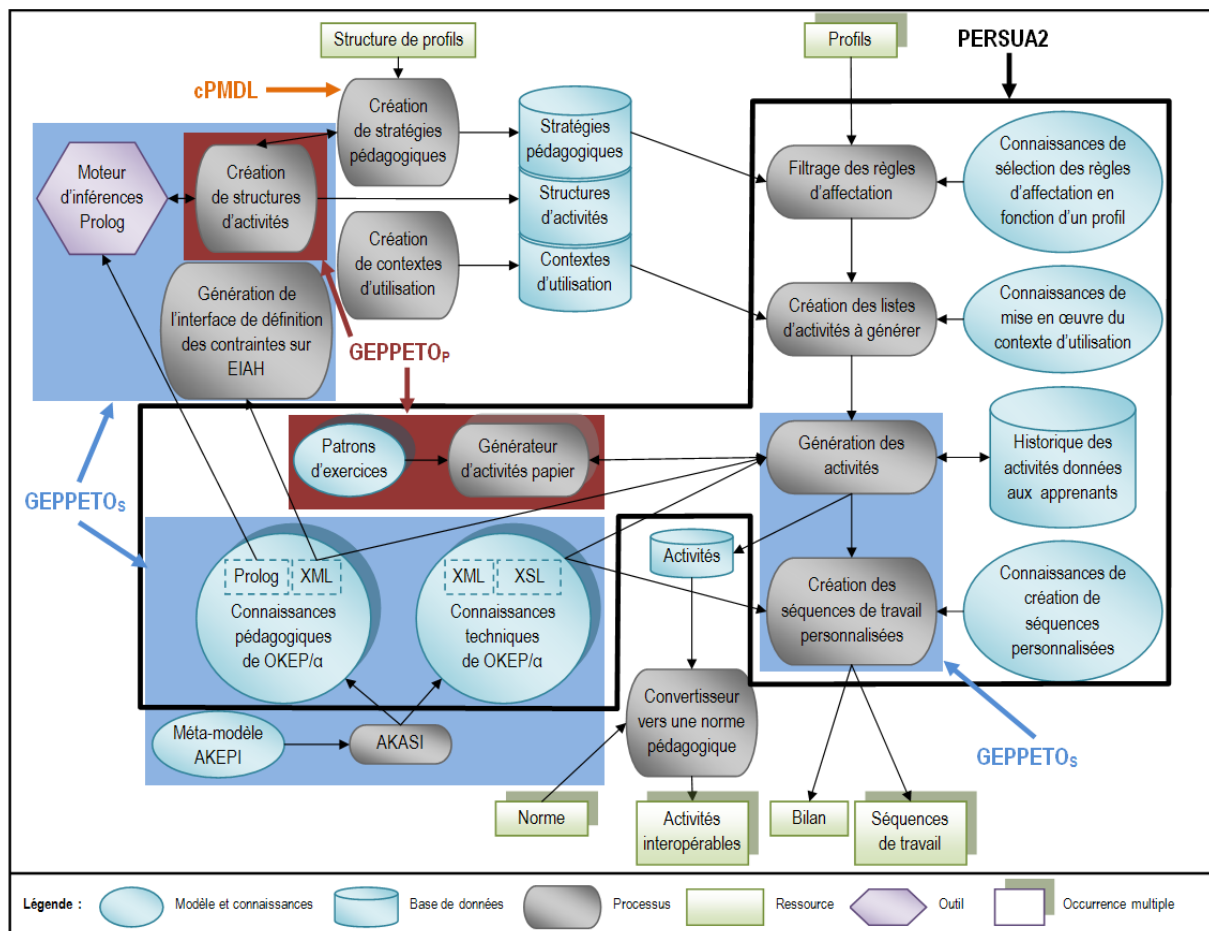


Figure 8-21 : Lien entre l'architecture d'Adapte et nos contributions théoriques.

Notre première contribution concerne le modèle PERSUA2 pour la personnalisation unifiée des activités proposées aux apprenants (cf. section 4.3 page 84). Ce modèle et son processus d'exploitation sont entourés en trait plein.

Notre seconde contribution concerne le modèle cPMDL pour contraindre un profil. Ce modèle est utilisé lors de la création des stratégies pédagogiques et est indiqué par une flèche en haut à gauche de la figure.

Notre troisième proposition concerne l'approche GEPPETO permettant d'adapter la création des activités aux buts pédagogiques des enseignants (cf. section 6.2 page 117). Cette approche a deux versants. Les modèles et processus de l'approche GEPPETO_p, pour les activités papier et de l'approche GEPPETO_s, pour les EIAH, sont entourés par des rectangles unis.

8.4. MODES D'UTILISATION D'ADAPTE

Nous venons de présenter le fonctionnement du logiciel Adapte et les principaux éléments de son architecture. Nous présentons maintenant les différents modes d'utilisation envisagés pour ce logiciel en précisant à chaque fois les possibilités offertes par le logiciel, les acteurs intervenant et selon quelle temporalité.

8.4.1. PERSONNALISATION DES ACTIVITÉS : AVEC INTERVENTION DE L'ENSEIGNANT ET EXPLOITATION DE PROFILS D'APPRENANTS

La personnalisation des séquences de travail à partir du profil des apprenants et en faisant intervenir l'enseignant est le mode d'utilisation pour lequel le logiciel Adapte a été imaginé initialement. Lorsque nous avons décrit le fonctionnement du logiciel dans la section précédente, nous l'avons donc fait en nous appuyant sur ce mode d'utilisation.

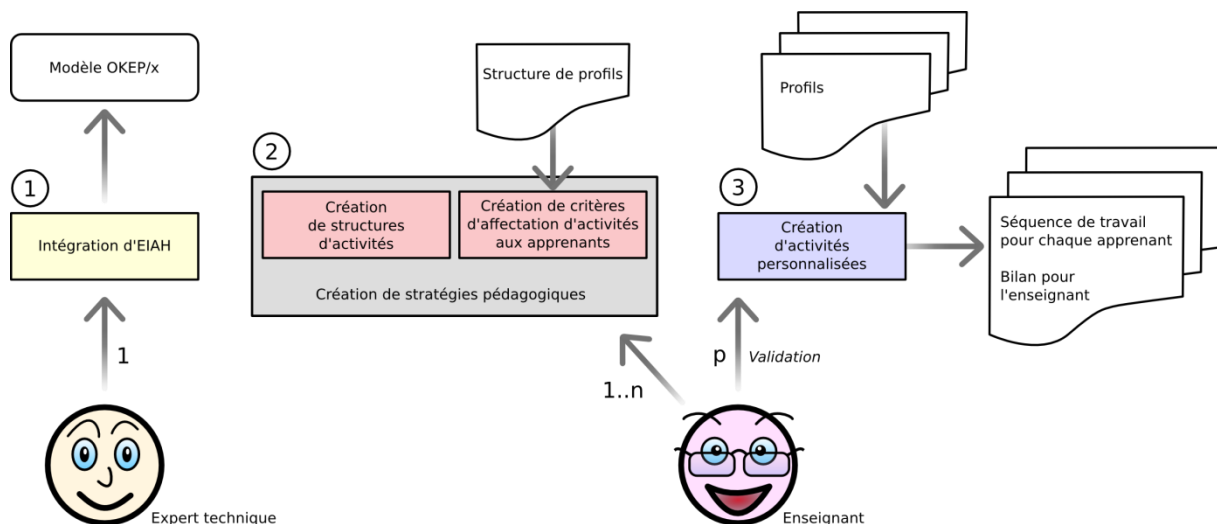


Figure 8-22 : Utilisation d'Adapte par l'enseignant avec des profils d'apprenants.

Dans ce mode d'utilisation, deux types d'acteurs interviennent (cf. Figure 8-22) : un expert (correspondant informatique de l'institution scolaire utilisant Adapte, concepteur d'Adapte ou concepteur du nouvel EIAH à personnaliser) et des enseignants.

Lors de l'initialisation du logiciel Adapte, l'expert intègre les connaissances sur les EIAH utilisés par les enseignants (cf. ① sur la Figure 8-22). Ensuite, chaque enseignant peut se servir librement du logiciel. Pour cela, il définit, réutilise ou modifie des stratégies pédagogiques et leurs contextes d'utilisation (cf. ② sur la Figure 8-22). Le système propose ensuite une séquence

de travail personnalisée pour chaque profil d'apprenant, séquence que l'enseignant peut ensuite modifier, valider ou supprimer (cf. ③ sur la Figure 8-22).

8.4.2. PERSONNALISATION DES ACTIVITÉS : SANS INTERVENTION DE L'ENSEIGNANT ET EXPLOITATION DE PROFILS D'APPRENANTS

La personnalisation automatique des séquences de travail à partir du profil des apprenants ne fait pas intervenir l'enseignant. Ce mode d'utilisation est prévu, mais n'est actuellement pas mis en œuvre dans Adapte. Il a pour but de récupérer les profils issus de la partie préparation d'EPROFILEA (cf. Figure 0-1 page 25) et de créer sans intervention humaine des séquences de travail adaptées à ces profils. Une fois que les apprenants auront effectué ces activités, leurs profils seront mis à jour et Adapte pourra proposer de nouvelles activités.

Les modèles proposés dans le cadre de cette thèse l'ont été en prenant en compte ce mode d'utilisation. Pour rendre possible sa mise en œuvre, il faut compléter l'environnement EPROFILEA en lui ajoutant un « méta-module » qui récupèrera les profils des apprenants, les mettra à jour grâce aux modules de la partie préparation et lancera Adapte avec les stratégies pédagogiques et le contexte d'utilisation adéquats.

Ce mode d'utilisation est prévu pour permettre à Adapte de jouer le rôle du module pédagogique de n'importe quel EIAH pouvant être décrit selon le méta-modèle AKEPI. En effet, cette boucle automatique n'a de sens que si les profils des apprenants sont mis à jour automatiquement. Or ce n'est pas le cas des profils remplis à partir des résultats des apprenants travaillant sur des activités papier. Ce mode d'utilisation est donc réservé à la personnalisation de logiciels pédagogiques possédant un profil d'apprenant pouvant être exporté.

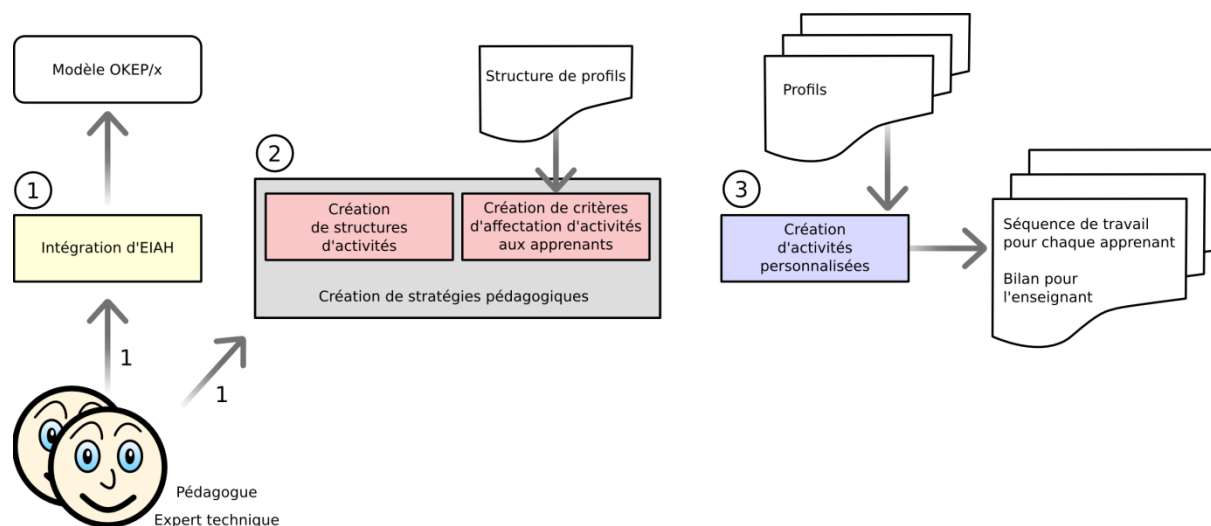


Figure 8-23 : Utilisation d'Adapte en boucle automatique à partir des profils d'apprenants.

Dans ce mode d'utilisation, deux types d'acteurs interviennent pour chaque EIAH à personnaliser (cf. Figure 8-23) : un expert (correspondant informatique de l'institution scolaire utilisant Adapte, concepteur d'Adapte ou concepteur du nouvel EIAH à personnaliser) et un pédagogue (par exemple, concepteur pédagogique de l'EIAH).

Lors de l'initialisation du logiciel Adapte, l'expert intègre les connaissances sur l'EIAH à personnaliser (cf. ① sur la Figure 8-23). Ensuite, le pédagogue définit une stratégie pédagogique et un contexte d'utilisation pour cet EIAH (cf. ② sur la Figure 8-23). Le système propose ensuite une séquence de travail personnalisée pour chaque profil d'apprenant (cf. ③ sur la Figure 8-23).

8.4.3. PERSONNALISATION DES ACTIVITÉS : AVEC INTERVENTION DE L'ENSEIGNANT SANS EXPLOITATION DE PROFILS D'APPRENANTS

La personnalisation des séquences de travail sans profil d'apprenant en faisant intervenir l'enseignant est une utilisation détournée du logiciel Adapte [Perriault 1989, Rabardel 1995]. Ce mode d'utilisation n'était pas prévu au départ. Il a été imaginé par les enseignants lors des différentes réunions de travail.

Dans ce mode d'utilisation, l'enseignant définit ses stratégies pédagogiques sans utiliser de structure de profils. Chaque règle d'affectation des activités aux apprenants a donc le format « Pour tous les apprenants, donner les activités correspondant aux structures d'activités $\text{StrAct}_1 \dots \text{StrAct}_n$ ». Seul le contexte d'utilisation va déterminer si les séquences proposées sont destinées à un apprenant particulier ou non. Ainsi, dans le contexte d'utilisation, l'enseignant a le choix entre demander une séquence de travail sans préciser son destinataire ou demander une séquence de travail pour des apprenants, mais sans fournir de profils. Dans le premier cas, Adapte crée une unique séquence qui peut par exemple servir d'énoncé pour une séance de travaux dirigés ou un examen. Dans le second cas, Adapte crée autant de séquences de travail qu'il y a d'apprenants. Ces séquences seront toutes identiques, sauf si le contexte d'utilisation contient des exceptions pour certains apprenants (par exemple, trois activités pour Manon, une pour Lucas et deux pour tous les autres) ou si le contexte d'utilisation est plus restreint ou au contraire plus large que la stratégie pédagogique (par exemple, la stratégie pédagogique indique de donner dix activités d'un même niveau de priorité et le contexte d'utilisation borne les séquences de travail à trois activités, dans ce cas, les activités de chaque séquence seront générées à partir de trois structures d'activités choisies aléatoirement parmi les dix).

Ce mode d'utilisation permet de créer des séquences d'activités contenant aussi bien des activités papier que des activités logicielles. Dans le cas où la séquence se compose uniquement d'activités papier, Adapte équivaut à un groupement de générateurs d'exercices. Dans le cas où la séquence se compose uniquement d'activités logicielles sur un unique EIAH, Adapte correspond à une interface de paramétrage de l'EIAH comme en possèdent par exemple les logiciels MATHENPOCHE [Mathenpoche 2002] ou AMBRE-ADD [Duclosson et al. 2005a].

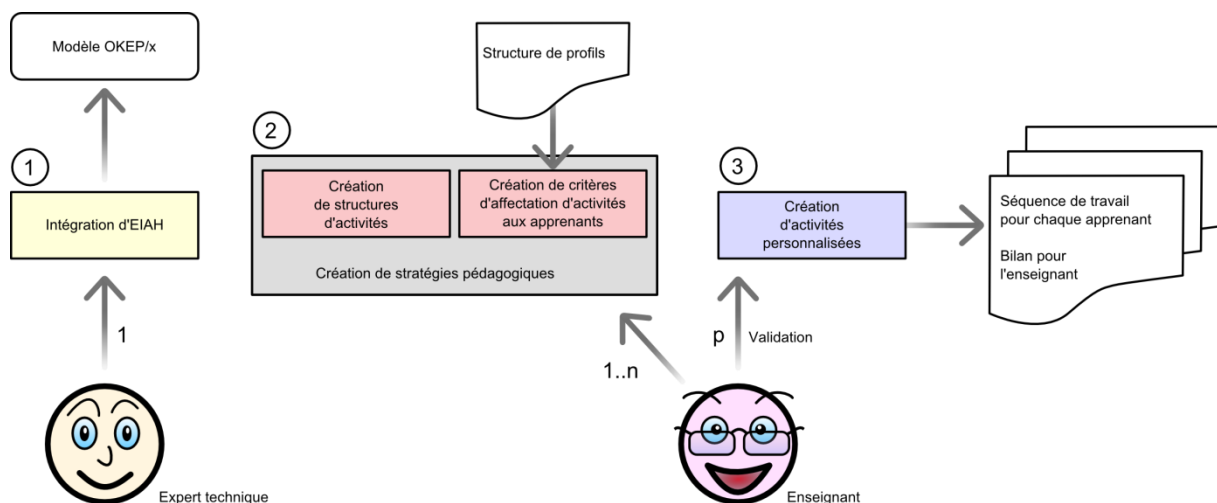


Figure 8-24 : Utilisation d'Adapte par l'enseignant sans profils d'apprenants.

Dans ce mode d'utilisation, deux types d'acteurs interviennent (cf. Figure 8-24) : un expert (correspondant informatique de l'institution scolaire utilisant Adapte, concepteur d'Adapte ou concepteur du nouvel EIAH à personnaliser) et des enseignants.

Lors de l'initialisation du logiciel Adapte, l'expert intègre les connaissances sur les EIAH utilisés par les enseignants (cf. ① sur la Figure 8-22). Ensuite, chaque enseignant peut se servir librement du logiciel. Pour cela, il définit, réutilise ou modifie des stratégies pédagogiques et leurs contextes d'utilisation (cf. ② sur la Figure 8-22). Le système propose ensuite soit une unique séquence de travail, soit autant de séquences de travail qu'il y a d'apprenants dans la liste d'apprenants du contexte pédagogique (cf. ③ sur la Figure 8-22). L'enseignant a toujours la possibilité de modifier, valider ou supprimer la ou les séquences de travail.

8.5. RETOUR SUR LES SCÉNARIOS D'USAGE

Nous avons présenté en introduction de ce manuscrit six scénarios montrant les enjeux de la personnalisation des activités pédagogiques dans les pratiques des enseignants (cf. page 27). À la fin de notre état de l'art, nous avons montré les limites actuelles de leur mise en œuvre (cf. section 3.1 page 73). Nous revenons à présent sur ces scénarios pour montrer en quoi nos contributions théoriques et l'implémentation qui a en été faite dans le logiciel Adapte peuvent améliorer ces situations.

SCÉNARIO 1 : AIDE PERSONNALISÉE À L'ÉCOLE PRIMAIRE

Alphonse, enseignant de CE2, encadre des séances de soutien pour les élèves en difficulté de son établissement. Pour préparer le contenu de ces séances de soutien, chaque enseignant fixe les objectifs à atteindre par ses élèves et fournit à ses collègues la liste des exercices à faire, liste créée à partir d'exercices issus des livres qu'il utilise. Alphonse souhaiterait s'appuyer sur toutes les informations dont ses collègues et lui disposent sur les connaissances des élèves (données issues des évaluations nationales, profils d'élèves créés par le logiciel AMBRE-ADD et connaissances qu'ils ont des élèves) pour proposer des séances de remédiation soit sous forme papier-crayon, soit avec des logiciels d'apprentissage adaptées à chacun des élèves.

Avec l'environnement EPROFILEA (cf. Figure 0-1 page 25), Alphonse et ses collègues peuvent récupérer la structure de profils correspondant aux évaluations nationales dans Bâtitseur et la fusionner avec celle correspondant au profil du logiciel AMBRE-ADD dans le module Groupe. Ils peuvent ainsi compléter l'unique structure de profils avec les données de tous les apprenants en utilisant le module Tornade. Au final, les enseignants disposeront d'un profil par élève contenant toutes les informations issues des évaluations nationales et du logiciel AMBRE-ADD. Chaque enseignant peut ensuite compléter la structure de profils pour ajouter les informations dont il a besoin puis compléter le profil de ses élèves dans Prose.

À partir de la structure de profils commune à tous les enseignants, chaque enseignant peut définir dans Adapte sa stratégie pédagogique pour le soutien permettant des créer des activités uniquement sur papier, uniquement sur un ou plusieurs logiciels pédagogiques qu'il connaît ou combinant les deux supports.

L'enseignant encadrant la séance de soutien, ici Alphonse, pourra ainsi, pour chaque élève présent, obtenir une séquence de travail adaptée aux buts pédagogiques de l'enseignant habituel de l'élève. Pour cela, Alphonse récupère la stratégie pédagogique de l'enseignant habituel de l'élève, définit le contexte d'utilisation propre à la séance de soutien (temps disponible, matériel et logiciels disponibles) et demande la création d'une séance de travail adapté au profil de l'élève. Alphonse peut faire cela pour les six élèves de sa séance.

Avec l'environnement EPROFILEA et le module Adapte, Alphonse et ses collègues peuvent mutualiser les informations dont ils disposent sur leurs élèves (informations issues des cours

classiques ou des séances de soutien) et obtenir rapidement des séances de remédiation adaptées à chacun des élèves. Ils gagnent le temps de traitement de chaque profil puisqu'ils ne le font qu'une seule fois lors de la définition de la stratégie pédagogique. Le traitement des profils est ensuite fait à chaque fois par Adapte.

SCÉNARIO 2 : COMBINER REMÉDIATION ET APPROFONDISSEMENT

Fred, professeur d'anglais au collège, souhaite, avec chacune de ses classes, faire une séance d'une heure en salle informatique durant laquelle ses élèves travailleraient une demi-heure sur des logiciels pédagogiques et une demi-heure avec lui en remédiation ou en approfondissement. Durant ces séances, il souhaite que les logiciels qu'il fait utiliser à ses élèves, PLANET ENGLISH [English], VOCABONE [VocabOne 2007] et SEPHONICS [Sephonics 2008] s'adaptent automatiquement aux besoins des élèves, ce qui n'est actuellement pas le cas.

Avec EPROFILEA (cf. Figure 0-1 page 25), Fred peut définir une structure de profils dans Bâtisseur, puis la compléter dans Prose avec ses propres données afin de créer un profil par élève. Ensuite, dans Adapte, il peut définir une stratégie pédagogique indiquant que les élèves ayant des difficultés de prononciation doivent travailler sur le logiciel SEPHONICS [Sephonics 2008], tandis que les autres doivent apprendre du nouveau vocabulaire grâce aux logiciels PLANET ENGLISH et VOCABONE. Pour obtenir des séquences de travail d'une demi-heure, il définit son contexte d'utilisation. À partir du profil des élèves, Adapte peut ainsi lui proposer une séquence de travail sur un ou deux des logiciels pédagogiques adaptée à chaque élève.

En utilisant l'environnement EPROFILEA et plus particulièrement le module Adapte, Fred peut créer des profils pour ses élèves afin d'obtenir des séquences de travail adaptées. L'avantage majeur est que Fred n'aura pas à maîtriser trois outils mais un seul pour personnaliser les trois logiciels pédagogiques.

SCÉNARIO 3 : FORMATION À DISTANCE PERSONNALISÉE

Amélie, professeur de français au collège, donne des cours à distance pour une vingtaine d'élèves ne pouvant être scolarisés. Elle souhaite construire un profil pour chaque élève avec les difficultés identifiées lors de la correction des rédactions qu'elle leur demande de faire. Elle se servirait de ces profils pour faire travailler les élèves en langue sur des exercices ayant comme support leur propre rédaction, mais avec des énoncés adaptés à leur profil.

Amélie peut, dans la première partie de l'environnement EPROFILEA (cf. Figure 0-1 page 25), construire un profil pour chacun de ses élèves. Elle peut ensuite dans Adapte définir une stratégie pédagogique utilisant le générateur d'exercices papier « Questions / réponses » pour faire travailler les élèves sur un document annexe qui se trouve être leur propre rédaction. En demandant de fournir les séquences d'activités au format HTML, elle pourra ainsi mettre à disposition les énoncés à chacun de ses élèves via la plateforme de formation.

Avec le module Adapte, Amélie n'aura plus à traiter manuellement les profils de ses vingt élèves afin de créer les exercices mais seulement à définir une seule fois sa stratégie pédagogique, puis à mettre à jour les profils avant chaque utilisation d'Adapte.

SCÉNARIO 4 : APPRENDRE À APPRENDRE

Manuela, professeur de biologie au collège, participe à l'encadrement de l'étude du soir pour les élèves de 6^{ème} et 5^{ème}. Durant cette étude, elle reprend avec les élèves des exercices déjà faits dans des disciplines variées, afin de leur donner des conseils méthodologiques. Elle souhaiterait

travailler non plus sur des exercices déjà faits par les élèves, mais sur des exercices ayant des énoncés équivalents.

Si chaque enseignant de son collège utilise Adapte pour créer ses feuilles d'exercices, Manuela pourra demander un nouvel énoncé des exercices proposées par les enseignants pour l'étude du soir. Cette utilisation d'Adapte se fera dans le mode sans profil, chaque enseignant se servant d'Adapte comme d'un générateur d'exercices papier-crayon.

SCÉNARIO 5 : GROUPE DE RÉVISION POUR LE BREVET DES COLLÈGES

Louise et Léon sont professeurs de français au collège. Ils possèdent, comme leurs collègues de mathématiques, un profil papier pour chacun de leurs élèves. Dans leur établissement, les élèves de 3^{ème} participent à des séances de révisions pour le brevet des collèges. Il y a trois séances d'une heure par semaine afin de réviser le français, les mathématiques et l'histoire-géographie. Chaque semaine, deux groupes sont formés par matière afin de travailler chacun sur un thème. Louise et Léon souhaiteraient obtenir le profil de leur groupe, constitué à partir des profils individuels des élèves, afin de créer une feuille de travail commune pour un groupe, mais adaptée au profil du groupe. Leurs collègues de mathématiques souhaiteraient également obtenir des profils de groupe pour adapter le logiciel MATHENPOCHE [Mathenpoche 2002] au besoin de chaque groupe d'élèves.

Dans la première partie d'EPROFILEA (cf. Figure 0-1 page 25), chaque enseignant peut intégrer les profils papier de ses élèves. Ensuite, avec le module Groupe, il peut créer deux profils de groupe par matière en listant les élèves contenus dans chaque groupe. À partir de ces profils de groupe et des stratégies pédagogiques définies par les enseignants de chaque matière, Adapte pourra proposer des séquences de travail papier-crayon en français et paramétrer le logiciel MATHENPOCHE pour les séances de révision en mathématiques.

SCÉNARIO 6 : PLAN LICENCE À L'UNIVERSITÉ

Lucie est maître de conférences en informatique. Dans le cadre du plan licence à l'université, elle organise trois séances de soutien d'1h30 durant chaque semestre et doit faire passer un QROC à ses étudiants lors de chaque séance de travaux dirigés.

En utilisant Adapte, elle pourrait créer ses énoncés de travaux dirigés et ainsi obtenir rapidement une autre version des exercices pour les séances de soutien. Elle pourrait également se servir du logiciel afin de créer les QROC qu'elle propose en examen. Dans les deux cas, elle doit utiliser Adapte en mode sans profil et sans liste d'étudiants afin d'obtenir à chaque fois une unique proposition de séquence de travail.

CHAPITRE 9. ÉVALUATION

PLAN DU CHAPITRE

9.1.	Introduction.....	233
9.2.	Méthodologie d'évaluation.....	233
9.3.	Grille d'analyse	233
9.3.1.	Évaluation de notre approche de la personnalisation de l'apprentissage	234
a.	Pertinence de l'approche : existence d'un besoin.....	234
b.	Pertinence de l'approche : utilisation de profils d'apprenants pour personnaliser l'apprentissage.....	234
c.	Pertinence de l'approche : utilisation d'un outil informatique pour aider l'enseignant dans sa tâche de personnalisation.....	234
9.3.2.	Évaluation du modèle PERSUA2.....	234
a.	Utilité.....	235
b.	Faisabilité	235
9.3.3.	Évaluation du modèle cPMDL.....	235
a.	Utilité.....	235
9.3.4.	Évaluation de l'approche GEPPETO	235
a.	Utilité.....	235
b.	Faisabilité	236
9.3.5.	Évaluation de l'approche GEPPETO _p	236
a.	Utilité de la typologie d'activités papier	236
b.	Utilité des patrons d'exercices.....	236
c.	Utilité de l'architecture générique des générateurs.....	236
d.	Utilisabilité.....	237
9.3.6.	Évaluation de l'approche GEPPETO _s	237
a.	Utilité du méta-modèle AKEPI.....	237
b.	Utilisabilité d'un modèle OKEP/x.....	237
c.	Efficacité	237
d.	Performance.....	237
9.3.7.	Évaluation du logiciel Adapte	238
a.	Utilisabilité.....	238
b.	Performance.....	238
c.	Utilité.....	238
9.4.	Présentation des expérimentations.....	238
9.4.1.	Mise en oeuvre du méta-modèle AKEPI	239
9.4.2.	Re-génération d'activités papier existantes.....	239
9.4.3.	Questionnaire sur les pratiques éducatives.....	240

9.4.4.	Expérimentation d'Adapte avec une enseignante	242
9.5.	Synthèse	243
9.5.1.	Évaluation de notre approche de la personnalisation de l'apprentissage	243
a.	Pertinence de l'approche : existence d'un besoin.....	243
b.	Pertinence de l'approche : utilisation de profils d'apprenants pour personnaliser l'apprentissage.....	244
c.	Pertinence de l'approche : utilisation d'un outil informatique pour aider l'enseignant dans sa tâche de personnalisation.....	244
9.5.2.	Évaluation du modèle PERSUA2.....	245
a.	Utilité.....	245
b.	Faisabilité.....	245
9.5.3.	Évaluation du modèle cPMDL.....	245
a.	Utilité.....	245
9.5.4.	Évaluation de l'approche GEPPETo	246
a.	Utilité.....	246
b.	Faisabilité	246
9.5.5.	Évaluation de l'approche GEPPETo _p	246
a.	Utilité de la typologie d'activités papier	246
b.	Utilité des patrons d'exercices.....	246
c.	Utilité de l'architecture générique des générateurs.....	247
d.	Utilisabilité.....	247
9.5.6.	Évaluation de l'approche GEPPETo _s	248
a.	Utilité du méta-modèle AKEPI.....	248
b.	Utilisabilité d'un modèle OKE _{p/x}	248
c.	Efficacité	248
d.	Performance.....	249
9.5.7.	Évaluation du logiciel Adapte	249
a.	Utilisabilité.....	249
b.	Performance.....	249
c.	Utilité.....	250
9.6.	Limites et perspectives d'évaluation.....	250
9.6.1.	Évaluation du modèle PERSUA2, du modèle cPMDL et de l'approche GEPPETo	251
9.6.2.	Évaluation de la typologie d'activités papier.....	251
9.6.3.	Évaluation du logiciel Adapte	252

9.1. INTRODUCTION

Dans cette thèse, nous avons présenté plusieurs contributions : un modèle de personnalisation unifiée des activités d'apprentissage (PERSUA2), un modèle de contraintes sur profils (cPMDL), une approche intégrant des modèles et processus pour adapter des activités pédagogiques (GEPPEO), la déclinaison de cette approche pour la personnalisation d'activités papier (GEPPEO_p) et son équivalent pour les activités logicielles (GEPPEO_s), enfin un logiciel implémentant ces contributions et permettant la génération d'activités personnalisées (Adapte). Nous avons pris soin, dans la mesure de nos moyens, d'évaluer chacune de nos propositions.

Dans ce chapitre, nous présentons notre méthodologie d'évaluation, notre grille d'analyse, puis les expérimentations menées dans le cadre de cette thèse. Nous présentons ensuite les résultats obtenus en reprenant notre grille d'analyse. Compte-tenu de l'ampleur du projet PERLEA dans lequel s'inscrit notre recherche, il ne nous a pas été possible (dans le cadre de cette thèse) de mettre en place une évaluation à grande échelle en conditions écologiques. Nous discutons de cet aspect et de nos perspectives à la fin de ce chapitre.

9.2. MÉTHODOLOGIE D'ÉVALUATION

Nous avons dans un premier temps identifié, pour chacune de nos contributions, les aspects à évaluer et les critères d'évaluation correspondants. Nous présentons cette grille d'analyse dans la section 9.3.

Ensuite, nous avons élaboré des expérimentations visant à évaluer chacun de ces aspects. Pour certains aspects, il nous a été facile de concevoir des expérimentations simples ou de vérifier les propriétés de nos propositions par nous-mêmes en concevant des tests unitaires. Mais rappelons que notre travail s'inscrit dans un projet de plus grande envergure et qu'il est parfois nécessaire de considérer l'ensemble du projet pour évaluer avec précision la pertinence de nos propositions. C'est pour cette raison que nous avons dû concevoir des expérimentations à plus grande échelle (et plus complexes), mais qui nous ont permis de valider ou de questionner plusieurs aspects à la fois. Le descriptif de chacune de ces expérimentations est présenté dans la section 9.4.

Enfin, nous avons repris notre grille d'analyse et nous avons discuté point par point chacun des critères étudiés. Le résultat de cette analyse est synthétisé dans la section 9.5.

9.3. GRILLE D'ANALYSE

Dans cette section, nous présentons la grille d'analyse que nous avons retenue pour nos évaluations. La présentation est faite sous forme synthétique : pour chacune de nos contributions, nous présentons le ou les critères d'analyse que nous avons retenus, et pour chaque critère, le point à analyser, la méthodologie envisagée et notre condition pour valider le critère.

9.3.1. ÉVALUATION DE NOTRE APPROCHE DE LA PERSONNALISATION DE L'APPRENTISSAGE

Pour traiter le problème de personnalisation de l'apprentissage, nous avons opté pour une approche visant à proposer aux enseignants un outil unique leur permettant d'intégrer les diverses informations dont ils disposent sur les apprenants, à travers l'utilisation de profils d'apprenants, et leur permettant d'adapter les activités proposées aux apprenants à leurs buts pédagogiques.

A. PERTINENCE DE L'APPROCHE : EXISTENCE D'UN BESOIN

Point à analyser	Les enseignants ont-ils des besoins en ce qui concerne l'apprentissage ? La personnalisation de l'apprentissage est-elle une démarche volontaire des enseignants et/ou souhaitée par les institutions ?
Méthodologie envisagée	Questionnaire multi-niveaux et multi-disciplines. Entretiens avec les enseignants partenaires du projet PERLEA.
Condition de validation	Estimation des besoins exprimés par les répondants.

B. PERTINENCE DE L'APPROCHE : UTILISATION DE PROFILS D'APPRENANTS POUR PERSONNALISER L'APPRENTISSAGE

Point à analyser	L'utilisation des profils d'apprenants pour la personnalisation des activités pédagogiques est-elle une réalité ?
Méthodologie envisagée	Questionnaire multi-niveaux et pluri-disciplines.
Condition de validation	Liée au nombre de réponses du questionnaire.

C. PERTINENCE DE L'APPROCHE : UTILISATION D'UN OUTIL INFORMATIQUE POUR AIDER L'ENSEIGNANT DANS SA TÂCHE DE PERSONNALISATION

Point à analyser	Les pédagogues sont-ils prêts à utiliser des outils informatiques lors de la tâche de personnalisation ?
Méthodologie envisagée	Questionnaire multi-niveaux et pluri-disciplines.
Condition de validation	Mesure des besoins exprimés par les répondants.

9.3.2. ÉVALUATION DU MODÈLE PERSUA2

Le modèle PERSUA2 représente les stratégies pédagogiques propres à chaque enseignant ainsi que les contextes dans lesquels ces stratégies peuvent être utilisées. Le principe du modèle est de définir des liens entre des contraintes sur profils permettant de sélectionner les apprenants ayant certaines caractéristiques dans leur profil, à des contraintes sur activités permettant d'adapter les activités aux intentions pédagogiques de l'enseignant.

A. UTILITÉ

Point à analyser	Le modèle permet-il de définir les besoins des enseignants ?
Méthodologie envisagée	Expérimentation avec plusieurs enseignants.
Condition de validation	Pas de limite d'expressivité lors de la création d'un modèle de personnalisation pour un niveau scolaire et une discipline non connus à l'avance.

B. FAISABILITÉ

Point à analyser	Le modèle est-il implémentable dans un logiciel ?
Méthodologie envisagée	Implémentation.
Condition de validation	Existence d'un système fonctionnel.

9.3.3. ÉVALUATION DU MODÈLE cPMDL

Le modèle cPMDL est une extension du langage de description des profils PDML. Cette extension permet de définir des contraintes afin de sélectionner des apprenants ayant certaines caractéristiques dans leur profil.

A. UTILITÉ

Point à analyser	Le modèle permet-il de définir les besoins des enseignants ?
Méthodologie envisagée	Expérimentation avec plusieurs enseignants.
Condition de validation	Pas de limite d'expressivité lors de la création de contraintes sur un profil pour un niveau scolaire et une discipline non connus à l'avance.

9.3.4. ÉVALUATION DE L'APPROCHE GEPPE TO

L'approche GEPPE TO a pour but de permettre aux enseignants d'intégrer leurs habitudes de travail lors du choix ou lors de la création des activités proposées aux apprenants.

A. UTILITÉ

Point à analyser	Cette approche peut-elle conduire à adapter les activités pédagogiques ?
Méthodologie envisagée	Étude théorique.
Condition de validation	Adaptation « théorique » de plusieurs types d'activités (papier et logicielles).

B. FAISABILITÉ

Point à analyser	Cette approche peut-elle être mise en œuvre dans un système pour adapter effectivement des activités ?
Méthodologie envisagée	Implémentation.
Condition de validation	Adaptation effective de plusieurs types d'activités (papier et logicielles).

9.3.5. ÉVALUATION DE L'APPROCHE GEPPETO_P

L'approche GEPPETO_P est la déclinaison de l'approche GEPPETO permettant d'adapter les activités papier. Elle s'appuie sur une typologie des activités papier. Plusieurs patrons d'exercices associés à des générateurs ont été proposés afin de créer des activités papier.

A. UTILITÉ DE LA TYPOLOGIE D'ACTIVITÉS PAPIER

Point à analyser	La typologie des activités papier est-elle exhaustive ?
Méthodologie envisagée	Validation faite par des experts en sciences de l'éducation.
Condition de validation	La typologie couvre le spectre des activités papier existantes.

B. UTILITÉ DES PATRONS D'EXERCICES

Point à analyser	Les patrons d'exercices permettent-ils de décrire des activités existantes ?
Méthodologie envisagée	Définition des activités de deux corpus (le corpus initial utilisé pour définir les patrons et un second corpus).
Condition de validation	Re-définition de tous les exercices des deux corpus.

C. UTILITÉ DE L'ARCHITECTURE GÉNÉRIQUE DES GÉNÉRATEURS

Point à analyser	L'architecture générique des générateurs permet-elle de définir une architecture propre à chaque type d'activités pris en compte dans GEPPETO _P ?
Méthodologie envisagée	Définition formelle de l'architecture de tous les générateurs utiles à la création des activités de notre typologie d'activités papier.
Condition de validation	Instanciation de l'architecture pour tous les patrons d'exercices.

D. UTILISABILITÉ

Point à analyser	Les générateurs d'exercices sont utilisables ?
Méthodologie envisagée	Tests unitaires.
Condition de validation	Obtention d'activités correspondant à la définition de contraintes sur activités.

9.3.6. ÉVALUATION DE L'APPROCHE GEPPETO_s

L'approche GEPPETO_s est la déclinaison de l'approche GEPPETO permettant d'adapter des activités logicielles. Elle s'appuie sur le méta-modèle AKEPI permettant de décrire un modèle OKEP propre à chaque EIAH à personnaliser.

A. UTILITÉ DU MÉTA-MODÈLE AKEPI

Point à analyser	Quel est le cadre d'application du méta-modèle ?
Méthodologie envisagée	Définition du modèle OKEP des EIAH contenus dans le corpus initial et d'EIAH ne faisant pas partie du corpus initial.
Condition de validation	Définition du modèle OKEP de différents types d'EIAH.

B. UTILISABILITÉ D'UN MODÈLE OKEP/X

Point à analyser	Le méta-modèle AKEPI permet-il de créer des modèles OKEP utilisables ?
Méthodologie envisagée	Tests unitaires.
Condition de validation	Création de modèles OKEP complets.

C. EFFICACITÉ

Point à analyser	Le méta-modèle peut-il être implémenté ?
Méthodologie envisagée	Tests unitaires.
Condition de validation	Le méta-modèle peut être utilisé par un système.

D. PERFORMANCE

Point à analyser	Quel est le temps de création d'un modèle OKEP ?
Méthodologie envisagée	Tests avec des experts.
Condition de validation	Définition de modèles d'EIAH de différents types.

9.3.7. ÉVALUATION DU LOGICIEL ADAPTE

Le logiciel Adapte est l'un des modules de l'environnement EPROFILEA. Il propose des séquences personnalisées d'activités adaptées aux buts pédagogiques de l'enseignant et propres à chaque profil d'apprenant. C'est dans ce logiciel que nous avons mis en œuvre nos contributions théoriques.

A. UTILISABILITÉ

Point à analyser	Ergonomie des interfaces du logiciel.
Méthodologie envisagée	Expérimentation avec des enseignants. Jugements d'experts.
Condition de validation	Respect des guides de style, des critères d'évaluation ergonomique. L'enseignant peut définir un modèle de personnalisation dans le logiciel sans aide extérieure.

B. PERFORMANCE

Point à analyser	Temps nécessaire à la génération de séquences d'activités personnalisées.
Méthodologie envisagée	Expérimentation avec des enseignants. Tests unitaires.
Condition de validation	Le temps nécessaire doit être acceptable pour un enseignant.

C. UTILITÉ

Point à analyser	Le comportement du système est-il celui attendu ?
Méthodologie envisagée	Expérimentation avec des enseignants. Tests unitaires.
Condition de validation	Confrontation des résultats obtenus par le système aux besoins exprimés par chaque enseignant : les séquences personnalisées doivent correspondre aux attentes de l'enseignant.

9.4. PRÉSENTATION DES EXPÉRIMENTATIONS

Dans cette section, nous présentons les expérimentations « complexes » menées dans le cadre de cette thèse : la mise en œuvre du méta-modèle AKEPI, la re-génération d'activités papier existantes, le questionnaire sur les pratiques éducatives et l'expérimentation d'Adapte avec une enseignante. Nous ne présentons pas les expérimentations plus simple comme les tests unitaires.

9.4.1. MISE EN OEUVRE DU MÉTA-MODÈLE AKEPI

Dans le cadre de l'approche GEPPETo_s, nous avons proposé le méta-modèle AKEPI permettant d'acquérir les connaissances nécessaires à la personnalisation d'un EIAH afin de créer un modèle OKEP de l'EIAH (cf. section 6.4.3 page 153).

Dans le but de définir le méta-modèle AKEPI, nous avons étudié trente logiciels pédagogiques pour identifier ce qui pouvait être personnalisable, et comment, dans de tels systèmes. Ceci nous a permis d'identifier cinq facettes de personnalisation (les activités, l'ordonnancement de ces activités, les fonctionnalités, les rétroactions et l'interface). Nous avons ensuite sélectionné onze systèmes permettant une personnalisation (via des fichiers de configuration ou via leur interface) de types variés afin de définir précisément le méta-modèle AKEPI : ACTIVEMATH [Melis et al. 2001], ANDES [VanLehn et al. 2005], AMBRE-ADD [Guin-Duclosson et al. 2002], APLUSIX [Bouhineau et al. 2006], MATHENPOCHE [Mathenpoche 2002], MINI LEHRER [MiniLehrer 2007], LILIMATH [LiliMath 2000], ORTHOGRAMME [Orthogramme 1999], ROBOTTEACH [Leroux 1998], TRILOCONTE [Triloconte 2007], TRI [Jean-Daubias 2007]. Pour chacun des onze EIAH, nous avons défini le modèle OKEP de l'EIAH.

Pour tester notre méta-modèle, nous l'avons ensuite utilisé pour définir les modèles OKEP de cinq nouveaux logiciels pédagogiques non pris en compte dans notre étude initiale : ABALECT [Chevé 2005], DISCRIMOT [Birbandt 2009], LECTRIS [Lameyse 2009], POINT PAR POINT [Maille 2005], TABLES AU TRÉSOR [Jean-Daubias 2009]. Nous avons pu, pour chacun d'eux, créer le modèle OKEP correspondant.

Il faut néanmoins noter que le cadre d'application du méta-modèle AKEPI se limite aux EIAH proposant des activités individuelles d'apprentissage. La raison de cette limite est que nous n'avons pas inclus de logiciel permettant un apprentissage collaboratif ou collectif dans notre étude préliminaire des logiciels. Toutefois, si un EIAH de ce type (collaboratif ou collectif) se personnalise de manière unique pour un groupe d'apprenants, le méta-modèle AKEPI permet alors de le décrire et le logiciel Adapte pourra le personnaliser à partir des profils de groupes.

La mise en œuvre de ce méta-modèle dans la partie « Intégration d'un nouvel EIAH » d'Adapte (cf. section 8.2.1 page 202) permet de définir un modèle propre à chaque EIAH. Cette définition par un expert peut durer de une heure pour de petits logiciels comme POINT PAR POINT [Maille 2005] à trois jours pour des logiciels plus complet comme AMBRE-ADD [Guin-Duclosson et al. 2002].

9.4.2. RE-GÉNÉRATION D'ACTIVITÉS PAPIER EXISTANTES

Dans le cadre de cette thèse, nous avons proposé une typologie des activités papier (cf. section 6.3.2 page 120) sur laquelle nous nous appuyons pour personnaliser les activités papier dans le cadre de GEPPETo_p. À cette typologie nous associons un ensemble de générateurs d'exercices (cf. section 6.3.3.d page 141).

Afin de définir cette typologie d'activités papier, nous avons construit un corpus d'exercices papier issus de différentes sources (cf. section 6.3.1 page 119) : programmes scolaires, générateurs d'exercices, sites internet proposant des bases d'exercices, exercices créés par des enseignants, livres scolaires et universitaires, travaux de recherche dans le domaine des EIAH.

Une fois la typologie définie, nous avons construit un second corpus pour établir si cette typologie et les générateurs qui lui sont associés permettaient de créer les exercices du second corpus. Pour cela, nous avons collecté tous les exercices utilisés par cinq enseignants de collègue

dans le cadre de leurs enseignements. Le second corpus contenait donc des exercices destinés à l'enseignement :

- Des Sciences et Vie de la Terre (SVT) pour des élèves de 6^{ème} et 3^{ème} ;
- De l'anglais pour des élèves de 5^{ème}, 4^{ème} et 3^{ème} ;
- De français pour des élèves de 6^{ème}, 5^{ème}, 4^{ème} et 3^{ème} ;
- Des mathématiques pour des élèves de 5^{ème}, 4^{ème} et 3^{ème} ;
- De géographie pour des élèves de 6^{ème} et 4^{ème} ;
- D'histoire pour des élèves de 6^{ème} et 4^{ème} ;
- D'éducation civique pour des élèves de 6^{ème}.

Nous avons pu recréer l'ensemble des exercices de ce corpus. Ceci ne valide que partiellement notre typologie. Nous savons que celle-ci n'est pas exhaustive, mais elle se veut assez large pour couvrir de nombreux exercices. Pour identifier les manques de notre typologie, un travail en partenariat avec des chercheurs en sciences de l'éducation est maintenant nécessaire.

9.4.3. QUESTIONNAIRE SUR LES PRATIQUES ÉDUCATIVES

En juin 2009, nous avons diffusé un questionnaire sur les pratiques éducatives auprès d'un large public (établissements d'enseignement allant de l'école maternelle à l'université). Ce questionnaire, disponible en Annexe K page 363, portait sur trois points : l'utilisation de profils d'apprenants, l'utilisation de logiciels pédagogiques et la personnalisation de l'enseignement.

Nous avons obtenu 95 réponses. Les résultats pouvant se rapporter à l'évaluation de cette thèse sont fournis en Figure 9-1.

Les constats que nous pouvons faire aux vues de ces résultats sont les suivant :

- Nous avons obtenu très peu de réponses d'enseignants de l'école primaire ou du collège (cf. 1 sur la Figure 9-1), alors que c'est le public qui nous paraît le plus concerné par les outils proposés dans le cadre du projet PERLEA. Ce faible taux de réponse est lié au mode de diffusion du questionnaire (surtout par messagerie électronique sur des listes de diffusion et par l'intermédiaire des enseignants avec lesquels nous avons des contacts) et limite la portée des résultats du questionnaire ;
- Une partie des enseignants créent des profils d'apprenants (environ un tiers, cf. 2 sur la Figure 9-1) pour les présenter aux élèves, mais surtout pour s'en servir afin de personnaliser l'enseignement qu'ils proposent (cf. 3 sur la Figure 9-1). Cette personnalisation concerne surtout le contenu et le niveau de difficulté des séquences d'activités proposées (cf. 5 sur la Figure 9-1).
- Les enseignants qui ont répondu au questionnaire sont plus intéressés par la personnalisation des activités papier que par celle des logiciels pédagogiques (cf. 6 sur la Figure 9-1). Cela peut s'expliquer par le fait que les directives émanant des institutions demandent de personnaliser les activités papier mais ne préconisent rien pour les logiciels pédagogiques (cf. 7 sur la Figure 9-1). De plus, très peu d'enseignants répondants utilisent des logiciels pédagogiques dans le cadre de leurs enseignements (cf. 4 sur la Figure 9-1).
- Les deux tiers des enseignants ayant répondu seraient prêts à utiliser un logiciel les aidant dans cette tâche de personnalisation (cf. 8 sur la Figure 9-1) d'autant que la majorité a accès à une salle informatique dans son établissement (cf. 9 sur la Figure 9-1) et qu'un tiers utilise quotidiennement des ordinateurs pour préparer ses enseignements (cf. 10 sur la Figure 9-1).

	Questions	Réponses	Pourcentage
1	Niveau d'enseignement	Université	43
		Lycée	28
		Collège	14
		Élémentaire	8
		Autres (formation continue...)	7
2	Création de profils pour ... (plusieurs réponses possibles)	Chaque élève	41
		Des groupes d'élèves au sein d'une classe	22
		Chaque classe	35
3	But de la création de profils (plusieurs réponses possibles)	Visualisation des profils	37
		Travail sur les profils	13
		Proposition d'activités personnalisées	52
4	Utilisation de logiciels pédagogiques	Oui	38
		Non	60
5	Personnalisation actuellement mise en œuvre ... (plusieurs réponses possibles)	Sur la longueur des séquences d'activités	14
		Sur le contenu des séquences	61
		Sur le niveau de difficulté	62
6	Intérêt pour ... (plusieurs réponses possibles)	Des exercices adaptés à chaque élève	52
		Des logiciels pédagogiques avec des sessions personnalisées	35
7	Demande extérieure de proposer ... (plusieurs réponses possibles)	Des exercices adaptés à chaque élève	36
		Des logiciels pédagogiques avec des sessions personnalisées	4
8	Prêt à utiliser un logiciel d'aide à la personnalisation des activités proposées aux élèves	Oui	63
		Non	20
9	Accès à une salle informatique dans l'établissement	Oui	86
		Non	3
10	Utilisation des ordinateurs pour préparer les enseignements	Quotidienne	60
		Hebdomadaire	23
		Mensuelle ou moins fréquente	2

Figure 9-1 : Extrait des résultats du questionnaire sur les pratiques éducatives diffusé en juin 2009.

Comme pour tout questionnaire, ces résultats sont à nuancer. En effet, les réponses ont été faites sur la base du volontariat, elles ne représentent qu'un panel de 95 enseignants pas forcément représentatif du public cible d'EPROFILEA et donc d'Adapte. Plusieurs enseignants nous ont par ailleurs contactés pour nous dire que ce questionnaire ne correspondait pas à leur façon de travailler et qu'ils ne souhaitaient donc pas y répondre. Nous ne généraliserons donc pas ces résultats, mais ceux-ci laissent toutefois penser que les outils proposés dans le cadre du projet PERLEA, et notamment le logiciel Adapte, peuvent être une aide pour les enseignants souhaitant mettre en place une personnalisation de l'apprentissage que ce soit à partir de profils d'apprenants ou non. En effet, les enseignants ont pour consigne de mettre en place cette personnalisation de l'apprentissage mais ils manquent de ressources pour le faire.

9.4.4. EXPÉRIMENTATION D'ADAPTE AVEC UNE ENSEIGNANTE

En avril 2009, nous avons mis à l'essai l'environnement EPROFILEA durant une semaine avec Bérengère, une enseignante de français au collège, que l'on peut qualifier de non experte puisqu'elle est extérieure au projet PERLEA, et n'est donc jamais intervenue dans la conception des modèles ou des outils proposés dans le cadre de ce projet. Durant cette semaine, une journée a été consacrée à l'évaluation du modèle PERSUA2 permettant une personnalisation unifiée de l'apprentissage et du logiciel Adapte le mettant en œuvre.

Pour pouvoir évaluer le modèle PERSUA2, nous avons tout d'abord demandé à Bérengère de définir sur papier la façon dont elle utiliserait les profils d'apprenants qu'elle avait précédemment créés dans EPROFILEA, pour proposer des séquences de travail adaptées à chacun de ses élèves. Pendant deux heures, elle a ainsi défini des règles de personnalisation pouvant s'appliquer à une structure de profils contenant 21 éléments regroupés en quatre briques. Ces éléments étaient définis pour neuf d'entre eux selon une échelle numérique (une note de 0 à 10) et pour les douze autres selon une échelle textuelle ordonnée (une liste énumérée de deux ou trois éléments selon les cas). Ces règles établissaient un lien entre la valeur d'un ou plusieurs des éléments du profil des élèves et des descriptifs d'exercices.

Une fois « son modèle » de personnalisation établi, nous lui avons demandé de le retranscrire dans le logiciel Adapte en créant la stratégie pédagogique correspondante. Pour cela, nous avons préalablement créé des structures d'activités correspondant à la description qu'elle avait faite des exercices. Cette étape préliminaire a été menée afin que l'évaluation la partie principale d'Adapte (la création des règles d'affectation des activités aux apprenants) et donc du modèle PERSUA2 ne soit pas biaisée par des problèmes lors de la création des structures d'activités. La création de la stratégie pédagogique à partir des notes prises sur papier a duré une heure et a abouti à la création d'une stratégie pédagogique contenant 24 règles d'affectation des activités aux apprenants.

Le logiciel Adapte, et donc le modèle PERSUA2 sur lequel il s'appuie, ont permis à Bérengère de créer une stratégie pédagogique correspondant entièrement à ses notes papier. Néanmoins, nous avons dû lui expliquer comment utiliser les opérateurs « et, ou » afin de traduire certaines de ses règles de personnalisation. De plus, dans certains cas, elle aurait souhaité contraindre l'enchaînement des exercices proposés. Par exemple, pour des élèves ayant une difficulté donnée, leur faire faire l'exercice 1, puis s'ils ont réussi cet exercice, alors leur faire faire l'exercice 2, sinon l'exercice 3. Notre modèle n'intègre pas ce type de règles. Toutefois, en modifiant l'énoncé des exercices, elle a pu décrire cette règle de la façon suivante « si un élève a telle difficulté alors lui faire faire les exercices 1, 2 et 3 ». Les contraintes sur l'enchaînement des exercices ont alors été intégrées au début du deuxième exercice par la phrase « Si vous avez réussi l'exercice précédent, passez à l'exercice 3, sinon faites uniquement cet exercice ».

Une fois la stratégie pédagogique définie, nous avons lancé la génération des séquences personnalisées pour une de ses deux classes de 6^{ème}. Pour cette classe de 23 élèves, la génération des séquences et leur enregistrement au format HTML a duré moins de 10 secondes. Afin de valider le processus associé au modèle PERSUA2 et sa mise en œuvre dans Adapte, nous avons étudié les 23 séquences d'activités générées et vérifié que chacune correspondait bien à la personnalisation que souhaitait mettre en place Bérengère.

Pour terminer cette évaluation, Bérengère a créé une seconde stratégie pédagogique sur une autre structure de profils, mais en utilisant cette fois directement le logiciel Adapte. Cette création a duré 25 minutes pour une stratégie contenant 13 règles.

9.5. SYNTHÈSE

Dans cette section, nous reprenons la grille d'analyse que nous avons retenue pour nos évaluations afin de présenter les résultats de nos différentes évaluations. La présentation est faite sous forme synthétique : pour chacune de nos contributions, nous rappelons le ou les critères d'analyse, et pour chaque critère, le point à analyser. Nous indiquons ensuite dans quelle partie du document sont décrites les évaluations qui conduisent à ces résultats, avant de présenter les résultats que nous analysons.

9.5.1. ÉVALUATION DE NOTRE APPROCHE DE LA PERSONNALISATION DE L'APPRENTISSAGE

A. PERTINENCE DE L'APPROCHE : EXISTENCE D'UN BESOIN

Point à analyser	La personnalisation de l'apprentissage est-elle un besoin pour les enseignants ? Est-ce une démarche volontaire et/ou souhaité par les institutions ?
Description de l'évaluation	Questionnaire – section 9.4.3.
Résultat	Les deux tiers des répondants personnalisent actuellement le contenu et le niveau des séquences d'activités proposées aux apprenants. L'intérêt des répondants porte plus sur la personnalisation des activités papier que des logiciels pédagogiques. Cela s'explique par le fait qu'on leur demande uniquement de personnaliser des activités papier et que peu d'entre eux utilisent des logiciels pédagogiques dans le cadre de leurs enseignements.
Analyse	La majorité des enseignants répondants appartient au milieu universitaire. Ces enseignants ne sont donc pas représentatifs du public cible d'EPROFILEA (enseignants à l'école élémentaire, au collège ou au lycée). Les résultats du questionnaire ont donc une portée limitée. Toutefois, la personnalisation de l'apprentissage semble être souhaitée par les enseignants et demandée par les institutions.

Point à analyser	La personnalisation de l'apprentissage est-elle un besoin pour les enseignants ? Est-ce une démarche volontaire et/ou souhaité par les institutions ?
Description de l'évaluation	Entretien avec des enseignants partenaires du projet.
Résultat	Ces enseignants expriment le souhait de personnaliser, souhait renforcé par les demandes des institutions pour le faire, mais ils soulignent le manque de moyens en temps pour le faire, ce qui conduit dans de nombreux cas à une absence de personnalisation.
Analyse	Les outils proposés dans le cadre du projet PERLEA et notamment le logiciel Adapte visent un gain de temps lors de la personnalisation des activités, ce qui pourrait rendre la personnalisation plus accessible aux enseignants.

B. PERTINENCE DE L'APPROCHE : UTILISATION DE PROFILS D'APPRENANTS POUR PERSONNALISER L'APPRENTISSAGE

Point à analyser	L'utilisation des profils d'apprenants pour la personnalisation des activités pédagogiques est-elle une réalité ?
Description de l'évaluation	Questionnaire – section 9.4.3.
Résultat	Parmi les enseignants répondants, 40% créent un profil pour chacun de leurs élèves, et parmi ceux-ci 52% créent les profils dans le but de proposer des activités personnalisées.
Analyse	Nous avons vu précédemment que les résultats du questionnaire ont une portée limitée. Toutefois, l'utilisation de profils pour personnaliser l'apprentissage semble être une pratique assez répandue chez les enseignants.

C. PERTINENCE DE L'APPROCHE : UTILISATION D'UN OUTIL INFORMATIQUE POUR AIDER L'ENSEIGNANT DANS SA TÂCHE DE PERSONNALISATION

Point à analyser	Les pédagogues sont-ils prêts à utiliser des outils informatiques lors de la tâche de personnalisation ?
Description de l'évaluation	Questionnaire – section 9.4.3.
Résultat	Parmi les enseignants répondants, 63% sont prêts à utiliser un outil informatique pour les aider lors de la personnalisation de l'apprentissage.
Analyse	Nous rappelons la portée limitée des résultats du questionnaire. Toutefois, proposer un outil informatique aux enseignants pour les aider dans leur tâche semble pertinent.

9.5.2. ÉVALUATION DU MODÈLE PERSUA2

A. UTILITÉ

Point à analyser	Le modèle permet-il de définir les besoins des enseignants ?
Description de l'évaluation	Expérimentation avec une enseignante – section 9.4.4.
Résultat	L'enseignante a pu définir un modèle de personnalisation, et ce modèle de personnalisation est conforme à son « projet » papier.
Analyse	Le résultat est encourageant puisqu'une enseignante a pu définir ses besoins en utilisant notre modèle. Il est maintenant nécessaire de refaire l'expérimentation avec un nombre plus important d'enseignants.
Remarque	Le modèle peut être étendu pour prendre en compte de nouveaux besoins.

B. FAISABILITÉ

Point à analyser	Le modèle est-il implémentable dans un logiciel ?
Description de l'évaluation	Mise en œuvre dans le logiciel Adapte - Chapitre 8.
Résultat	L'implémentation est entièrement effectuée, le logiciel fonctionne pleinement.
Analyse	Le modèle est implémentable dans un logiciel.

9.5.3. ÉVALUATION DU MODÈLE cPMDL

A. UTILITÉ

Point à analyser	Le modèle permet-il de définir les besoins des enseignants ?
Description de l'évaluation	Expérimentation avec une enseignante – section 9.4.4.
Résultat	L'enseignante a pu contraindre le profil sans se limiter.
Analyse	Le résultat est encourageant puisqu'une enseignante a pu définir ses besoins en utilisant notre modèle. Il est maintenant nécessaire de refaire l'expérimentation avec un nombre plus important d'enseignants.
Remarque	Le modèle peut être étendu pour prendre en compte de nouveaux besoins.

9.5.4. ÉVALUATION DE L'APPROCHE GEPPE TO

A. UTILITÉ

Point à analyser	Cette approche peut-elle être utilisée pour adapter les activités pédagogiques ?
Description de l'évaluation	Déclinaison pour la personnalisation des activités papier et logicielles – section 6.3 et 6.4.
Résultat	Proposition des approches GEPPE TO _P et GEPPE TO _S .
Analyse	Cette approche peut être utilisée pour adapter les activités pédagogiques.

B. FAISABILITÉ

Point à analyser	Cette approche peut-elle être utilisée dans un système pour spécifier des activités ?
Description de l'évaluation	Mise en œuvre dans le logiciel Adapte - Chapitre 8.
Résultat	Mise en œuvre des approches GEPPE TO _P et GEPPE TO _S dans le logiciel Adapte qui permet d'adapter des activités pédagogiques.
Analyse	Cette approche peut être utilisée dans un système pour spécifier des activités.

9.5.5. ÉVALUATION DE L'APPROCHE GEPPE TO_P

A. UTILITÉ DE LA TYPOLOGIE D'ACTIVITÉS PAPIER

Point à analyser	La typologie des activités papier est-elle exhaustive ?
Description de l'évaluation	Validation des experts en sciences de l'éducation.
Résultat	Non effectuée à l'heure actuelle.
Analyse	Cette évaluation n'est pas de notre ressort.

B. UTILITÉ DES PATRONS D'EXERCICES

Point à analyser	Est-il possible de décrire des activités existantes à l'aide des patrons d'exercices ?
Description de l'évaluation	Redéfinition de tous les exercices contenus dans deux corpus d'activités papier – section 9.4.2

Résultat	Les patrons d'exercices permettent de définir les activités contenues dans les deux corpus (le corpus initial et celui de l'expérimentation).
Analyse	Il est possible de décrire des activités existantes à l'aide des patrons d'exercices.

C. UTILITÉ DE L'ARCHITECTURE GÉNÉRIQUE DES GÉNÉRATEURS

Point à analyser	L'architecture générique proposée pour les générateurs permet-elle de définir une architecture propre à chaque type d'activité pris en compte dans GEPPETO _p ?
Description de l'évaluation	Définition formelle de l'architecture des générateurs utiles – section 6.3.3.d et Annexe G.
Résultat	Instanciation de l'architecture générique pour tous les patrons d'exercices, excepté pour le générateur « Démonstration ».
Analyse	Des travaux complémentaires doivent être menés pour prendre en compte des démonstrateurs afin de définir le générateur « Démonstration », mais la définition de l'architecture des sept autres générateurs nous paraît suffisante pour valider notre architecture générique de générateurs semi-automatiques d'exercices.

D. UTILISABILITÉ

Point à analyser	Les générateurs d'exercices sont utilisables ?
Description de l'évaluation	Tests unitaires.
Résultat	À l'heure actuelle, nous n'avons pas implémenté tous les générateurs d'exercices papier. Nous en avons implémenté cinq au sein du logiciel Adapte (Travail sur texte, Travail sur illustration, Tables, Problème scientifique et Questions – réponses). Dans certains, toutes les options ne sont pas disponibles, mais globalement, ils permettent de créer les exercices en fonction des contraintes sur activités spécifiées par l'utilisateur d'Adapte.
Analyse	Les générateurs implémentés sont fonctionnels. Néanmoins il reste à faire une évaluation ergonomique des interfaces de chacun des générateurs.

9.5.6. ÉVALUATION DE L'APPROCHE GEPPETO_s

A. UTILITÉ DU MÉTA-MODÈLE AKEPI

Point à analyser	Quel est le cadre d'application du méta-modèle ?
Description de l'évaluation	Définition du modèle OKEP de divers EIAH - section 9.4.1.
Résultat	Nous avons pu définir le modèle OKEP des onze EIAH de notre corpus initial et de cinq EIAH externes à ce corpus. Nous n'avons pour l'instant trouvé aucun EIAH proposant un apprentissage individuel ne pouvant être décrit avec le méta-modèle AKEPI.
Analyse	Nous pouvons définir le cadre d'application du méta-modèle AKEPI : il permet de décrire les logiciels pédagogiques de divers types (tuteur, micromonde, simulateur, exerciceur) proposant un apprentissage individuel. Concernant les logiciels proposant un apprentissage collaboratif ou collectif, nous n'en n'avons pas retenu pour définir le méta-modèle AKEPI, nous ne pouvons donc ni infirmer, ni confirmer qu'ils peuvent être décrits grâce au méta-modèle AKEPI.

B. UTILISABILITÉ D'UN MODÈLE OKEP/X

Point à analyser	Le méta-modèle AKEPI permet-il de créer des modèles OKEP utilisables ?
Description de l'évaluation	Tests unitaires.
Résultat	Nous avons pu définir, dans la partie « Intégration du nouvel EIAH » du logiciel Adapte, le modèle OKEP des onze EIAH de notre corpus initial et de cinq EIAH externes à ce corpus. Ces modèles OKEP sont utilisables dans la seconde partie du logiciel Adapte proposant les séquences de travail personnalisées.
Analyse	Le méta-modèle AKEPI permet de créer des modèles OKEP utilisables.

C. EFFICACITÉ

Point à analyser	Le méta-modèle peut-il être implémenté ?
Description de l'évaluation	Mise en œuvre dans le logiciel Adapte - Chapitre 8.
Résultat	L'implémentation est entièrement effectuée, le logiciel fonctionne pleinement.
Analyse	Le méta-modèle est implémentable dans un logiciel.

D. PERFORMANCE

Point à analyser	Quel est le temps de création d'un modèle OKEP ?
Description de l'évaluation	Définition du modèle OKEP de divers EIAH - section 9.4.1.
Résultat	Le temps de création d'un modèle varie d'une heure à trois jours sur les EIAH testés.
Analyse	Le temps de création d'un modèle OKEP dépend de la complexité et du nombre d'activités proposées dans l'EIAH à décrire.

9.5.7. ÉVALUATION DU LOGICIEL ADAPTE

A. UTILISABILITÉ

Point à analyser	Ergonomie des interfaces du logiciel.
Description de l'évaluation	Expérimentation avec des enseignants.
Résultat	Les enseignants ont parfois besoin d'explications sur les actions à effectuer à l'interface.
Analyse	Une évaluation ergonomique du logiciel devrait permettre de résoudre certains des problèmes rencontrés, mais il manque surtout une aide au sein du logiciel pour guider les enseignants dans l'utilisation d'un outil aux premiers abords assez complexe.

B. PERFORMANCE

Point à analyser	Temps nécessaire à la génération de séquences personnalisées.
Description de l'évaluation	Expérimentation avec une enseignante – section 9.4.4.
Résultat	<p>Lors de la première utilisation du logiciel, l'enseignante a mis 1h30 pour convertir une stratégie pédagogique définie sur papier en modèle de personnalisation intégré au logiciel Adapte.</p> <p>Lors de la seconde utilisation, l'enseignante a mis 25 minutes pour définir, directement sur le logiciel, une autre stratégie pédagogique afin de créer un modèle de personnalisation contenant moitié moins de règles que le premier.</p>

Analyse	Le temps de définition d'une stratégie pédagogique dans Adapte est assez important. Toutefois, ce temps semble décroître rapidement au fur et à mesure de l'utilisation du logiciel par les enseignants. De plus, la création d'une stratégie pédagogique ne se fait qu'une seule fois, pour plusieurs utilisations du logiciel. Enfin, pour définir de nouvelles stratégies pédagogiques, les enseignants peuvent repartir de stratégies existantes et ainsi gagner du temps.
---------	--

C. UTILITÉ

Point à analyser	Le comportement du système est-il celui attendu ?
Description de l'évaluation	Expérimentation avec une enseignante – section 9.4.4.
Résultat	Lors de l'expérimentation, les séquences de travail proposées correspondaient aux intentions pédagogiques de l'enseignante.
Analyse	Le résultat est encourageant puisqu'une enseignante a pu obtenir des séquences adaptées à ses méthodes de travail et aux profils de ses élèves. Il est maintenant nécessaire de refaire l'expérimentation avec un nombre plus important d'enseignants.

Point à analyser	Le comportement du système est-il celui attendu ?
Description de l'évaluation	Tests unitaires.
Résultat	Adapte permet de personnaliser les EIAH du cadre d'application d'AKEPI. Toutefois, lorsque la configuration nécessite d'utiliser des générateurs contenus dans l'EIAH à personnaliser, il est nécessaire que le logiciel Adapte et l'EIAH à personnaliser se trouvent sur le même ordinateur. De plus, dans le cas où Adapte fournit des fichiers de configuration, deux cas peuvent poser problème : si le logiciel n'est pas sur le même ordinateur qu'Adapte ou si un même fichier de configuration doit être modifié différemment pour plusieurs apprenants, alors les fichiers de configuration sont fournis à l'enseignant dans un ou plusieurs dossiers propres à Adapte, fichiers que l'enseignant devra ensuite placer dans le ou les dossiers indiqués par Adapte.
Analyse	Notre approche n'apporte pas de solution concernant les problèmes de communication entre systèmes distants ou concernant le paramétrage d'applications installées sur des machines distantes.

9.6. LIMITES ET PERSPECTIVES D'ÉVALUATION

Dans ce chapitre, nous avons présenté la grille d'analyse que nous avons définie pour évaluer nos contributions théoriques et leur mise en œuvre. Nous avons ensuite présenté les évaluations

que nous avons menées dans le cadre de cette thèse avant de présenter leurs résultats en fonction de notre grille d'analyse. Nous montrons à présent, pour chaque contribution, les limites de ces évaluations et indiquons les perspectives d'évaluation qui pourraient permettre de réduire ces limites.

9.6.1. ÉVALUATION DU MODÈLE PERSUA2, DU MODÈLE cPMDL ET DE L'APPROCHE GEPPETO

Origine de la limite	Test avec un seul enseignant.
Motif de la non-évaluation à ce stade	Nécessite une évaluation globale de l'environnement EPROFILEA.
Solution envisagée	Expérimentation à grande échelle avec plusieurs enseignants non impliqués dans le projet PERLEA.
Pré-requis pour la mise en œuvre de la solution envisagée	Avoir un environnement EPROFILEA entièrement opérationnel et fiable techniquement.
Motivations	Les modèles PERSUA2, cPMDL et le méta-modèle AKEPI sont ouverts et garantissent donc la possibilité d'évolution de notre proposition. Il reste maintenant à définir ce qui doit évoluer et le coût de cette évolution, d'où le besoin d'expérimentations plus poussées à ce stade. Rappelons cependant que l'objectif de la thèse n'est pas d'évaluer EPROFILEA, mais puisque nos contributions théoriques sont implémentées au sein d'un environnement, les expérimentations avec les enseignants peuvent difficilement se faire sans prendre en compte tout l'environnement.

9.6.2. ÉVALUATION DE LA TYPOLOGIE D'ACTIVITÉS PAPIER

Origine de la limite	Validation de l'exhaustivité de la typologie d'activités papier.
Motif de la non-évaluation à ce stade	Manque de compétences adéquates.
Solution envisagée	Validation par des experts en sciences de l'éducation.
Pré-requis pour la mise en œuvre de la solution envisagée	Aucun.
Motivations	Compléter la typologie avec d'éventuelles activités identifiées comme manquantes.

9.6.3. ÉVALUATION DU LOGICIEL ADAPTE

Origine de la limite	Problème d'ergonomie des interfaces.
Motif de la non-évaluation à ce stade	Non prioritaire dans le cadre de cette thèse.
Solution envisagée	Analyse par des ergonomes.
Pré-requis pour la mise en œuvre de la solution envisagée	Aucun.
Motivations	La mise en œuvre des contributions au sein du logiciel Adapte permet de valider les modèles proposés. Mais si l'on souhaite que nos propositions soient acceptées par les enseignants et donc utilisées dans les établissements scolaires, il faut fournir aux enseignants un outil adapté, c'est-à-dire un outil utile (qui les aide dans leur tâche) et utilisable (qui possède une interface rendant possible sont utilisation). Ce critère d'utilisabilité implique donc la nécessité de faire une évaluation ergonomique du logiciel.
Origine de la limite	Performances non évaluées.
Motif de la non-évaluation à ce stade	Non prioritaire dans le cadre de cette thèse.
Solution envisagée	Tests unitaires classiques (performances, bugs, portage) selon une méthodologie de génie logiciel usuelle (à déterminer).
Pré-requis pour la mise en œuvre de la solution envisagée	Aucun.
Motivations	Nécessaire dans une perspective de valorisation.
Remarque	Même si ces tests de performances n'ont pas été faits, nous pouvons noter que, malgré l'absence de certaines fonctionnalités, le système fonctionne correctement et donne des résultats satisfaisants.

Les limites identifiées lors de l'évaluation de nos contributions ouvrent des perspectives de recherches intéressantes qui seront discutées dans la conclusion de ce manuscrit.

CONCLUSION

Dans le cadre de cette thèse en informatique, nous avons abordé la question de la personnalisation de l'apprentissage. L'objectif de la personnalisation de l'apprentissage est d'adapter des activités pédagogiques en prenant en compte à la fois les spécificités des apprenants et les objectifs pédagogiques de leurs enseignants. Les situations dans lesquelles la personnalisation est possible sont variées : enseignement dans les écoles, collèges et lycées mais aussi à l'Université ou encore dans le cadre de la formation à distance. De plus, la personnalisation concerne aussi bien les activités papier que les logiciels pédagogiques.

Cette recherche consistait à identifier un processus qui permette à la fois de personnaliser des séances de travail sur papier et des séances de travail sur des logiciels pédagogiques. Nous souhaitons que ce processus permette de prendre en compte les spécificités de chaque apprenant, en s'appuyant sur son profil, mais qu'il prenne également en compte les buts et les habitudes pédagogiques des enseignants. Enfin, nous souhaitons que ce processus soit implémentable dans un système externe aux logiciels à personnaliser.

Notre problématique s'est donc décomposée en trois points : comment exploiter les profils d'apprenants pour prendre en compte les individualités des apprenants ? Comment adapter une activité pour prendre en compte les besoins et habitudes pédagogiques d'un enseignant ? Et enfin, comment attribuer une activité à un apprenant ?

Pour répondre au premier point, nous avons proposé le modèle cPMDL. Cette extension du langage de modélisation des profils PMDL permet de contraindre les profils des apprenants afin de sélectionner ceux ayant les caractéristiques requises dans les contraintes. cPMDL nous permet donc d'exploiter les informations contenues dans les profils dans le processus de personnalisation.

Pour répondre au second point, nous avons proposé l'approche GEPPEO. Cette approche s'appuie sur des modèles et des processus génériques permettant d'adapter les activités en fonction des intentions pédagogiques des enseignants grâce à la définition de contraintes sur activités. Nous avons décliné cette approche pour permettre l'adaptation des activités papier (GEPPEO_p) ainsi que pour l'adaptation d'activités logicielles au sein d'EIAH (GEPPEO_s).

Pour répondre au troisième point, nous avons proposé le modèle PERSUA2 qui permet de lier les contraintes sur profils de cPMDL aux contraintes sur activités de GEPPEO. Ces liens, nommés règles d'affectation, sont ensuite hiérarchisés selon leur degré de priorité pour former une stratégie pédagogique qui sera associée à un ou plusieurs contextes d'utilisation.

Nous avons également développé Adapte, un logiciel intégré à l'environnement informatique EPROFILEA. Adapte permet de fournir des séquences de travail adaptées au profil de chaque élève et aux buts pédagogiques de l'enseignant. C'est dans ce logiciel que nous avons mis en œuvre ces propositions, ce qui nous a permis d'une part de montrer leur faisabilité technique et d'autre part, de les évaluer par des expérimentations concrètes impliquant des enseignants.

Nous avons ainsi étudié la mise en œuvre du modèle cPMDL au travers d'une expérimentation avec une enseignante. cPMDL a permis à l'enseignante de définir l'ensemble des contraintes qui lui semblaient utiles pour sélectionner les apprenants en fonction des données contenues dans

leurs profils. Toutefois, notre expérimentation n'est pas exhaustive et d'autres enseignants pourraient avoir d'autres besoins qu'il nous est difficile d'anticiper. Notre modèle étant ouvert, il est théoriquement possible de modéliser ces nouveaux besoins aisément. Néanmoins, seules des expérimentations à plus grande échelle nous permettront d'identifier l'étendue de ces besoins et éventuellement de faire émerger des verrous que nous n'avons pas identifiés jusqu'à présent. Pour cela, il nous faudra donc concevoir des scénarios expérimentaux impliquant un plus grand nombre d'enseignants et couvrant une plus grande diversité de situations pédagogiques. Par ailleurs, le langage PDML permet de modéliser des profils contenant des informations sur les apprenants à un instant donné, mais il ne permet pas de modéliser l'évolution de ces informations dans le temps. Notre modèle cPMDL reposant sur ce langage, nos contraintes sur profils ne peuvent donc pas porter sur les aspects évolutifs des profils. Or, il nous paraît important de proposer une telle possibilité aux enseignants. Pour cela, il est nécessaire d'étudier la façon d'étendre PMDL pour prendre en compte l'aspect évolutif des profils. Il nous faudra ensuite travailler avec des enseignants afin de déterminer les types de contraintes qui pourraient être définies sur ces profils évolutifs, et surtout étudier comment ces contraintes pourraient servir le processus de personnalisation.

Les évaluations menées sur l'approche GEPPETO nous ont permis de valider son principe et sa faisabilité. Toutefois, la typologie des activités papier sur laquelle s'appuie la déclinaison GEPPETO_P doit encore être validée par des experts de sciences de l'éducation. Pour les activités logicielles, nous avons défini le cadre d'application du méta-modèle AKEPI permettant de décrire les EIAH dans GEPPETO_S. Ce cadre d'application limite l'utilisation de l'approche à l'adaptation des logiciels pédagogiques proposant des activités individuelles d'apprentissage. Comme les autres modèles que nous avons proposés, le méta-modèle AKEPI est ouvert et peut donc être étendu afin de prendre en compte des EIAH proposant des situations d'apprentissage collectives et/ou collaboratives. Une perspective immédiate pour ce travail est donc de décrire cette nouvelle catégorie d'EIAH avec notre méta-modèle AKEPI. Il faudra ensuite essayer d'adapter les activités des logiciels de ce type suivant notre méthodologie, afin de vérifier que les modèles et les outils que nous proposons permettent de surmonter les nouveaux obstacles inhérents aux spécificités de ces logiciels.

Concernant le modèle PERSUA2 et sa mise en œuvre dans le logiciel Adapte, les expérimentations ont montré l'utilité du modèle et la pertinence des personnalisations proposées par notre logiciel. Ces évaluations ont également mis en évidence la complexité de la tâche demandée aux enseignants lors de la formalisation de leurs règles de personnalisation. Pour cet aspect critique du processus de personnalisation, il apparaît donc essentiel d'assister l'enseignant avec une aide et des outils appropriés. Fournir une telle assistance est un processus complexe qui soulève plusieurs questions et oriente vers de nouvelles perspectives de recherche. Comment apporter une aide pertinente sans perturber l'enseignant dans sa tâche principale ? Que nécessite une telle approche ? De quelles connaissances le système d'assistance doit-il être doté pour que l'aide apportée soit à la fois personnalisée, pertinente dans le contexte de la tâche et cohérente avec les pratiques pédagogiques de l'enseignant ? Comment exploiter les interactions entre l'enseignant et le système pour acquérir de nouvelles connaissances et donc améliorer l'efficacité de l'assistance proposée au fil des usages ? Disposer d'un outil d'assistance flexible et répondant simplement aux besoins des utilisateurs est un atout majeur pour tout outil, en particulier parce qu'il accroît les chances que l'utilisateur s'habitue à l'environnement, le fasse sien et l'exploite à la hauteur des possibilités qu'il offre. Nous considérons donc que la mise en œuvre d'un outil d'assistance « intelligent » et efficace est une perspective importante de ce travail et nous envisageons de développer un tel outil en prenant en compte les questionnements précédents.

Nous envisageons donc d'améliorer notre système en proposant une aide à deux niveaux : une aide classique guidant l'utilisateur sur les questions techniques relatives au logiciel et une aide dite « intelligente » assistant l'utilisateur dans ses choix de personnalisation. Dans le cadre de notre approche, cette aide intelligente doit posséder trois facettes liées aux trois étapes principales de la création d'un modèle de personnalisation et de son exploitation dans Adapte.

Dans la première étape, l'enseignant définit sa stratégie pédagogique en créant des règles d'affectation liant des contraintes sur profils à des contraintes sur activités. Il serait alors utile de conseiller l'enseignant sur les parties du profil à utiliser pour personnaliser les activités ou encore sur le choix des activités à fournir en fonction des parties de profils utilisées pour la personnalisation. Le système ne sera en mesure de fournir une telle aide que lorsqu'il disposera d'informations sur les profils et qu'il pourra évaluer la pertinence des différents critères en fonction des besoins. Or, actuellement, le système ne manipule pas de connaissances sur les profils mais uniquement sur leur structure. De plus, aucun modèle de ce type de connaissances n'est prévu. Cependant, il serait tout à fait envisageable d'acquérir ces connaissances à partir de l'analyse des usages du système. Une solution possible serait de mettre en place un système de raisonnement à partir de l'expérience dans une perspective d'acquisition des connaissances [Mille 2006, Cordier 2008, Cordier et al. 2009]. Un tel système observerait les utilisateurs exploitant des profils possédant une même structure et pourrait ainsi conseiller un enseignant en fonction de ses actions précédentes ou en fonction des actions observées chez d'autres enseignants. Cette solution est d'autant plus valable que l'environnement est utilisé par plusieurs enseignants. En effet, plus l'expérience accumulée sera importante, plus les conseils proposés pourront être pertinents. S'appuyant sur les expériences d'acteurs différents, un tel système proposera nécessairement une assistance et des conseils inspirés d'habitudes de travail divergentes et d'intentions pédagogiques opposées, mais cela ne constitue pas une limite car nous nous plaçons dans une perspective d'assistance et non de résolution de problème. Cette hétérogénéité des points de vue sur un problème peut même être source de richesse.

Dans la seconde étape, l'enseignant définit le contexte dans lequel il souhaite utiliser une stratégie pédagogique donnée. Il peut ensuite définir des exceptions pour certains élèves limitant par exemple le nombre d'activités à fournir à un apprenant en difficulté, changeant la police d'écriture pour un élève dyslexique ou encore demandant une séquence de travail plus longue pour des élèves ayant des facilités dans certaines matières. Ces exceptions permettent à l'enseignant d'intégrer au modèle de personnalisation des connaissances dont il dispose sur ses élèves mais qui ne sont pas représentées dans les profils. Or, il serait pertinent qu'à terme, ces éléments soient intégrés dans les profils et dans le modèle de personnalisation afin d'alléger la tâche de l'enseignant. Cela implique de déterminer les éléments à ajouter au profil et de définir des règles d'affectation liant ces nouveaux éléments à des contraintes sur la séquence d'activités. Cette perspective remet partiellement en question le choix que nous avons effectué dans notre modèle de personnalisation : les stratégies pédagogiques lient le contenu d'une séquence d'activités aux données du profil des apprenants, et c'est le contexte qui agit sur la longueur de cette séquence. Pour intégrer les exceptions au modèle, il faudrait modifier le modèle afin qu'il permette de contraindre le contenu de la séquence de travail par une règle d'affectation.

Enfin, dans la dernière étape, Adapte utilise le modèle de personnalisation d'un enseignant pour générer des séquences de travail adaptées aux profils des apprenants. Ces séquences de travail sont soumises à l'enseignant pour validation. Il peut alors en modifier le contenu (ajout ou suppression d'activité, nouvel énoncé pour une activité) ou la présentation (ordre des activités, mise en page pour les activités papier, etc.). Ces modifications effectuées par l'enseignant pourraient être exploitées comme des « sources de connaissances » pour améliorer progressivement les connaissances du système d'assistance. En effet, il est possible d'inférer, à partir de ces modifications, un ensemble de connaissances utiles à l'assistance (sous forme de

règles d'affectations, de stratégies pédagogiques, de nouveaux éléments à prendre en compte dans les profils, de contraintes sur profils, etc.). Dans une telle optique, une question ouverte est de savoir quelles méthodes et quels outils mettre en place pour acquérir de telles connaissances. C'est une problématique d'intelligence artificielle fortement orientée à la fois vers l'ingénierie des connaissances et les interactions homme-machine. En effet, mettre en place une telle solution demande une réflexion approfondie sur les mécanismes d'interaction à déployer et sur la façon de coupler outil d'assistance et moteur de raisonnement dans une optique d'aide intelligente. Cette mise en place requiert également une réflexion sur la gestion des connaissances acquises.

Ainsi, nos perspectives pour ce travail sont à la fois liées à un souhait d'amélioration concrète du système existant et à une volonté « d'aller plus loin » dans l'approche que nous proposons en construisant un outil d'assistance intelligent. Les besoins d'améliorations concrètes doivent être satisfaits afin de garantir que l'environnement soit facilement utilisable par un public diversifié et en conditions réelles. Nos perspectives à plus long terme visent à déployer une approche plus flexible qui saura s'adapter aux enseignants et évoluer avec eux. Nous pensons que cette propriété est dorénavant indispensable pour qu'un logiciel soit adopté par ses utilisateurs, particulièrement dans le domaine des EIAH.

RÉFÉRENCES BIBLIOGRAPHIQUES

- [Aamodt 2005] Aamodt, A. « Case-based reasoning and Intelligent tutoring ». SAIS-SSLS, Sweden, pp. 8-22, 20-22 avril 2005.
- [Ainsworth 2000] Ainsworth, S. « Redeem : Its authoring tools and human teaching ». Workshop on Modelling Human Teaching Tactics - ITS2000, 2000.
- [Alexander et al. 1977] Alexander, C., Ishikawa, S., Silverstein, M., Jacobson, M., Fikdahl-King, I., Angal, S. (1977). « A pattern language: towns, buildings, construction ». New York: Oxford University Press, 1977.
- [Allen et al. 1998] Allen, P., Frost, S. « Component-based development for enterprise systems: applying the SELECT perspective ». New York, USA, 1998.
- [Anderson et al. 1985] Anderson, J. R., Reiser, B.-J. « The LISP tutor ». Byte, vol. 10 n°4, pp. 159-175, avril 1985.
- [Anjaneyulu 1997] Anjaneyulu, K. S. R. « Concept Level Modelling on the WWW ». Workshop "Intelligent Educational Systems on the World Wide Web" at the 8th World Conference on Artificial Intelligence in Education (AIED'97) Kobe, Japan, 18-22 août 1997.
- [Antoniadis et al. 2004] Antoniadis, G., Ponton, C. « MIRTO : Un système au service de l'enseignement des langues ». *UNTELE'2004*, Compiègne, 16-20 mars 2004.
- [Arnaud 1998] Arnaud, P. « Chimie Physique – Cours », 4ème Édition Revue Et Corrigée, 1998.
- [Béguin et al. 2000] Béguin, P., Rabardel, P. « Designing for instrument mediated activity », *Information technology in human activity, Designing for instrument mediated activity, Scandinavian Journal of Information Systems* 12: 173-190, 2000.
- [Bouhineau et al. 2005a] Bouhineau, D., Bronner, A., Chaachoua, H., Nicaud, J.-F. « Patrons d'exercices pour APLUSIX : Une étape du développement de l'EIAH occasion d'un travail entre didacticiens et informaticiens ». *Environnements Informatiques pour l'Apprentissage Humain (EIAH'2005)*, Montpellier, France, pp. 377-382, 25-27 mai 2005.
- [Bouhineau et al. 2005b] Bouhineau, D., Bronner, A., Chaachoua, H., Nicaud, J.-F. « Informaticiens et didacticiens peuvent-ils travailler ensemble ? Une étape du développement de l'EIAH APLUSIX : la mise en place de patrons d'exercices et d'une carte de tests », *Les cahiers Leibniz*, Grenoble, France, mai 2005.
- [Bouhineau et al. 2006] Bouhineau, D., Nicaud, J.-F. « Aplusix, un EIAH de l'algèbre ». *Environnements informatiques pour l'apprentissage humain (Traité IC2, série Cognition et traitement de l'information)*. M. Grandbastien, J.-M. L., pp. 333-350, 2006.

- [Bourda 2001] Bourda, Y. « Objets pédagogiques, vous avez dit objets pédagogiques ? ». Congrès Gutenberg, pp. 71-79, mai 2001.
- [Bourdeau et al. 2002] Bourdeau, J., Mizoguchi, R. « Collaborative ontological engineering of instructional design knowledge for an its authoring environment ». Artificial Intelligence in Education (AIED'2002), 2002.
- [Bourguin 2000] Bourguin, G. « Un support informatique à l'activité coopérative fondé sur la Théorie de l'Activité : Le projet DARE », Thèse de doctorat, Université des Sciences et Technologies de Lille, 2000.
- [Brady et al. 2005] Brady, A., Conlan, O., Wade, V. « Towards the Dynamic Personalized Selection and Creation of Learning Objects ». World Conference on E-Learning in Corporate, Government, Healthcare, and Higher Education (E-Learn 2005), AACE, pp. 1903-1909, 2005.
- [Bruillard 1997] Bruillard, E. « Les machines à enseigner ». Paris, France, Hermès, 1997.
- [Bruillard et al. 2000] Bruillard, E., Delozanne, E., Leroux, P., Delannoy, P., Dubourg, X., Jacoboni, P., Lehuen, J., Luzzati, D., Teutsch, P. « Quinze ans de recherche informatique sur les sciences et techniques éducatives au LIUM ». Sciences et Techniques Educatives, vol. 7 n°1, pp. 87-145, 2000.
- [Brusilovsky et al. 1996] Brusilovsky, P., Schwarz, E., Weber, G. « An intelligent tutoring system on World Wide Web ». Intelligent Tutoring Systems. Frasson, C., Gauthier, G. and Lesgold, A. Berlin, Springer Verlag. volume 1086, pp. 261-269, 1996.
- [Brusilovsky et al. 1998] Brusilovsky, P., Eklund, J., Schwarz, E. « Web-based Education for All: A Tool for Development Adaptive Courseware ». Seventh International World Wide Web Conference, Computer Networks and ISDN Systems, pp. 291-300, 14-18 avril 1998.
- [Brusilovsky 1999] Brusilovsky, P. « Adaptive and Intelligent Technologies for Web-based Education ». Künstliche Intelligenz, Special Issue on Intelligent Systems and Teleteaching, pp. 19-25, 1999.
- [Bull et al. 2003] Bull, S., McEvoy, A. T., Reid, E. « Learner Models to Promote Reflection in Combined Desktop PC / Mobile Intelligent Learning Environments ». Workshop on Learner Modelling for Reflection, International Conference on Artificial Intelligence in Education 2003 (AIED'2003), Sydney, Australia, pp. 199-208, 2003.
- [Burnier 2000] Burnier, M. « D'échec en échec jusqu'au succès ? 35 de projets d'Enseignement Assisté par Ordinateur ». Terminal, vol. 83, pp. 15-32, 2000.
- [Burton 1982] Burton, R. R. « Diagnosing bugs in a simple procedural skill ». Intelligent Tutoring Systems. London, Academic Press, 1982.

- [Chen et al. 1998] Chen, W., Hayashi, Y., Jin, L., Ikeda, M., Mizoguchi, R. « An Ontology-based Intelligent Authoring Tool ». International Conference on Computers in Education (ICCE'98), pp. 41-49, 1998.
- [Choquet 2007] Choquet, C. « Ingénierie et réingénierie des EIAH : l'approche REDiM ». Habilitation à Diriger des Recherches en Informatique, Laboratoire d'Informatique de l'Université du Maine (LIUM), 10 décembre 2007.
- [Claudal 2007] Claudal, Y. « Jeux de culture générale », 2007.
- [Cogne et al. 1998] Cogne, A., David, J.-P., Lacombe, C. « Production d'exercices hypermédias et mise en œuvre pédagogique ». 8èmes Journées Informatique et Pédagogie des Sciences Physiques, Montpellier, 1998.
- [Conejo et al. 2004] Conejo, R., Guzmán, E., Millán, E., Trella, M., Pérez-De-La-Cruz, J. L., Ríos, A. « SIETTE : A Web-Based Tool for Adaptive Testing ». International Journal of Artificial Intelligence in Education (IJAIED), vol. 14, pp. 29-61, 2004.
- [Cordier 2008] Cordier, A. « Interactive and Opportunistic Knowledge Acquisition in Case-Based Reasoning ». Thèse de doctorat en Informatique, Université de Lyon 1, 13 novembre 2008.
- [Cordier et al. 2009] Cordier, A., Mascret, B., Mille, A. « Extending Case-Based Reasoning with Traces ». *Grand Challenges for reasoning from experiences, Workshop at IJCAI'09*, Pasadena, CA. 2009.
- [David et al. 1996] David, J.-P., Cogne, A., Dutel, A. « Hypermedia exercises prototyping and modelising ». Computer Aided Learning and Instruction in Science and Engineering. Heidelberg, S. B., pp. 252-260, 1996.
- [De Bra et al. 1998] De Bra, P., Calvi, L. « AHA! An open Adaptive Hypermedia Architecture ». The New Review of Hypermedia and Multimedia, vol. 4, pp. 115-139, 1998.
- [De Jong 2004] De Jong, T. « Learning Complex Domains and Complex tasks, the Promise of Simulation Based Training ». International Conference on Computer Aided Learning In Engineering education (CALIE'2004), Grenoble, France, pp. 17-23, 2004.
- [De la Passardière et al. 2003] De la Passardière, B., Grandbastien, M. « Présentation de LOM v1.0 standard IEEE, 2002 ». Revue Sciences et Techniques Educatives, Hors série 2003 "Ressources numériques, XML et éducation", pp. 211-218, avril 2003.
- [De la Passardière et al. 2004] De la Passardière, B., Jarraud, P. « ManUeL, un profil d'application de LOM pour C@mpuSciences ». Sciences et Technologies de l'Information et de la Communication pour l'Education et la Formation (STICEF), vol. 11, pp. 11-57, 2004.
- [Delestre et al. 2004] Delestre, N., Bourda, Y. « Utilisation de la norme ISO11179 pour améliorer l'interopérabilité entre les différents schémas de

- métadonnées pédagogiques ». Technologies de l'Information et de la Connaissance dans l'Enseignement Supérieur et l'Industrie (TICE'2004), Compiègne, France, pp. 191-197, 20-22 octobre 2004.
- [Delozanne et al. 2002] Delozanne, E., Grugeon, B., Jacoboni, P. « Analyses de l'activité et IHM pour l'éducation ». Interaction Homme Machine 2002, Poitiers, France, pp. 25-32, 2002.
- [Delozanne et al. 2007] Delozanne, E., Le Calvez, F., Merceron, A., Labat, J.-M. « Design Patterns en EIAH : vers un langage de Patterns pour l'évaluation des apprenants ». Sciences et Technologies de l'Information et de la Communication pour l'Education et la Formation (STICEF), vol. 14, pp. 45-80, 2007.
- [Duclosson et al. 2005a] Duclosson, N., Jean-Daubias, S., Riot, S. « AMBRE-enseignant : un module partenaire de l'enseignant pour créer des problèmes ». Environnements Informatiques pour l'Apprentissage Humain (EIAH'2005), Montpellier, France, pp. 353-358, 25-27 mai 2005.
- [Duclosson et al. 2005b] Duclosson, N., Jean-Daubias, S., Riot, S. « AMBRE-enseignant : un module partenaire de l'enseignant pour faciliter l'intégration de AMBRE en classe », Rapport de recherche, LIRIS, Villeurbanne, France, 2005.
- [Engeström 1987] Engeström, Y. « Learning by expanding: an activity-theoretical approach to developmental research », Helsinki, Orienta-Konsultit Oy, 1987.
- [Eyssautier-Bavay 2008] Eyssautier-Bavay, C. « Modèles, langage et outils pour la réutilisation de profils d'apprenants ». Thèse de doctorat en Informatique, Université Joseph Fourier, 26 mai 2008.
- [Eyssautier-Bavay et al. 2009] Eyssautier-Bavay, C., Jean-Daubias, S. « Un modèle de processus de gestion de profils d'apprenants ». Environnements Informatiques pour l'Apprentissage Humain (EIAH'2009), Le Mans, France, pp. 69-76, 23-26 juin 2009.
- [Futtersack et al. 2000] Futtersack, M., Labat, J.-M. « Quelle planification pédagogique dans les EIAH ? ». Sciences et Technologies de l'Information et de la Communication pour l'Éducation et la Formation (STICEF), vol. 7, pp. 165-177, 2000.
- [Girard et al. 2007] Girard, S., Johnson, H. « DividingQuest: opening the learner model to teachers ». Environnements Informatiques pour l'Apprentissage Humain (EIAH'2007), Lausanne, Suisse, pp. 329-334, 27-29 juin 2007.
- [Giroire 1989] Giroire, H. « Un système à base de connaissances pour la génération d'exercices dans les domaines liés au monde réel ». Thèse de l'Université, Université Paris VI, 16 janvier 1989.
- [Giroux et al. 1995] Giroux, S., Pachet, F., Paquette, G., Girard, J. « Des Systèmes Conseillers Epiphytes ». Revue d'Intelligence Artificielle, vol. 9 n°2, pp. 165-190, 1995.

- [Grandbastien et al. 2002] Grandbastien, M., Bouyt, B., Claes, G., Bourda, Y., Duval, E. « Les ressources numériques : Enjeux de la normalisation ». Technologies de l'Information et de la Connaissance dans l'Enseignement Supérieur et l'Industrie (TICE'2002), Villeurbanne, France, 13-15 novembre 2002.
- [Gruber 1993] Gruber, T. R. « Towards Principles for the Design of Ontologies Used for Knowledge Sharing ». Formal Ontology in Conceptual Analysis and Knowledge Representation, 1993.
- [Grugeon 1995] Grugeon, B. « Étude des rapports institutionnels et des rapports personnels des élèves à l'algèbre élémentaire dans la transition entre deux cycles d'enseignement : BEP et Première G ». Thèse de doctorat, Université Paris 7, 1995.
- [Guéraud et al. 2007] Guéraud, V., Michelet, S., Adam, J.-M. « Suivi de classe à distance : propositions génériques et expérimentation en électricité ». Environnements Informatiques pour l'Apprentissage Humain (EIAH'2007), Lausanne, Suisse, pp. 167-178, 27-29 juin 2007.
- [Guin-Duclosson et al. 2002] Guin-Duclosson, N., Jean-Daubias, S., Nogry, S. « The AMBRE ILE: How to Use Case-Based Reasoning to Teach Methods ». Intelligent Tutoring Systems (ITS'2002), San Sebastian (Espagne) & Biarritz (France), Springer, Lecture Notes in Computer Science, pp. 782-791, 2002.
- [Guin 1997] Guin, N. « Reformuler et classer un problème pour le résoudre. L'architecture SYRCLAD et son application à quatre domaines ». Thèse de doctorat en Informatique, spécialité Intelligence Artificielle, Université Paris 6, 12 décembre 1997.
- [Hibou et al. 2003] Hibou, M., Labat, J.-M., Spagnol, J.-P. « Génération de feuilles d'exercices de géométrie à l'aide d'énoncés indexés automatiquement ». Environnements Informatiques pour l'Apprentissage Humain (EIAH'2003), Strasbourg, France, pp. 247-258, 15-17 avril 2003.
- [Hofmann 1990] Hofmann, Y. « Introduction de l'informatique dans la formation en soins infirmiers ». Informatique et Santé, vol. 3, pp. 17-24, 1990.
- [Hû 2001] Hû, O. « Contribution à l'évaluation des logiciels multimédias pédagogiques ». Thèse de doctorat en Informatique, Université de technologie de Compiègne, 30 août 2001.
- [J'ADE 2007a] J'ADE. « Guide d'installation et d'utilisation », Ministère Français de l'éducation nationale, de l'enseignement supérieur et de la recherche, 2007.
- [J'ADE 2007b] J'ADE. « Les indicateurs », Ministère Français de l'éducation nationale, de l'enseignement supérieur et de la recherche, 2007.
- [Jean-Daubias 2003] Jean-Daubias, S. « Exploitation de profils d'apprenants ». Environnements Informatiques pour l'Apprentissage Humain (EIAH'2003), Strasbourg, France, pp. 535-538, 15-17 avril 2003.

- [Jean-Daubias 2004] Jean-Daubias, S. « De l'intégration de chercheurs, d'experts, d'enseignants et d'apprenants à la conception d'un EIAH ». Technologies de l'Information et de la Connaissance dans l'Enseignement Supérieur et l'Industrie (TICE'2004), Compiègne, France, pp. 290-297, 20-22 octobre 2004.
- [Jean-Daubias et al. 2005] Jean-Daubias, S., Eyssautier-Bavay, C. « An environment helping teachers to track students' competencies ». Workshop LEMORE, Artificial Intelligence in Education (AIED'2005), Pays-Bas, pp. 19-23, 2005.
- [Jean-Daubias et al. 2009a] Jean-Daubias, S., Eyssautier-Bavay, C., Lefevre, M. « Uniformisation de la structure de profils d'apprenants issus de sources hétérogènes ». Environnements Informatiques pour l'Apprentissage Humain (EIAH'2009), Le Mans, France, pp. 77-84, 23-26 juin 2009.
- [Jean-Daubias et al. 2009b] Jean-Daubias, S., Guin, N. « AMBRE-teacher: a module helping teachers to generate problems ». 2nd Workshop on Question Generation, International Conférence on Artificial Intelligence in Education (AIED 2009), Brighton, Grande-Bretagne, AIED 2009 Workshops Proceedings 1, pp. 38-42, 6-7 juillet 2009.
- [Jean-Daubias et al. 2009c] Jean-Daubias, S., Lefevre, M., Guin, N. « Generation of exercises within the PERLEA project ». 2nd Workshop on Question Generation, International Conférence on Artificial Intelligence in Education (AIED 2009), Brighton, Grande-Bretagne, AIED 2009 Workshops Proceedings 1, pp. 38-42, 6-7 juillet 2009.
- [Jean-Daubias et al. 2009d] Jean-Daubias, S., Lefevre, M., Guin, N. « Adapte, un outil générique pour proposer des activités pédagogiques personnalisées ». Workshop Prise en Compte de l'Utilisateur dans les Systèmes d'Information (PeCUSI), INFORSID 2009, Toulouse, France, pp. 51-62, 26 mai 2009.
- [Jean-Daubias et al. 2009e] Jean-Daubias, S., Lefevre, M., Guin, N. « Generation of paper and pencil activities within the PERLEA project », Rapport de recherche RR-LIRIS-2009-016 (version longue de l'article du Workshop on Question Generation, AIED 2009), LIRIS, Villeurbanne, France, 2009.
- [Jean 2000] Jean, S. « PÉPITE : un système d'assistance au diagnostic de compétences ». Thèse de doctorat, Université du Maine, 21 janvier 2000.
- [Johnson 1986] Johnson, W. L. « Intention-based diagnosis of errors in novice programs ». Palo Alto, CA, Morgan Kaufmann Series In Research Notes In Artificial Intelligence, 1986.
- [Kaptelinin 1996] Kaptelinin, V. « Activity Theory: Implications for Human-Computer Interaction », Context and consciousness : activity theory and Human-Computer Interaction. A., N. B. Cambridge, Ma, MIT Press: 103-116, 1996.

- [Kay et al. 1997] Kay, J., Kummerfeld, B. « User models for customized hypertext ». *Intelligent hypertext: Advanced techniques for the World Wide Web*. Nicholas, C. a. M., J., Springer-Verlag, Berlin. volume 1326, pp. 47-69, 1997.
- [Keenoy 2003] Keenoy, K. « SeLeNe-Preliminary Report: Learning Objects, Meta-Data and Standards », 2003.
- [Keenoy et al. 2004] Keeney, K., De Freitas, S., Levene, M., Jones, A., Brasher, A., Waycott, J., Kaszas, P., Turcsanyi-Szabo, M., Montandon, L. « Personalised trails and learner profiling within e-Learning environments », *Kaleidoscope D22.4.1*, 2004.
- [Koper 2001] Koper, R. « Modeling units of study from a pedagogical perspective. The pedagogical meta-model behind EML », *Rapport de recherche*, Open University of the Netherlands, juin 2001.
- [Lefevre et al. 2007a] Lefevre, M., Jean-Daubias, S., Guin, N. « Génération de feuilles d'exercices adaptées aux profils d'apprenants dans le cadre du projet PERLEA ». Poster, 3ème Conférence en Environnements Informatiques pour l'Apprentissage Humain (EIAH'2007), Lausanne, Suisse, 27-29 juin 2007.
- [Lefevre et al. 2007b] Lefevre, M., Jean-Daubias, S., Guin, N. « Adapte, un module pour proposer des activités personnalisées », *Rapport de recherche RR-LIRIS-2007-027*, LIRIS, Villeurbanne, France, 2007.
- [Lefevre et al. 2007c] Lefevre, M., Jean-Daubias, S., Guin, N. « Génération d'exercices au sein du projet PERLEA », *Rapport de recherche RR-LIRIS-2007-030*, LIRIS, Villeurbanne, France, 2007.
- [Lefevre 2008] Lefevre, M. « Adapte, un outil pour proposer des activités personnalisées au sein du projet PERLEA ». Poster, 2nde Rencontres Jeunes Chercheur en Environnements Informatiques pour l'Apprentissage Humain (RJC EIAH 2008), Lille, France, pp. 145-146, 15-16 mai 2008.
- [Lefevre et al. 2008a] Lefevre, M., Guin, N., Jean-Daubias, S. « Adapte, a tool for the teacher to personalize activities », *Rapport de recherche RR-LIRIS-2008-008* (version longue de l'article d'ITS 2008), LIRIS, Villeurbanne, France, 2008.
- [Lefevre et al. 2008b] Lefevre, M., Guin, N., Jean-Daubias, S. « Adapte, a tool for the teacher to personalize activities ». 9th International Conference on Intelligent Tutoring Systems (ITS'2008), Montréal, Canada, ISBN 978-3-540-69130-3, pp. 699-701, 23-27 juin 2008.
- [Lefevre et al. 2008c] Lefevre, M., Jean-Daubias, S., Guin, N. « Generation of exercises within the PERLEA project », *Rapport de recherche RR-LIRIS-2008-011*, LIRIS, Villeurbanne, France, 2008.
- [Lefevre et al. 2009a] Lefevre, M., Cordier, A., Jean-Daubias, S., Guin, N. « A Teacher-dedicated Tool Supporting Personalization of Activities ». *ED-MEDIA 2009 - World Conference on Educational Multimedia*,

- Hypermedia & Telecommunications, Honolulu, Hawaii, pp. 1136-1141, 22-26 juin 2009.
- [Lefevre et al. 2009b] Lefevre, M., Jean-Daubias, S., Guin, N. « Supporting Acquisition of Knowledge to Personalize Interactive Learning Environments through a Meta-Model ». 17th International Conference on Computers in Education (ICCE 2009), Hong Kong, 30 novembre - 4 decembre 2009.
- [Lefevre et al. 2009c] Lefevre, M., Jean-Daubias, S., Guin, N. « Personnaliser des séquences de travail à partir de profils d'apprenants ». Poster, 4ème Conférence en Environnements Informatiques pour l'Apprentissage Humain (EIAH'2009), Le Mans, France, 23-26 juin 2009.
- [Lefevre et al. 2009d] Lefevre, M., Jean-Daubias, S., Guin, N. « Generation of pencil and paper exercises to personalize learners' work sequences: typology of exercises and meta-architecture for generators ». E-Learn 2009 (World Conference on E-Learning in Corporate, Government, Healthcare & Higher Education), Vancouver, Canada, 26-30 octobre 2009.
- [Lefevre et al. 2009e] Lefevre, M., Mille, A., Jean-Daubias, S., Guin, N. « A Meta-Model to Acquire Relevant Knowledge for Interactive Learning Environments Personalization ». Adaptive 2009, Athènes, Grèce, 15-20 novembre 2009.
- [Leroux 1998] Leroux, P. « RoboTeach - Guide d'utilisation », Laboratoire d'Informatique de l'Université du Maine, mars 1998.
- [Leroux 2002] Leroux, P. « Machines partenaires des apprenants et des enseignants - Étude dans le cadre d'environnements supports de projets pédagogiques ». Habilitation à Diriger des Recherches en Informatique, Université du Maine, 20 décembre 2002.
- [Leroux 2006] Leroux, P. « Micromondes et robotique pédagogique ». Environnements informatiques pour l'apprentissage humain (Traité IC2, série Cognition et traitement de l'information). M. Grandbastien, J.-M. Labat, pp. 311-332, 2006.
- [Leroux et al. 2007] Leroux, P., Jean-Daubias, S. « EIAH partenaires des acteurs de la situation d'apprentissage ». Environnements informatisés et ressources numériques pour l'apprentissage : conception et usages, regards croisés. Hermès-Lavoisier, Monique Baron, Dominique Guin, Luc Trouche, pp. 107-136, 2007.
- [Lopez-Cuadrado et al. 2007] Lopez-Cuadrado, J., Armendariz, A., Lopistéguy, P., Dagorret, P. « Calibrage d'une banque d'items pour la génération de tests d'évaluation adaptatifs - Un exemple dans le cadre de "Hezinet" ». Atelier Prise en Compte de l'Utilisateur dans les Systèmes d'Information - INFORSID 2007, Perros-Guirec, France, 22 mai 2007.

- [López et al. 1998] López, J. M., Millán, E., Pérez-De-La-Cruz, J. L., Triguero, F. « ILESA: a Web-based Intelligent Learning Environment for the Simplex Algorithm ». 4th International conference on Computer Aided Learning and Instruction in Science and Engineering (CALISCE'98), Göteborg, Sweden, pp. 399-406, 1998.
- [Marcelle-Rosselle 2001] Marcelle-Rosselle, M. « Conception d'un atelier d'expérimentation de logiciels éducatifs - Application en géométrie ». Thèse de doctorat en Informatique, Université Henri Poincaré - Nancy 1, 21 septembre 2001.
- [Melis et al. 2001] Melis, E., Andrés, E., Büdenbender, J., Frischauf, A., Gogvadze, G., Libbrecht, P., Pollet, M., Ullrich, C. « ActiveMath: A Generic and Adaptive Web-Based Learning Environment ». International Journal of Artificial Intelligence in Education (IJAIED), vol. 12, pp. 385-407, 2001.
- [Mille 2006] Mille, A. « From case-based reasoning to traces-based reasoning ». Annual Reviews in Control, vol. 30 n°2, pp. 223-232, ELSEVIER, ISSN 1367-5788, 2006.
- [Mitrovic 1998] Mitrovic, A. « A Knowledge-Based Teaching System for SQL ». Association for the Advancement of Computing in Education (AACE), 1998.
- [Mostow et al. 2001] Mostow, J., Aist, G. « Evaluating tutors that listen: An overview of Project LISTEN. ». Smart Machines in Education. Menlo Park, CA, MIT/AAAI Press, pp. 169-234, 2001.
- [Mostow et al. 2004] Mostow, J., Beck, J., Bey, J., Cuneo, A., Sison, J., Tobin, B., Valeri, J. « Using automated questions to assess reading comprehension, vocabulary, and effects of tutorial interventions ». Technology, Instruction, Cognition and Learning, vol. 2, pp. 97-134, 2004.
- [Mufti-Alchawafa et al. 2004] Mufti-Alchawafa, D., Luengo, V., Vadcard, L. « Architecture d'un environnement d'aide à l'apprentissage de la chirurgie orthopédique ». Technologies de l'Information et de la Connaissance dans l'Enseignement Supérieur et l'Industrie (TICE'2004), Compiègne, France, 20-22 octobre 2004.
- [Murray et al. 1998] Murray, T., Condit, C., Haugsjaa, E. « MetaLinks: A preliminary framework for concept-based adaptive hypermedia ». Workshop "WWW-Based Tutoring" at 4th International Conference on Intelligent Tutoring Systems (ITS'98), San Antonio, TX, 1998.
- [Murray 1999] Murray, T. « Authoring intelligent tutoring systems : an analysis of the state of the art ». International Journal of Artificial Intelligence in Education, vol. 10, pp. 98-129, 1999.
- [Murray 2003a] Murray, T. « Eon: Authoring Tools for Content, Instructional Strategy, Student Model, and Interface Design ». Authoring Tools for Advanced Technology Learning Environments. T. Murray, S. B., and S. Ainsworth, editors, Kluwer Academic Publisher, 2003.

- [Murray 2003b] Murray, T. « MetaLinks: Authoring and Affordances for Conceptual and Narrative Flow in Adaptive Hyperbooks ». *International Journal of Artificial Intelligence in Education (IJAIED)*, vol. 13, pp. 199-233, 2003.
- [Nicaud 1993] Nicaud, J.-F. « Le projet Aplusix ». *La revue de l'EPI*, vol. 72, pp. 213-223, décembre 1993.
- [Nicaud et al. 2003] Nicaud, J.-F., Bouhineau, D., Chaachoua, H., Huguet, T., Bronner, A. « A computer program for the learning of algebra: description and first experiment ». *Eleventh International PEG Conference*, St. Petersburg, Russie, juin 2003.
- [Nodenot 2005] Nodenot, T. « Contribution à l'ingénierie dirigée par les modèles en EIAH : le cas des situations-problèmes coopératives ». *Habilitation à Diriger des Recherches en Informatique, Laboratoire d'Informatique de l'Université de Pau et des Pays de l'Adour (LIUPPA)*, 30 novembre 2005.
- [Nodenot 2006] Nodenot, T. « Étude du potentiel du langage IMS-LD pour scénariser des situations d'apprentissage : résultats et propositions ». *Colloque SCENARIOS 2006*, Lyon, pp. 57-63, 14 avril 2006.
- [Nogry et al. 2004] Nogry, S., Jean-Daubias, S., Duclosson, N. « ITS Evaluation in Classroom: The Case of AMBRE-AWP ». *Intelligent Tutoring Systems (ITS'2004)*, Lacey, Brasil, pp. 511-520, 2004.
- [Oubahssi 2005] Oubahssi, L. « Conception de plates-formes logicielles pour la formation à distance, présentant des propriétés d'adaptabilité à différentes catégories d'utilisateurs et d'interopérabilité avec d'autres environnements logiciels ». *Thèse de doctorat en Informatique, Université René Descartes - Paris V*, décembre 2005.
- [Paquette 1992] Paquette, G. « Metaknowledge in the LOUTI development system ». *Canadian Society for Computational Study of Intelligence (CSCSI-92)*, Vancouver, mai 1992.
- [Paquette et al. 1993a] Paquette, G., Aubin, C., Bourdeau, J., Crevier, F., Paquin, C., Ruelland, D. « Modélisation des connaissances de design pédagogique dans un atelier de génie didactique (AGD) », *Rapport de recherche, LICEF, Télé-université*, décembre 1993.
- [Paquette et al. 1993b] Paquette, G., Bergeron, G., Bourdeau, J. « The virtual classroom revisited ». *Conference Teleteaching'93*, Trondheim, Norvège, North-Holland Publishing Co., 0-444-81585-6 pp. 639-646, août 1993.
- [Paquette et al. 1994] Paquette, G., Pachet, F., Giroux, S. « Épitalk, un outil générique pour la construction de systèmes conseillers ». *Sciences et Techniques Educatives*, vol. 1 n°3, septembre 1994.
- [Paquette et al. 1996a] Paquette, G., Girard, J. « AGD: a course engineering support system ». *3rd International Conference on Intelligent Tutoring Systems (ITS-96)*, Montreal, Canada, pp. 382-391, 1996.

- [Paquette et al. 1996b] Paquette, G., Pachet, F., Giroux, S., Girard, J. « EpiTalk, a generic tool for the development of advisor systems ». *International Journal of Artificial Intelligence in Education*, vol. 7, pp. 349-370, 1996.
- [Paquette et al. 1997a] Paquette, G., Aubin, C., Crevier, F. « Design and Implementation of Interactive TeleLearning Scenarios ». *International Council for Distance Education (ICDE'97)*, PennState University, USA, juin 1997.
- [Paquette et al. 1997b] Paquette, G., Crevier, F., Aubin, C. « Méthode d'ingénierie d'un système d'apprentissage (MISA) ». *Revue Informations In Cognito*, vol. 8, 1997.
- [Paquette et al. 2002] 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*, vol. 9 n°2-3, 2002.
- [Pastre et al. 1989] Pastre, D., Bledsoe, W. « MUSCADET : An Automatic Theorem Proving System using Knowledge and Metaknowledge in Mathematics ». *Artificial Intelligence Journal*, vol. 38 n°3, pp. 257-318, 1989.
- [Pecego 1996] Pecego, G. « Un générateur d'énoncés de problèmes de géométrie ». *Dixième congrès RFIA*, Rennes, France, pp. 1009-1017, janvier 1996.
- [Pecego 1998] Pecego, G. « SYGEP, un Système de Génération d'Énoncés de Problèmes dans des domaines variés ». Thèse de l'Université, Université Paris VI, 12 juin 1998.
- [Pernin 2003] Pernin, J.-P. « Objets pédagogiques : unités d'apprentissage, activités ou ressources ? ». *Revue Sciences et Techniques Educatives, Hors série 2003 « Ressources numériques, XML et éducation »*, pp. 179-210, avril 2003.
- [Pernin et al. 2004] Pernin, J.-P., Lejeune, A. « Dispositifs d'apprentissage Instrumentés par les Technologies : vers une ingénierie centrée sur les scénarios ». *Technologies de l'Information et de la Connaissance dans l'Enseignement Supérieur et l'Industrie (TICE'2004)*, Compiègne, France, 20-22 octobre 2004.
- [Perriault 1989] Perriault, J. « La logique de l'usage, essai sur les machines à communiquer », 1989.
- [Prévit et al. 2007] Prévit, D., Delozanne, E., Grugeon, B. « Génération d'exercices de diagnostic de compétences en algèbre ». *Environnements Informatiques pour l'Apprentissage Humain (EIAH'2007)*, Lausanne, Suisse, pp. 545-556, 27-29 juin 2007.
- [Py 2001] Py, D. « Environnements interactifs d'apprentissage et démonstration en géométrie ». Habilitation à diriger des recherches, Université de Rennes 1 - Institut de Formation Supérieure en Informatique et en Communication, 5 juillet 2001.

- [Rabardel 1995] Rabardel, P. « Les hommes et les technologies - approche cognitive des instruments contemporains », 1995.
- [Ramandalahy et al. 2009] Ramandalahy, M., Vidal, P., Huet, P., Broisin, J. «Partage et réutilisation d'un profil ouvert de l'apprenant ». *Environnements Informatiques pour l'Apprentissage Humain (EIAH'2009)*, Le Mans, France, pp. 85-92, 23-26 juin 2009.
- [Rasseneur-Coffinet 2004] Rasseneur-Coffinet, D. « Saafir : un environnement support à l'appropriation d'une formation à distance par l'apprenant ». Thèse de doctorat en Informatique, Université du Maine, 5 novembre 2004.
- [Rasseneur et al. 2003] Rasseneur, D., Jacoboni, P., Tchounikine, P. « Visualisation multi points de vue d'un FOAD » . *Environnements Informatiques pour l'Apprentissage Humain (EIAH'2003)*, Strasbourg, France, pp. 559-562, 15-17 avril 2003.
- [Rebaï et al. 2004] Rebaï, I., Labat, J.-M. « Des métadonnées pour la description des composants logiciels pédagogiques ». *Technologies de l'Information et de la Connaissance dans l'Enseignement Supérieur et l'Industrie (TICE'2004)*, Compiègne, France, pp. 80-87, 20-22 octobre 2004.
- [Rebaï 2006] Rebaï, I. « Conception et développement d'une plate-forme permettant la capitalisation de composants logiciels pour les environnements d'apprentissage humain ». Thèse de doctorat en Informatique, l'Université René Descartes – Paris V, 11 janvier 2006.
- [Ríos et al. 1999] Ríos, A., Millán, E., Trella, M., Pérez-De-La-Cruz, J. L., Conejo, R. « Internet Based Evaluation System ». *9th World Conference of Artificial Intelligence and Education (AIED'99)*, Le Mans, IOS Press, Amsterdam, pp. 387-394, 1999.
- [Ritter et al. 1995] Ritter, S., Koedinger, K. R. « Toward lightweight tutoring agents ». *World Conference on Artificial Intelligence in Education (AIED'05)*, pp. 91-98, 1995.
- [Ritter et al. 1996] Ritter, S., Koedinger, K. R. « An Architecture for Plug-in Tutor Agents ». *Journal of Artificial Intelligence in Education*, vol. 7 n°3-4, pp. 315-347, 1996.
- [Ritter et al. 1998] Ritter, S., Blessing, S. B. « Authoring Tools for Component-Based Learning Environments ». *Journal of the Learning Sciences*, vol. 7 n°1, pp. 107 - 132, 1998.
- [Rosselle 2003] Rosselle, M. « Étude des objectifs d'une plate-forme de coopération pour les EIAH ». *Environnements Informatiques pour l'Apprentissage Humain (EIAH'2003)*, Strasbourg, France, 15-17 avril 2003.
- [Rosselle et al. 2004] Rosselle, M., Grandbastien, M. « Towards interoperable ILEs: results from a case study in the geometry domain ». *International*

- Conference on Computer Aided Learning In Engineering education (CALIE'2004)*, Grenoble, France, 2004.
- [Rosselle et al. 2005] Rosselle, M., Bessagnet, M.-N., Carron, T. « Comment intégrer des logiciels issus de la recherche en EIAH ? ». *Sciences et Technologies de l'Information et de la Communication pour l'Éducation et la Formation (STICEF)*, vol. 12, 2005.
- [Rueda et al. 2006] Rueda, U., Larrañaga, M., Arruarte, A., Elorriaga, J.-A. « DynMap+ : A Concept Mapping Approach to Visualize Group Student Models », *Innovative Approaches for Learning and Knowledge Sharing, EC-TEL'06*, Crete, Grèce, pp. 383-397, 2006.
- [Rus et al. 2009] Rus, V., Graesser, A. C. « The Question Generation Shared Task and Evaluation Challenge », Workshop Report. ISBN:978-0-615-27428-7., 2009.
- [Sørmo et al. 2002] Sørmo, F., Aamodt, A. « Knowledge communication and CBR ». *6th European Conference on Case-Based Reasoning (ECCBR 2002)*, pp. 47-59, 2002.
- [Spagnol 2001] Spagnol, J.-P. « Automatisation du raisonnement et de la rédaction de preuves en géométrie de l'enseignement secondaire ». Thèse de doctorat en informatique, Université René Descartes - Paris V, 12 octobre 2001.
- [Tchounikine 2009] Tchounikine, P. « Précis de recherche en ingénierie des EIAH », en ligne sur le Web, 2009 (<http://membres-liglab.imag.fr/tchounikine/Precis.html>).
- [Türker et al. 2006] Türker, M. A., Görgün, İ., Conlan, O. « The Challenge of Content Creation to facilitate Personalized eLearning Experiences ». *International Journal on E-Learning (IJEL)*, vol. 5 n°1, pp. 11-17, janvier 2006.
- [Van Joolingen et al. 2003] Van Joolingen, W.-R., De Jong, T. « Simquest: Authoring educational simulations ». *Authoring tools for advanced technology educational software: Toward cost-effective production of adaptive, interactive, and intelligent educational software*, T. Murray, S. Blessing & S. Ainsworth (Eds), Kluwer Academic Publishers, pp. 1-31, 2003.
- [Van Marcke 1998] Van Marcke, K. « GTE: An epistemological approach to instructional modeling ». *Instructional Science*, vol.26 n°3-4, pp. 147-191, 1998.
- [VanLehn et al. 2005] VanLehn, K., Lynch, C., Schulze, K., Shapiro, J. A., Shelby, R., Taylor, L., Treacy, D., Weinstein, A., Wintersgill, M. « The Andes Physics Tutoring System: Lessons Learned ». *International Journal of Artificial Intelligence in Education*, vol. 15, pp. 147-204, 2005.
- [Villiot-Leclercq 2005] Villiot-Leclercq, E. « Capitaliser, diffuser, réutiliser l'expertise pédagogique pour la conception de scénarios pédagogiques : des outils et des méthodes pour enrichir les pratiques dans un

- contexte d'enseignement à distance ». *SIF 2005 "Les institutions éducatives face au numérique"*, Paris, France, décembre 2005.
- [Vincent et al. 2005] Vincent, C., Delozanne, E., Grugeon, B., Gélis, J.-M., Rogalski, J., Coulange, L. « Des erreurs aux stéréotypes : Des modèles cognitifs de différents niveaux dans le projet Pépité ». *Environnements Informatiques pour l'Apprentissage Humain (EIAH'2005)*, Montpellier, France, pp. 297-308, 25-27 mai 2005.
- [Vivet 1984] Vivet, M. « Expertise mathématique et informatique : CAMELIA, un logiciel pour raisonner et calculer ». Thèse d'Etat, Université Paris 6, 1984.
- [Vu Minh et al. 2006] Vu Minh, C., Luengo, V., Vadcard, L. « A Bayesian Network Based Approach for Student Diagnosis in Complex and Ill-structured Domains ». *Technologies de l'Information et de la Connaissance dans l'Enseignement Supérieur et l'Industrie (TICE'2006)*, Toulouse, France, 25-27 octobre 2006.
- [Weber 1996] Weber, G. « Individual selection of examples in an intelligent learning environment ». *Journal of Artificial Intelligence in Education*, vol. 7 n°1, pp. 3-31, 1996.
- [Wenger 1987] Wenger, E. « Artificial Intelligence and Tutoring Systems ». Los Altos, CA, Morgan Kaufmann, 1987.
- [Winkels 1992] Winkels, R. « Explorations in Intelligent Tutoring and Help ». IOS Press, Amsterdam, 1992.
- [Zapata-Rivera et al. 2004] Zapata-Rivera, J.-D., Greer, J. « Interacting with Inspectable Bayesian Student Models », *International Journal of Artificial Intelligence in Education*, vol. 14, pp.1-37, 2004.

RÉFÉRENCES NETOGRAPHIQUES

- [ActiveMath 2007] « ActiveMath ».
<http://www.activemath.org/>
(dernière visite : septembre 2009).
- [AdvancedGeometryITS 2009] « Advanced Geometry ITS ».
<http://www.cs.cmu.edu/~mazda/AdvGeo/>
(dernière visite : septembre 2009).
- [Alfanet 2005] « Alfanet ».
<http://adenu.ia.uned.es/alfanet/>
(dernière visite : septembre 2009).
- [AMBRE 2000] « AMBRE ».
<http://www710.univ-lyon1.fr/~nguoin/ambre.html>
(dernière visite : août 2007).
- [AMBRE-add 2005] N. Guin, S. Jean-Daubias, P. Daubias, F. Fallet-Kahn, L. Boulanger.
« Logiciel d'enseignement de méthodes pour les problèmes additifs s'appuyant sur le cycle du RàPC 2005 ».
<http://liris.cnrs.fr/publis/?id=4470>
(dernière visite : septembre 2009).
- [AMBRE-enseignant 2004] S. Riot, S. Jean-Daubias, N. Guin.
« Logiciel de configuration et de génération d'exercices de AMBRE-add ».
<http://liris.cnrs.fr/publis/?id=4469>
(dernière visite : septembre 2009).
- [Andes 2009] « Andes Physics Tutor ».
<http://www.andestutor.org/>
(dernière visite : septembre 2009).
- [Aplusix 2009] « Aplusix ».
<http://aplustix.imag.fr/>
(dernière visite : septembre 2009).
- [ARIADNE 2006] ARIADNE. « A European Association open to the World, for Knowledge Sharing and Reuse, E-Learning for all, International Cooperation in Teaching, Serving the Learning Citizen ».
<http://www.ariadne-eu.org/>
(dernière visite : septembre 2009).
- [B.O. 2005] « B.O. hors série n° 5 du 25 août 2005 ».
<http://www.education.gouv.fr/bo/2005/hs5/default.htm>
(dernière visite : juillet 2009).
- [Birbandt 2009] Birbandt, P. « Encyclopédagogie - Ecole primaire ».
<http://p.birbandt.free.fr/index.htm>
(dernière visite : septembre 2009).

- [C.N.D.P. 2009] C.N.D.P. « Centre National de Documentation Pédagogique ».
<http://www.cndp.fr/accueil.htm>
(dernière visite : septembre 2009).
- [Camier 2003] Camier, R. « Tables multimédia ».
<http://pagesperso-orange.fr/r.camier/>
(dernière visite : septembre 2009).
- [Camus 2007] Camus, L. « AnglaisFacile.com ».
<http://www.anglaisfacile.com/>
(dernière visite : août 2009).
- [Chevé 2005] Chevé, P. « Abalect ».
<http://pagesperso-orange.fr/philippe.cheve/abalect.htm>
(dernière visite : septembre 2009).
- [ConnaisTuLAnatomie 2002] « Connais-tu l'anatomie ? »
<http://users.skynet.be/bd/connaistulanatomie/>
(dernière visite : septembre 2009).
- [Creexo 2007] « Creexo ».
<http://perso.orange.fr/creexo/>
(dernière visite : septembre 2009).
- [Dromer 2009] Dromer, E. « GECH (Géographie Education Civique Histoire) ».
<http://pagesperso-orange.fr/gech/>
(dernière visite : septembre 2009).
- [DublinCore 1995] DublinCore. « Dublin Core Metadata Initiative ».
<http://dublincore.org/documents/dcmi-terms/>
(dernière visite : septembre 2009).
- [DublinCoreFr 2000] DublinCoreFr. « DublinCore Français ».
<http://www-rocq.inria.fr/~vercoust/METADATA/DC-fr.1.1.html>
(dernière visite : septembre 2009).
- [English] « Planet English ».
<http://www.anglaisfacile.com/free/lejeu/index.php>
(dernière visite : septembre 2009).
- [Enseignons.be 2004] « Enseignons.be »
<http://www.enseignons.be/>
(dernière visite : septembre 2009).
- [ExerciseGeneratorPlus] « Exercise Generator Plus ».
<http://www.clarity.com.hk/program/exercisegenerator.htm>
(dernière visite : août 2007).
- [Exomatiks] « Exomatiks. »
<http://exomatiks.free.fr/mb/>
(dernière visite : avril 2009).

- [GenEval 1997] « GenEval ».
<http://www.ac-grenoble.fr/champo/site/prepa/bcpst/Ariadne/ariadne.htm>
(dernière visite : septembre 2009).
- [Geonote] « Geonote ».
<http://praxis.inrp.fr/praxis/projets/geomatique/geonote/>
(dernière visite : septembre 2009).
- [gStudy] « gStudy ».
<http://www.learningkit.sfu.ca/site/pages/software/gstudy/index.html>
(dernière visite : septembre 2009).
- [iClass] « iClass ».
<http://www.iclass.info/iclass01.asp>
(dernière visite : juillet 2009).
- [IMS-LD] IMS-LD. « IMS Learning Design. Information Model, Best Practice and Implementation Guide, Binding document, Schemas ».
<http://www.imsglobal.org/learningdesign/index.html>
(dernière visite : août 2007).
- [IMS-LIP 2001] « Spécification finale v1.0 d'IMS-LIP ».
<http://www.imsglobal.org/profiles/lipbest01.html>
(dernière visite : mars 2001).
- [IMS_RDCEO 2001] « Spécification finale v1.0 d'IMS RDCEO ».
<http://www.imsglobal.org/competencies/index.html>
(dernière visite : octobre 2002).
- [InterModeller] « InterModeller ».
<http://www.parlog.com/en/intermodeller.html>
(dernière visite : août 2007).
- [ISLE-EIAH 2005] ISLE-EIAH. « Cluster de Recherche ISLE (Informatique, Signal, Logiciels Embarqué) de la région Rhône-Alpes - Projet Personnalisation des EIAH (Environnements Informatiques pour l'Apprentissage Humain) »
<http://cluster-isle-eiah.liris.cnrs.fr/index.php>
(dernière visite : septembre 2009).
- [IUFM_Paris 2005] IUFM Paris. « Formation IUFM Paris ».
http://formation.paris.iufm.fr/archiv_05/corbion/Sites/maths/math.htm
(dernière visite : août 2009).
- [Jean-Daubias 2007] Jean-Daubias, S. « Tri ».
<http://liris.cnrs.fr/stephanie.jean-daubias/projets/logiciel-TriSelectif.html>
(dernière visite : septembre 2009).

- [Jean-Daubias 2009] Jean-Daubias, S. « Tables au trésor ».
<http://liris.cnrs.fr/stephanie.jean-daubias/projets/logiciel-TablesAuTresor.html>
(dernière visite : septembre 2009).
- [Jobin 2007] Jobin, G. G.
<http://www.gilles-jobin.org/>
(dernière visite : août 2009).
- [Lameyse 2009] Lameyse, B. « Déclic en classe ».
<http://lameyse.free.fr/nouvellepape2.htm>
(dernière visite : septembre 2009).
- [LiliMath 2000] « LiliMath ».
<http://lilimath.free.fr/>
(dernière visite : septembre 2009).
- [Lingot 1994] Lingot. « Projet LINGOT ».
<http://pepите.univ-lemans.fr/>
(dernière visite : septembre 2009).
- [LISTEN 1993] « Project LISTEN ».
<http://www.cs.cmu.edu/~listen/index.html>
(dernière visite : septembre 2009).
- [LOM 2002] LOM. « LOM v1.0. Final Draft Standard for Learning Object Metadata. IEEE Standards Department. IEEE P1484.12.1-2002 ».
<http://ltsc.ieee.org/wg12/>
(dernière visite : novembre 2008).
- [Madoré 2007] Madoré, J.-L.
<http://www.famille-madore.fr/exercicescm.html>
(dernière visite : août 2009).
- [Maille 2005] Maille, V. « Point par Point ».
<http://www.toocharger.com/fiches/logiciels/point-par-point/3665.htm>
(dernière visite : septembre 2009).
- [Mathenpoche 2002] « Mathenpoche ».
<http://mathenpoche.sesamath.net/>
(dernière visite : août 2007).
- [MiniLehrer 2007] « Mini Lehrer ».
<http://mini-lehrer.softonic.fr/>
(dernière visite : septembre 2009).
- [Momes.Net 2009] « Momes.Net ».
<http://www.momes.net/>
(dernière visite : septembre 2009).
- [Orthogramme 1999] « Orthogramme ».
<http://pages.infinit.net/logitron/log30.htm>
(dernière visite : septembre 2009).

- [Ortholud 2004] « Ortholud ». <http://www.ortholud.com/index.html>
(dernière visite : août 2009).
- [PAPI 2002] « Version de travail de la spécification PAPI ». <http://edutool.com/papi/>
(dernière visite : février 2002).
- [Pépité] « Pépité ». <http://pepите.univ-lemans.fr/>
(dernière visite : août 2007).
- [PERLEA 2002] « Projet PERLEA ». <http://liris.cnrs.fr/stephanie.jean-daubias/projets/p-perlea.html>
(dernière visite: septembre 2009).
- [SCORM 2001] SCORM. « Advanced Distributed Learning. SCORM Metadata set ». <http://www.adlnet.gov/scorm/index.aspx>
(dernière visite : novembre 2008).
- [Sephonics 2008] « Sephonics ». <http://www.wartoft.nu/software/sephonics/>
(dernière visite : septembre 2009).
- [Sesamath 2002] Sesamath. « MathADoc ». <http://mathadoc.sesamath.net/>
(dernière visite : août 2009).
- [SimQuest 1996] « SimQuest ». <http://www.simquest.nl/learn.htm>
(dernière visite : septembre 2009).
- [SQL-Tutor] « SQL-Tutor ». <http://www.cosc.canterbury.ac.nz/tanja.mitrovic/sql-tut.html>
(dernière visite : août 2007).
- [Teleos 2003] « Teleos ». <http://www-clips.imag.fr/arcade/projets/teleos/TELEOS.html>
(dernière visite : septembre 2009).
- [Triloconte 2007] « Triloconte 1.0.1 ». <http://triloconte.softonic.fr/>
(dernière visite : septembre 2009).
- [UniversDesExperts 2003] « L'Univers des Experts ». <http://experts-univers.com/>
(dernière visite : septembre 2009).
- [VocabOne 2007] « VocabOne 02 ». <http://www.zeniko.ch/static/vocabone.html>
(dernière visite : septembre 2009).

[WayangOutpost]

« Wayang Outpost ».
<http://k12.usc.edu/WO/>
(dernière visite : août 2007).

[WordPress 2009]

« Exomatiks, logiciel éducatif de mathématiques ».
<http://exomatiks.free.fr/mb/>
(dernière visite : septembre 2009).

ANNEXES

ANNEXE A. CORPUS DE LOGICIELS PÉDAGOGIQUES

Cette annexe contient la description générale des trente logiciels pédagogiques étudiés pour proposer l'approche GEPPEToS.

	Système	Provenance	Catégorie	Discipline	Niveau
1	AMBRE-ADD [AMBRE 2000, Nogry et al. 2004]	Laboratoire de recherche	Tuteur intelligent	Mathématiques – Problème additif	CE1-CE2
2	APLUSIX [Nicaud 1993, Aplusix 2009]	Laboratoire de recherche puis commercialisation	Micromonde	Mathématiques – Algèbre	Collège et lycée
3	ROBOTTEACH Leroux 1998, Leroux 2006]	Laboratoire de recherche puis commercialisation	Micromonde	Robotique pédagogique	Collège et formation d'adultes
4	TÉLÉOS [Teleos 2003]	Laboratoire de recherche	Simulateur	Chirurgie orthopédique	Adulte
5	TPÉLEC	Laboratoire de recherche	Micromonde	Electricité	
6	MIRTO [Antoniadis et al. 2004]	Laboratoire de recherche		Langue	
7	PÉPITE [Pépité, Jean 2000]	Laboratoire de recherche	Système de tests	Mathématiques – Algèbre	Fin de 3 ^{ème} - début de 2 ^{nde}
8	GÉONOTE [Geonote]	Laboratoire de recherche	Micromonde		
9	MATHENPOCHE [Mathenpoche 2002]	Pratiques d'enseignants	Exerciseur	Mathématiques	Collège
10	LILIMATH [LiliMath 2000]	Pratiques d'enseignants	Exerciseur	Mathématiques	Collège et lycée
11	MINI LEHRER [MiniLehrer 2007]	Logiciel gratuit	Exerciseur	Allemand	Débutant
12	EXOMATICKS [WordPress 2009]	Commerce	Exerciseur	Mathématiques	Collège et lycée

13	TRILICONTE	Logiciel gratuit	Exerciseur	Français	Enfants 8 - 12 ans
14	ORTHOGRAMME [Orthogramme 1999]	Commerce	Exerciseur	Orthographe	
15	TABLES MULTIMEDIA [Camier 2003]	Logiciel gratuit	Exerciseur	Calcul mental	
16	EVALUTEL	Commerce		Algèbre - géométrie	De la 6 ^{ème} à terminale
17	FRACTIONS	Logiciel gratuit	Exerciseur	Mathématiques - Fractions	
18	TRI [Jean-Daubias 2007]	Pratiques d'enseignant	Tuteur intelligent	Ecologie	Maternelle
19	ACTIVE MATH [Melis et al. 2001, ActiveMath 2007]	Laboratoire de recherche	Exerciseur	Mathématiques	Lycée et université
20	ADVANCED GEOMETRY TUTOR [AdvancedGeometryITS 2009]	Laboratoire de recherche	Tuteur intelligent	Géométrie	
21	ALFANET [Alfanet 2005]	Laboratoire de recherche	Boite à outils	Divers	
22	ANDES [VanLehn et al. 2005, Andes 2009]	Laboratoire de recherche	Tuteur intelligent	Physique	
23	EPSILON	Laboratoire de recherche	Environnement collaboratif	Divers	
24	GSTUDY [gStudy]	Laboratoire de recherche	Boite à outils	Divers	
25	HABIPRO	Laboratoire de recherche	Environnement collaboratif	Divers	
26	INTERMODELLER [InterModeller]	Laboratoire de recherche	Boite à outils	Construction de modèle de classification	
27	M-ECOLAB	Laboratoire de recherche		Ecologie	Enfants de 10 - 11 ans
28	SQL-TUTOR [SQL-Tutor, Mitrovic 1998]	Laboratoire de recherche	Tuteur intelligent	Programmation informatique	
29	WAYANG OUTPOST [WayangOutpost]	Laboratoire de recherche	Micromonde	Mathématiques	Lycée
30	CREEK-TUTOR [Aamodt 2005]	Laboratoire de recherche	Tuteur intelligent	Mathématiques, programmation, diagnostic médical	

ANNEXE B. EXEMPLE DE PROFIL EXPRIMÉ DANS LE LANGAGE PMDL

Cette annexe contient un exemple de profil d'apprenant décrit avec la langage PMDL ainsi que sa représentation graphique. Cet exemple est issu de [Eyssautier-Bavay 2008].

```
profil « Évaluation Nationale CE2 » p1_Adeline 19/11/07
commentaire « Ce profil est basé sur les évaluations nationales CE2, repassées au mois de novembre pour bilan »

Informations_eleve
Inf1 « Nom » chaîne de caractères
Inf2 « Prénom » chaîne de caractères
Inf3 « Date de naissance » date
Inf4 « Redoublant » booléen
Inf5 « Commune d'habitation » (Bron, Caluire, Lyon, Villeurbanne)

element « Algèbre » E1
commentaire « issu du livret du maître, évaluations nationales 2003 »
liste_composantes
ponderrees
echelles
« notes de 0 à 10 » EchE1_1 (0..10) 1 etiquette « /10 »
« pourcentage » EchE1_2 (0..1) 0.01 etiquette « % sujet couvert »
1, E1_C1 « Maîtrise de l'addition », 1
    2, feuille, E1_C2 « Maîtrise de l'addition des nombres entiers », 2
    2, feuille, E1_C3 « Maîtrise de l'addition des nombres décimaux », 1, commentaire « limite du programme de CE2 »
1, feuille, E1_C4 « Maîtrise de la soustraction », 1

element « Géométrie » E2
liste_composantes
non_ponderrees
echelles
« notes de 0 à 10 » EchE1_1 (0..10) 1 etiquette « /10 »
1, E2_C1 « Connaissance du rectangle »
    2, feuille, E2_C2 « Connaissance des propriétés de parallélisme »
    2, feuille, E2_C3 « Connaissance des propriétés d'angle droit »
1, E2_C4 « Connaissance du carré »
    2, feuille, E2_C5 « Connaissance des propriétés de parallélisme »
    2, feuille, E2_C6 « Connaissance des propriétés d'angle droit »
    2, feuille, E2_C7 « Connaissance de l'équivalence des côtés »

element « Résolution de divisions » E3
liste_repartition
nbre_valeurs 2
1 etiquette « valeur de répartition »
2 etiquette « nombre de réussites »
1, feuille, E3_C1 « Méthode euclidienne »
1, feuille, E3_C2 « Méthode des soustractions successives »
1, feuille, E3_C3 « Méthode graphique par paquets »
1, feuille, E3_C4 « Méthode éronée »
    2, feuille, E3_C5 « Erreur d'étourderie »
    2, feuille, E3_C6 « Erreur de calcul »
    2, feuille, E3_C7 « Erreur de raisonnement »

element « Mathématiques » E4
graphe
echelles_composantes
« pourcentage » EchE1_2 (0..1) 0.01 etiquette « % réussite »
E4_C1 « Cadre algébrique »
E4_C2 « Cadre numérique »
E4_C3 « Cadre textuel »
echelles_liens
« Textuelle 3 niveaux » EchE4_1, vrai (Assez Bien; Bien; Très bien) etiquette « rédaction »
E4_L1 (E4_C1; E4_C2)
E4_L2 (E4_C1; E4_C3)
E4_L3 (E4_C3; E4_C2)

element « Commentaire global » E5
texte
```

Partie structure de profil

Informations_eleve_p

Inf1 : « Dupont »

Inf2 : « Paul »

Inf3 : 13/09/97

Inf4 : faux

Inf5 : Lyon

element_p E1

commentaire « Il y a encore besoin de fournir du travail en Algèbre, ne pas relâcher les efforts ! »

liste_composantes_p

E1_C2 (3; 0.6)

E1_C3 (10; 1)

E1_C4 (7; 0.8) commentaire « Des progrès, bravo »

element_p E2

liste_composantes_p

E2_C2 (6)

E2_C3 (10)

E2_C5 (4.5)

E2_C6 (2)

E2_C7 (6.25)

element_p E3

liste_repartition_p

nbre_questions 20

E3_C1 (6; 4)

E3_C2 (4; 4)

E3_C3 (5; 3)

E3_C5 (3; 0)

E3_C6 (2; 0)

E3_C7 (0; 0)

element_p E4

graphe_p

composantes

E4_C1 (0.8)

E4_C2 (0.65)

E4_C3 (0.9)

liens

E4_L1 (« Bien »)

E4_L2 (« Assez Bien »)

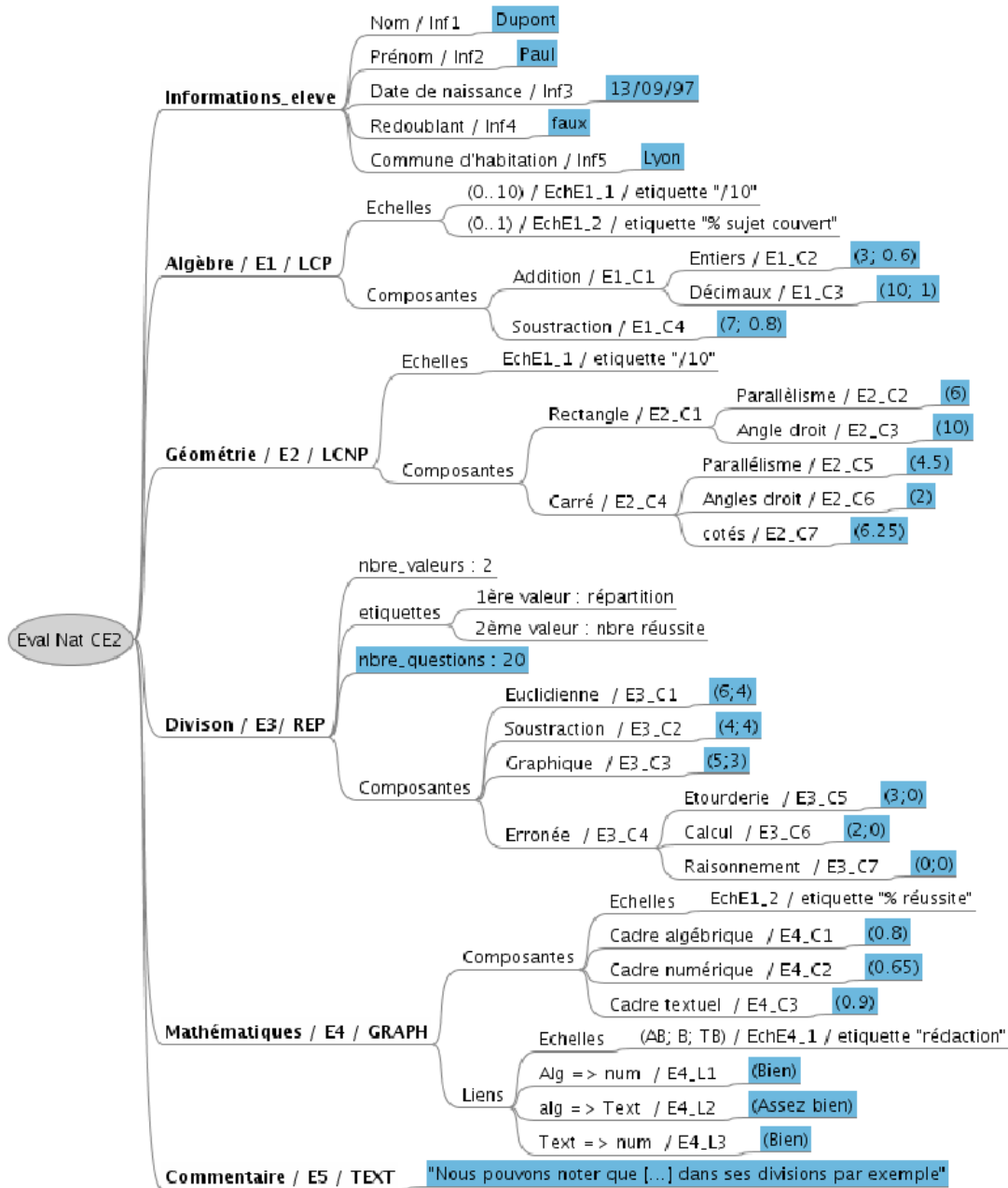
E4_L3 (« Bien »)

element_p E5

texte_p

« Nous pouvons noter que le profil de Paul concernant l'évaluation nationale en CE2 est relativement bon. Toutefois, il pourrait être encore meilleur s'il faisait davantage attention à ses calculs. Paul a le potentiel pour progresser encore, surtout dans la méthodologie de calcul. Il a en effet une forte proportion d'erreurs dans ses divisions par exemple. »

Partie données du profil



ANNEXE C. FORMULES DE CONVERSION D'ÉCHELLES

Dans cette annexe, nous donnons les formules permettant de convertir une valeur donnée dans une échelle textuelle ou numérique vers une autre échelle de type numérique.

PASSAGE D'UNE ECHELLE NUMERIQUE VERS UNE AUTRE ECHELLE NUMERIQUE

Soit (BI1, BS1, pas1) l'échelle initiale et (BI2, BS2, pas2) l'échelle dans laquelle on souhaite obtenir la valeur. BI représente les bornes inférieures, BS les bornes supérieures.

Soit Y la valeur donnée dans l'échelle initiale et Y' la valeur que l'on souhaite obtenir.

Y' est calculée avec la formule :

$$Y' = (Y - (BI1 - BI2)) * ((BS2 - BS1) / (BS1 - BI1))$$

Equation A-1 : Formule permettant convertir une valeur donnée dans une échelle numérique vers une autre échelle numérique.

PASSAGE D'UNE ECHELLE TEXTUELLE VERS UNE AUTRE ECHELLE NUMERIQUE

Soit (X1..Xn) l'échelle initiale et (BI1, BS1, pas1) l'échelle dans laquelle on souhaite la valeur où BI représente les bornes inférieures, BS les bornes supérieures.

Soit Y la valeur donnée dans l'échelle initiale et Y' la valeur que l'on souhaite obtenir.

Y' est calculée avec la formule :

$$Y' = (n - \text{indice de Y dans la liste}) * 1 / (n - 1) * 100$$

Equation A-2 : Formule permettant convertir une valeur donnée dans une échelle textuelle vers une échelle numérique

ANNEXE D. EXEMPLES D'ACTIVITÉS PAPIER

Cette annexe contient des exemples d'activités papier pouvant appartenir à la typologie que nous avons présentée dans la section 6.3.2 page 120. Ces exemples sont regroupés par type d'activités.

ACTIVITÉS PAPIER DE TYPE « IDENTIFICATION DES PARTIES DU TEXTE » (CF. A1 SUR LA FIGURE 6-2)

Exemple A-1 : Exercice à trous avec liste des mots à placer [Claudal 2007].

Des mots de ce texte ont disparu, saurez-vous les remettre en place ?

Quand vous serez bien _____, au soir à la _____,
Assise auprès du feu, dévidant et filant,
Direz chantant mes vers, en vous émerveillant :
« _____ me célébrait du temps que j'étais belle. »
Lors vous n'aurez servante oyant telle _____,
Déjà sous le labeur à demi sommeillant,
Qui au bruit de Ronsard ne s'aïlle réveillant,
Bénissant votre nom de _____ immortelle.
Je serai sous la terre, et _____ sans os
Par les ombres myrteux je prendrai mon repos ;
Vous serez au _____ une vieille accroupie,
Regrettant mon _____ et votre fier dédain.
Vivez, si m'en croyez, n'attendez à demain :
Cueillez dès aujourd'hui les _____ de la _____.

Mots à replacer : Fantôme, vie, vieille, amour, louange, roses, foyer, Ronsard, nouvelle, chandelle.

Exemple A-2 : Exercice à trous sans liste des mots à placer, issu du logiciel Adapte.

Remplir les trous.

Maître _____, sur un arbre perché,
Tenait en son _____ un _____.
Maître _____, par l'odeur alléché,
Lui tint à peu près ce langage :
« Hé ! bonjour, Monsieur du Corbeau.
Que vous êtes joli ! que vous me semblez beau !
Sans mentir, si votre _____
Se rapporte à votre _____,
Vous êtes le Phénix des hôtes de ces bois. »
À ces mots le Corbeau ne se sent pas de joie :
Et pour montrer sa belle voix,
Il ouvre un large bec, laisse tomber sa proie.
Le Renard s'en saisit, et dit : « Mon bon Monsieur,
Apprenez que tout _____
Vit aux dépens de celui qui l'écoute.
Cette leçon vaut bien un fromage, sans doute. »

Le Corbeau honteux et confus,
Jura, mais un peu tard, qu'on ne l'y prendrait plus.

Il faut, autant qu'on peut, obliger tout le monde :
On a souvent besoin d'un plus petit que _____.
De cette vérité deux Fables feront _____,
Tant la chose en preuves abonde.
Entre les pattes d'un _____,
Un _____ sortit de terre assez à l'étourdie :
Le _____ des animaux, en cette occasion,
Montra ce qu'il était, et lui donna la vie.
Ce bienfait ne fut pas perdu.
Quelqu'un aurait-il jamais cru
Qu'un Lion d'un Rat eût affaire ?
Cependant il advint qu'au sortir des forêts
Le Lion fut pris dans des rets,
Dont ses rugissements ne le purent défaire.
Sire Rat accourut, et fit tant par ses dents
Qu'une maille rongée emporta tout l'ouvrage.
Patience et longueur de temps
Font plus que _____ ni que _____.

Exemple A-3 : Exercice à trous avec pour chaque trou un indice sur le mot à placer [Camus 2007].

Conjuguer au prétérit :

1. I (go) _____ to the swimming pool.
2. My uncle (build) _____ a house.
3. I (walk) _____ through the garden.
4. Did you (have) _____ a nice week end?
5. We (stay) _____ at home.
6. Yes, I (do) _____.
7. My TV (be) _____ broken.
8. I (buy) _____ another one.
9. Did you (go) _____ to the museum?
10. We (play) _____ Monopoly.

Exemple A-4 : Exercice à trous sans liste de mots à placer, issu des pratiques d'enseignants.

Compléter le texte ci-dessous :

La respiration est permise par le _____, un muscle situé sous les poumons qui s'abaisse lorsque les _____ se remplissent d'air et remonte lorsque ceux-ci se vident. Un cycle respiratoire est composé d'une et d'une expiration. En _____, on apporte l'air dans nos poumons. L'air passe par la gorge, le larynx puis la trachée et il arrive dans les bronches, qui composent les poumons. Ces bronches se ramifient, un peu comme les branches d'un arbre. Elles se terminent par les _____, des bronches de plus petit calibre. C'est à ce niveau que se trouvent les _____, le lieu des échanges gazeux.

Exemple A-5 : Exercice d'identification de noms dans un texte [Madoré 2007].

Souligne les noms contenus dans les phrases suivantes.

- Quand tu auras ton vêtement neuf, je vérifierai la solidité des boutons.
- Le pêcheur suit des yeux le petit flotteur rouge.
- J'emporte un roman policier pour lire dans le train.
- Ils se retrouvent à la piscine le mercredi : c'est leur meilleur moment de détente de la semaine.

Et maintenant, compte les mots soulignés. Si tu n'as pas d'erreur, tu dois en trouver 13.

Exemple A-6 : Exercice d'identification des éléments pertinents d'un texte [Madoré 2007].

Barre les renseignements qui ne servent pas pour répondre à la question: "À quelle heure arrivera-t-elle à la maison? "

Maman arrive à 15 h 30 au supermarché. Elle achète pour 98 € de marchandises et 30 litres d'essence. Elle repart à 17 heures et roule pendant 1 heure pour rentrer à la maison.

ACTIVITÉS PAPIER DE TYPE « ANALYSE DU TEXTE » (CF. A2 SUR LA FIGURE 6-2)

Exemple A-7 : Exercice d'analyse grammaticale de texte [Madoré 2007].

Es-tu capable d'indiquer entre parenthèses par N, A ou V si les mots soulignés sont des noms (N), des adjectifs (A) ou des verbes (V).

Les pulls et les chaussettes () sèchent sur le fil à linge.
 Repasse les serviettes dès qu'elles seront () sèches.
 Ce sont des rôtis bien () tendres.
 Mes voisins adoptent deux enfants et les () éduquent de leur mieux.
 Il leur faut beaucoup de () force pour achever ce travail.
 Ma grand-mère prend deux biscottes et les () émiette dans son bol.
 Les spectateurs qui sont moins () proches () s'approchent.
 On cultive de vastes étendues de blé dans les () fermes de la Beauce.
 Ils ouvrent et () ferment la porte pour s'assurer qu'elle ne grince pas.
 Elle ramasse quelques fruits et les () enveloppe dans du papier journal.
 Si tu () forces sur cette branche, elle va casser.
 Les veaux () têtent leur mère.

Exemple A-8 : Exercice d'analyse grammaticale de texte [Madoré 2007].

Souligne les COD⁸ d'un trait et les COI⁹ de deux traits.

Apollo 11 emporte le commandant Neil Armstrong et les pilotes Michael Collins et Edwin Aldrin.
 Armstrong et Aldrin laissent Collins en orbite autour de la Lune.
 Neil Armstrong descend l'échelle du module lunaire Eagle.
 L'astronaute touche le sol lunaire.

⁸ Compléments d'objet direct

⁹ Compléments d'objet indirect

Ses pas laissent des empreintes.
Sa démarche est gauche.
Avec son collègue Edwin Aldrin, il recueille 21 kg de roches lunaires.
L'astronaute parlera de son expédition.
La joie emplira les cœurs.
Armstrong et Aldrin raconteront leur aventure au monde entier.
Ils penseront longtemps à ce voyage extraordinaire.

ACTIVITÉS PAPIER DE TYPE « SENS DU TEXTE » (CF. A3 SUR LA FIGURE 6-2)

Exemple A-9 : Exercice de traduction, issu du logiciel Adapte.

Donner la traduction du texte de William Shakespeare.

To be or not to be: that is the question:
Whether it's nobler in the mind to suffer
The slings and arrows of outrageous fortune
Or to take arms against a sea of troubles,
And by opposing end them.

Exemple A-10 : Exercice de traduction, issu du logiciel Adapte.

Donner la traduction du texte de William Shakespeare.

To die, to sleep;
To sleep: perchance to dream: ay, there's the rub;
For in that sleep of death what dreams may come
When we have shuffled off this mortal coil,
Must give us pause.
Thus conscience does make cowards of us all;
And thus the native hue of resolution
Is sicklied o'er with the pale cast of thought,
And enterprises of great pith and moment
With this regard their currents turn awry,
And lose the name of action.

Exemple A-11 : Exercice de définition, issu du logiciel Adapte.

Donner la définition des termes suivants : cytoplasme, noyau, lysosome, mitochondrie.

Exemple A-12 : Exercice de définition, issu du logiciel Adapte.

Donner la définition des termes suivants : membrane cellulaire, réticulum endoplasmique lisse / rugueux, appareil de Golgi, centrosome.

ACTIVITÉS PAPIER DE TYPE « TRANSPOSITION DU TEXTE » (CF. A4 SUR LA FIGURE 6-2)

Exemple A-13 : Exercice de correction orthographique [Madoré 2007].

Observe les terminaisons des noms. Rectifie-les si tu le juges nécessaire.

un collier de perle	un marchand de légume
des soulier couverts de boue	des rue envahies de piéton
des regard d'amitié	un texte rempli d'erreur
une pipe bourrée de tabac	une maison à volet verts
des camion de sable	des cri de joie

Exemple A-14 : Exercice de transposition au pluriel [Madoré 2007].

Es-tu capable d'ajouter la marque du pluriel si le sens l'exige ?

un mouvement d'impatience	un trousseau de clé
une famille de marmotte	un vent de sable
une pluie de feuille	une grêle de dragée
une ligne d'arbre	une file de voiture
un pneu de voiture	une enveloppe couverte de timbre
un sachet de poivre	un jardinier chaussé de sabot
un ciel chargé de nuage	un fracas de vaisselle
une caisse de livre	des animaux en liberté
une semaine de vacance	un sac de pomme de terre

Exemple A-15 : Exercice de changement de temps [UniversDesExperts 2003].

Recopie ce texte à l'imparfait de l'indicatif.

Une fillette se débat dans un fleuve ou nagent des crocodiles. Une agitation extraordinaire règne au grand village de Mossaka. Des femmes et des enfants sortent des cases, courent et pataugent dans la boue qui leur monte jusqu'aux genoux.

Exemple A-16 : Exercice de changement de temps [Enseignons.be 2004].

Recopie ce texte à l'imparfait de l'indicatif.

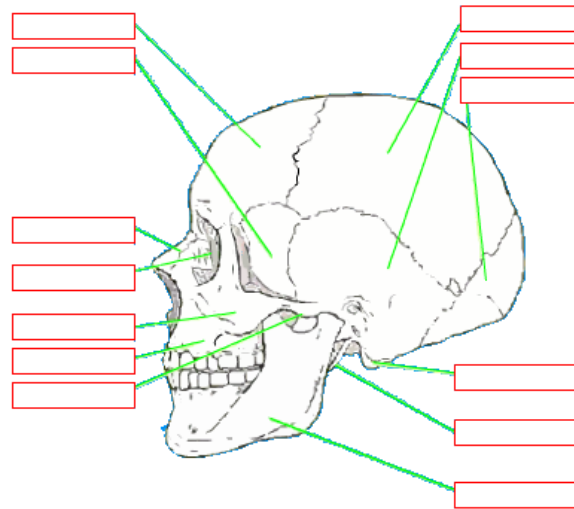
Quand le gendarme monte la garde à l'entrée, les fillettes sont heureuses. L'agent leur rend service, les aide au passage clouté, leur apprend à traverser. C'est agréable de compter sur un ami qui nous protège du danger.

ACTIVITÉS PAPIER DE TYPE « ANNOTATION D'ILLUSTRATION » (CF. B1 SUR LA FIGURE 6-2)

Exemple A-17 : Exercice d'annotation d'illustration [ConnaisTuLAnatomie 2002].

Connais-tu l'anatomie ? Place les noms d'os à leur place sur la figure :

Apophyse mastoïde, Apophyse styloïde, Apophyse zygomatique, Frontal, Lacrymal, Mandibule, Maxillaire, Nasal, Occipital, Pariétal, Temporal, Zygomatique.



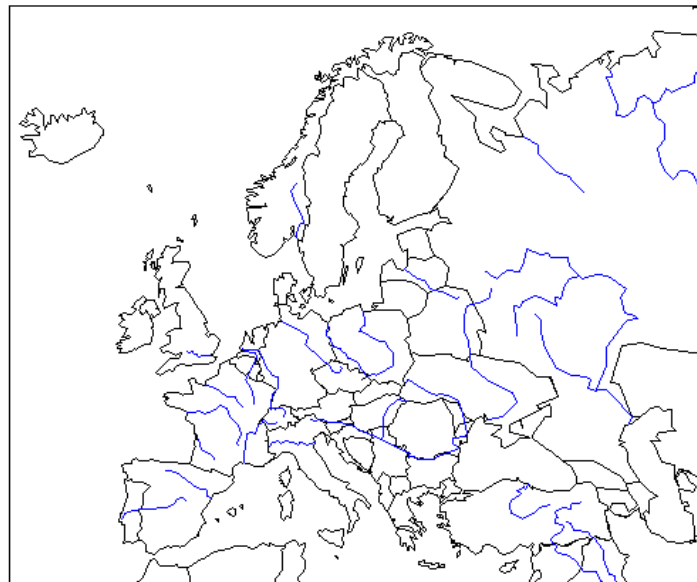
Exemple A-18 : Exercice d'annotation d'illustration [Dromer 2009].

Écrivez en les plaçant sur cette carte de l'Europe :

- en noir : la Pologne, la Suède, l'Albanie, l'Autriche, puis la Sardaigne,
- en rouge, de manière approximative : Rome, Copenhague, Berlin, Athènes,
- en bleu : le Rhin.

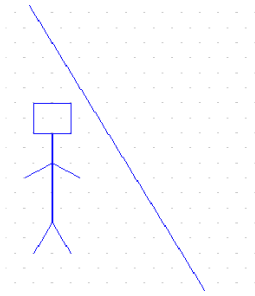
Coloriez :

- en marron : l'Oural,
- en orange : les Alpes.



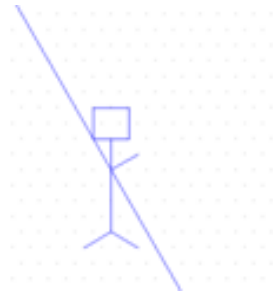
Exemple A-19 : Exercice de symétrie, issu des pratiques d'enseignants.

Construis le symétrique de la figure suivante :



Exemple A-20 : Exercice de symétrie, issu des pratiques d'enseignants.

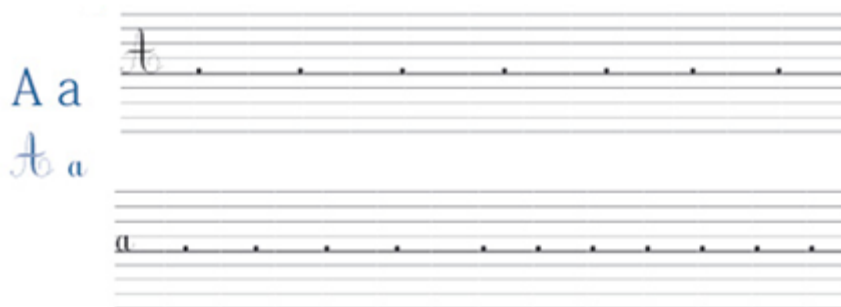
Construis le symétrique de la figure suivante :



ACTIVITÉS PAPIER DE TYPE « COPIE D'UN MODÈLE » (CF. B2 SUR LA FIGURE 6-2)

Exemple A-21 : Exercice d'écriture, issu des pratiques d'enseignants.

Remplir les lignes d'écriture suivante :



Exemple A-22 : Exercice d'écriture, issu des pratiques d'enseignants.

Recopier 10 fois la clef de sol sur la partition suivante :



ACTIVITÉS PAPIER DE TYPE « CLASSEMENT D'OBJETS » (CF. C1 SUR LA FIGURE 6-2)

Exemple A-23 : Exercice de classement de nombres, issu des pratiques d'enseignants.

Ranger les listes de nombres suivants par ordre croissant :

1, -2, 2, 3, -1, -3, 0
7/5, 6, -0.7, 0.2, 100, 18, 0, 3/4.

Exemple A-24 : Exercice de texte en désordre, issu des pratiques d'enseignants.

Remettre la phrase suivante dans l'ordre :

sick He made by eating much chocolate himself too

Exemple A-25 : Exercice de texte en désordre [Camus 2007].

Remettre cette lettre dans l'ordre :

Dear Alice and Jim,
Would you having a look for me ?
The conversation was excellent and the food delicious !
I think I left a pair of brown trousers in the wardbrode of my room.
I had a wonderful time.
Please can you let me know if you find them?
Thank you for having me to stay last weekend.
It was lovely to see you all. See you again soon.
Could you do something for me ?
Thanks a lot.
Love - Jack

Exemple A-26 : Exercice de texte en desordre [Camus 2007].

Remettre cette lettre dans l'ordre :

Dear Reception,
Could you possibly check if this is so ?
The service was superb and the food delicious !
I have lost a pair of brown trousers, which I think I left in the wardrobe of my room.
We had a very pleasant stay.
I look forward to hearing from you.
Many thanks for the weekend break that my wife and I enjoyed at your hotel recently.
We hope to visit your hotel again soon.
I would like to ask you a favour.
I would be most grateful !
Yours sincerely - Jack Higgins.

ACTIVITÉS PAPIER DE TYPE « GROUPEMENT D'OBJETS » (CF. C2 SUR LA FIGURE 6-2)

Exemple A-27 : Exercice d'association de forme, issu des pratiques d'enseignants.

Entourer les triangles orientés vers la droite : ● ▲ ■ ● ◀ ▲ ▶ ▲ ■ ■ ● ▲

Exemple A-28 : Exercice d'association des images à des mots [Ortholud 2004].

Reliez le nom d'un sport et le dessin qui le représente :

Le karaté	
Le Sumo	
Le judo	
Le vtt	
Le lancer du javelot	
Le football	

Exemple A-29 : Exercice de détection de familles de mots [Madoré 2007].

Serais-tu capable de séparer les deux familles de mots ?

frais – franchir – fraîcheur – franchissable – rafraîchir – franchissement –
rafraîchissement – infranchissable – défraîchi

ACTIVITÉS PAPIER DE TYPE « TABLE DE CONJUGAISON » (CF. D1 SUR LA FIGURE 6-2)

Exemple A-30 : Exercice de conjugaison française, issu des pratiques d'enseignants.

Conjuguer les verbes MANGER, BOIRE et DIGERER au présent de l'indicatif.

Exemple A-31 : Exercice de conjugaison française [Madoré 2007].

Écris les verbes aux temps et personnes demandés.

avoir – 1ère pers. sing. - présent de l'indicatif :
être – 2e pers. sing. - imparfait de l'indicatif :
aller - 2e pers. plur. - futur simple :
annoncer - 3e pers. plur. - passé composé de l'indicatif :
choisir - 2e pers. plur. - présent de l'indicatif :

venir - 2e pers. sing. - futur simple :
 boire - 1ère pers. sing. - imparfait de l'indicatif :
 refaire - 2e pers. sing. - imparfait de l'indicatif :
 défaire - 2e pers. plur. - futur simple :
 courir - 2e pers. sing. - futur simple :

Exemple A-32 : Exercice de conjugaison française, issu des pratiques d'enseignants.

Remplis le tableau en conjuguant les verbes au présent de l'indication.

	Manger	Finir	Boire	Etre
Je				
Tu				
Il				
Nous				
Vous				
Elles				

ACTIVITÉS PAPIER DE TYPE « TABLE DE MATHÉMATIQUES » (CF. D2 SUR LA FIGURE 6-2)







Exemple A-33 : Exercice de table de multiplication, issu des pratiques d'enseignants.

Écrire la table de multiplication des nombre 2, 5 et 9.

ACTIVITÉS PAPIER DE TYPE « AUTRE TABLEAU À DOUBLE ENTRÉE » (CF. D3 SUR LA FIGURE 6-2)

Exemple A-34 : Exercice de complétion d'un tableau à double entrée [IUFM_Paris 2005].

Complète le tableau.

Créations	ROUGE	BLEU	VERT	JAUNE
Formes				
				
				
				

Exemple A-35 : Exercice de complétion d'un tableau à double entrée [Jobin 2007].

Calculez les valeurs numériques des expressions algébriques ci-dessous selon la valeur accordée à la variable y .

Expression algébrique	Si $y = 1$	Si $y = -2$
$3y - 2$	<input type="text"/>	<input type="text"/>
$-2y + 1$	<input type="text"/>	<input type="text"/>
$-y + 3$	<input type="text"/>	<input type="text"/>
$5y - 2y$	<input type="text"/>	<input type="text"/>

Exemple A-36 : Exercice de complétion d'un tableau à double entrée [Madoré 2007].

Serais-tu capable d'écrire sur chaque ligne des mots de la même famille ?

Noms	Verbes	Adjectifs	Adverbes
		sec	
actualité			
	généraliser		
		accidentel	
	salir		
ouverture			
			longuement
	faiblir		
		réel	
		courant	
	élargir		

Exemple A-37 : Exercice d'utilisation d'un tableau à double entrée, issu des pratiques d'enseignants.

Trouver les valeurs des paramètres a , b , c de la fonction décrite dans la deuxième colonne du tableau.

x	$ax^2 + bx + c$
-1	6
0	3
2	3

Exemple A-38 Exercice de complétion d'un tableau à double entrée, issu des pratiques d'enseignants.

Complète les 9 cases du tableau avec les chiffres de 1 à 9.

Le produit de chaque colonne doit donner le résultat indiqué, de même pour chaque ligne.

<input type="text"/>	<input type="text"/>	<input type="text"/>	140
<input type="text"/>	<input type="text"/>	<input type="text"/>	16
<input type="text"/>	<input type="text"/>	<input type="text"/>	162
21	90	192	

ACTIVITÉS PAPIER DE TYPE « EXPRESSION ALGÈBRIQUE » (CF. E SUR LA FIGURE 6-2)

Exemple A-39 : Exercice portant sur une expression algébrique, des pratiques d'enseignants.

Calculer $2 + 3$.

Exemple A-40 : Exercice portant sur une expression algébrique, des pratiques d'enseignants.

Développer $(2x + 3)(5x + 4) + 6(2x + 3)$.

Exemple A-41 : Exercice portant sur une expression algébrique, des pratiques d'enseignants.

Résoudre $1,5x + 8 < -4,5x$.

Exemple A-42 : Exercice portant sur une expression algébrique, des pratiques d'enseignants.

Résoudre $\begin{cases} x - 2y = 4 \\ x + 4y = -3 \end{cases}$

ACTIVITÉS PAPIER DE TYPE « PROBLÈME SCIENTIFIQUE » (CF. F SUR LA FIGURE 6-2)

Exemple A-43 : Exercice portant sur un problème additif [Duclosson et al. 2005a].

Tu avais 4 billes, tu en as gagné 10 pendant la récréation. Combien as-tu de billes dans la poche maintenant ?

Exemple A-44 Exercice portant sur un problème de dénombrement [Guin 1997].

On dispose d'un jeu de 32 cartes. On en tire 5 simultanément. Quel est le nombre de tirages contenant exactement 2 valets et 2 cœurs ?

Exemple A-45 Exercice portant sur un problème de dénombrement [Guin 1997].

Un sac contient 9 jetons numérotés de 1 à 9. On tire 3 jetons successivement, en remettant à chaque fois le jeton tiré dans le sac avant de tirer le suivant. On écrit côte à côte chacun des trois chiffres tirés, dans l'ordre du tirage, formant ainsi un nombre de 3 chiffres. Combien peut-on obtenir de résultats différents ?

Exemple A-46 Exercice portant sur un problème de chimie [Guin 1997].

Un réservoir fermé renferme de l'air à 35°C et sous une pression de 7 bars. Que devient la pression quand la température s'abaisse à 10°C ?

ACTIVITÉS PAPIER DE TYPE « DEMONSTRATION » (CF. G SUR LA FIGURE 6-2)

Exemple A-47 : Exercice de démonstration [Sesamath 2002].

Tracer un parallélogramme ABCD. La bissectrice de DAB coupe la droite (DC) en I. La bissectrice de BCD coupe la droite (AB) en J.

1. Démontrer que les droites (AI) et (JC) sont parallèles
2. Démontrer que AJCI est un parallélogramme et que les droites (IJ), (AC) et (BD) sont concourantes.

ACTIVITÉS PAPIER DE TYPE « QUESTIONS - RÉPONSES » (CF. H SUR LA FIGURE 6-2)

Exemple A-48 : QCM, issu des pratiques enseignantes.

Que veut dire le mot anglais CAR en français :

- Chat
- Bus
- Voiture

Exemple A-49 : QCM [Ortholud 2004].

Dans ce quiz, trouver la bonne orthographe :



- Une abeille
- Une abaille
- Une abbeille



- Un hipauptame
- Un hippopotame
- Un hipoppotame



- Un rhinoceros
- Un rhynocéraus
- Un rhinocéros

Exemple A-50 : Extrait d'un QCM [Dromer 2009].

Répondre au questionnaire. Pour chaque question il peut y avoir aucune ou plusieurs réponses possibles.

1. La république de Weimar (novembre 1918 à janvier 1933) est
 - a. une dictature
 - b. une démocratie

2. L'économie allemande est ruinée
 - a. par une inflation galopante en 1923
 - b. à partir de 1930 par la crise économique mondiale
3. Le nombre d'adhérents au parti nazi dirigé par Hitler est de 27000 en 1925, et de
 - a. 390 000 en 1933
 - b. 3 900 000 en 1933
4. Les nazis sont pour
 - a. le traité de Versailles
 - b. le pangermanisme
 - c. le communisme
 - d. l'inégalité entre les hommes
 - e. la supériorité des Aryens
 - f. l'antisémitisme

Exemple A-51 : QROC, issu des pratiques enseignantes.

Répondre aux questions suivantes :

- Combien y a-t-il de nombres premiers qui sont divisibles par 3 ?
- Quelle est la décomposition en produit de facteurs premiers de 18 ?
- $3 \times 5 \times 2$ est la décomposition en produit de facteurs premiers de :
- Quelle est la décomposition en produit de facteurs premiers de 200 ?

ANNEXE E. SÉMANTIQUE DE LA REPRÉSENTATION GRAPHIQUE UTILISÉE POUR PRÉSENTER LES MODÈLES

Dans cette annexe, nous donnons le sens des éléments graphiques utilisés pour présenter les modèles.



Nœud composé de



Choix entre les noeuds



Élément obligatoire



Élément optionnel



Détails affichés



Détails masqués



Élément terminal



Nombre d'occurrences

ANNEXE F. LES PATRONS D'EXERCICES DE L'APPROCHE GEPPETO_P

Dans la section 6.3.3.c, nous avons présenté le patron d'exercices correspondant à la catégorie « Travail sur texte » de notre typologie d'activités papier (cf. Figure 6-2). Dans cette annexe, nous présentons quatre autres patrons correspondant à cette typologie. Les patrons relatifs aux catégories « Expression algébrique », « Problème scientifique classifiable » et « Démonstration » n'étant pas entièrement définis, nous ne les présentons pas. Nous suivons le même principe de présentation : la définition du patron accompagnée d'exemples de structures d'exercices créés par le logiciel Adapte (cf. Chapitre 8) à partir des patrons. La sémantique des graphiques utilisés pour présenter les définitions des patrons est présentée dans l'Annexe E page 305.

PATRON B – TRAVAIL SUR ILLUSTRATION

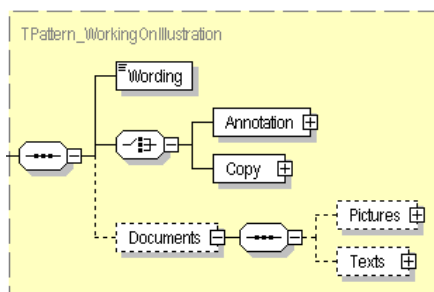


Figure A-1 : Patron d'exercices « Travail sur illustration ».

Le patron « Travail sur illustration » permet de créer des activités papier appartenant à la catégorie B de la typologie présentée sur la Figure 6-2. Ce patron, présenté Figure A-1, précise qu'une activité de travail sur illustration est formée d'une consigne, *Wording*, du contenu d'un des deux patrons opérationnels rattachés au travail sur illustration, *Annotation* ou *Copy*, et éventuellement de documents, *Pictures* ou *Texts*.

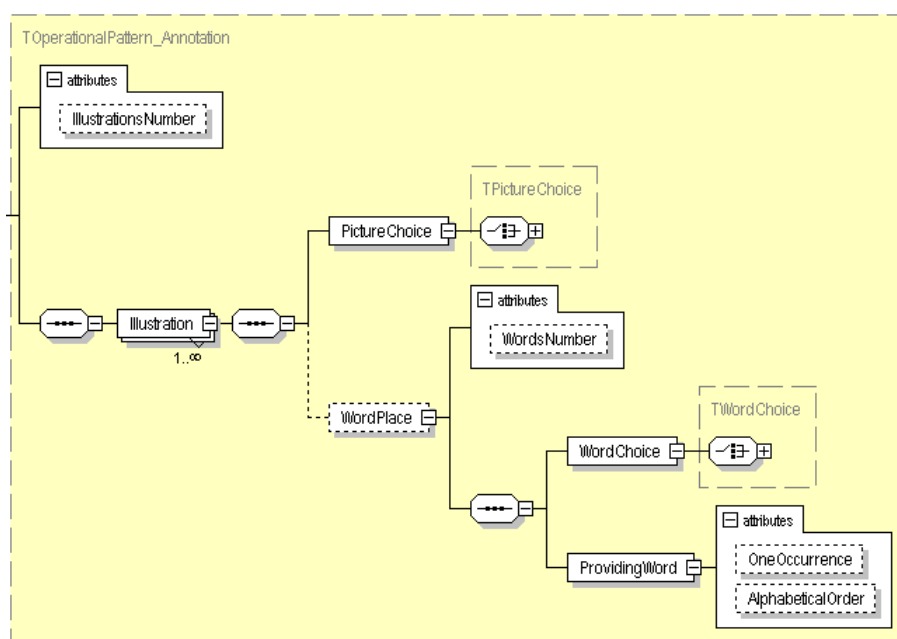


Figure A-2 : Patron opérationnel « Annotation d'illustration ».

Le patron opérationnel « Annotation d'illustration », présenté Figure A-2, décrit une activité comme un ensemble non vide d'éléments *Illustration* associé au nombre *IllustrationsNumber*. Ces éléments s'ajoutent à ceux contenus dans le patron « Travail sur illustration ».

Chaque élément *Illustration* contient les contraintes permettant de créer une activité sur une illustration. Le fait de pouvoir avoir plusieurs éléments *Illustration* permet de créer une structure d'exercices qui permettra de générer des activités portant sur des illustrations différentes. Nous revenons sur ce point dans les exemples. Un élément *Illustration* contient donc des indications permettant de choisir une illustration, *PictureChoice*, et éventuellement des indications permettant de sélectionner les mots à placer sur l'illustration, *WordPlace*.

L'élément *PictureChoice* est défini comme précédemment (cf. Figure 6-8, page 132).

L'élément *WordPlace* spécifie comment choisir les mots à placer par l'apprenant sur l'illustration, *WordChoice*, et comment présenter l'activité, en indiquant si les mots sont fournis à l'apprenant, *ProvidingWord*, et le cas échéant dans quel ordre doit être triée la liste et si celle-ci doit contenir toutes les occurrences d'un même mot.

Ce choix des mots peut se faire de deux manières (cf. Figure A-3). Dans le premier cas, les mots sont déterminés de manière définitive en fournissant la liste des mots à placer, *FileIdentifier*. Dans le second cas, les mots sont choisis lors de la génération des exercices en respectant les contraintes *WordConstraints*. Ces contraintes peuvent porter sur la langue du mot, *Language*, sa nature (nom, adjectif, etc.), *Nature*, la famille de mot auquel il appartient, *Family*, les thèmes auxquels il fait référence, *Subject* et sa longueur, *LettersNumber*. L'élément *Excluded* permet d'interdire le choix de certains mots.

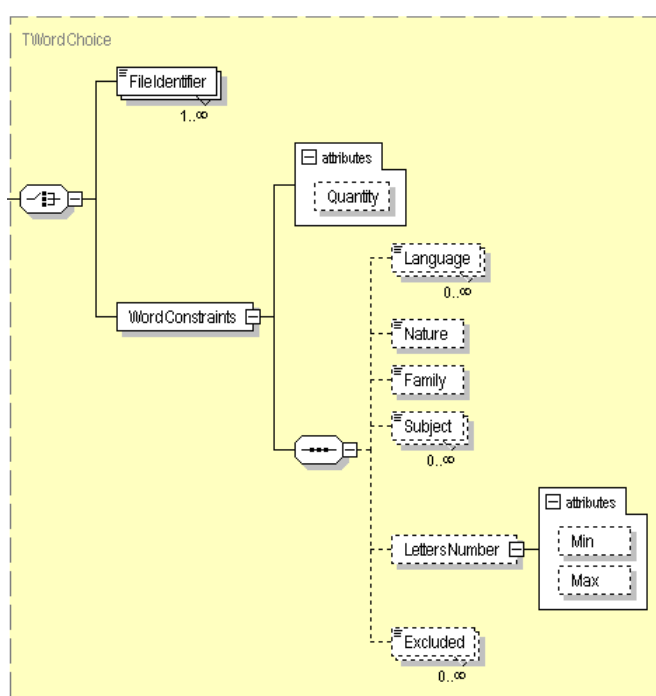


Figure A-3 : Choix d'un mot dans un patron d'exercices.

L'élément *WordsNumber* (cf. Figure A-2) indique le nombre de mots que l'apprenant devra placer sur l'illustration. Ce nombre est toujours supérieur à un, et inférieur ou égal au nombre d'éléments contenus dans *WordChoice*. Une structure d'exercices pourra ainsi contenir dix éléments dans *WordChoice* et un *WordNumber* valant cinq. Dans ce cas, l'apprenant n'aura à placer que cinq mots sur les dix.

L'élément *IllustrationsNumber* détermine le nombre d'illustrations qu'un exercice contiendra. Ce nombre est toujours supérieur à un et il est inférieur ou égal au nombre d'éléments *Illustration* contenus dans la structure d'exercices.

Nous allons à présent voir deux exemples de structures d'exercices créées avec le patron opérationnel « Annotation d'illustration ». La première structure d'exercices, fournie Figure A-4, permet de générer un exercice de biologie. La consigne de l'exercice est fournie dans l'élément *Wording*. L'unique illustration est l'illustration 24 de la base de données, qui correspond à une image d'un crâne humain. Les treize mots à placer sont précisés dans les éléments *FileIdentifier*. L'élément *ProvidingWord* indique qu'il faut donner la liste complète des mots à l'apprenant dans l'ordre alphabétique.

```

<ExercisesStructure>
  <Metadata>...</Metadata>
  - <WorkingOnIllustration>
    <Wording>Connais-tu l'anatomie ? Place les noms d'os à leur place sur la figure :</Wording>
    - <Annotation IllustrationsNumber="1">
      - <Illustration>
        - <PictureChoice>
          <FileIdentifier>24</FileIdentifier>
        </PictureChoice>
        - <WordPlace WordsNumber="13">
          - <WordChoice>
            <FileIdentifier>Nasal</FileIdentifier>
            <FileIdentifier>Maxillaire</FileIdentifier>
            <FileIdentifier>Zygomatique</FileIdentifier>
            <FileIdentifier>Temporal</FileIdentifier>
            <FileIdentifier>Frontal</FileIdentifier>
            <FileIdentifier>Pariétal</FileIdentifier>
            <FileIdentifier>Occipital</FileIdentifier>
            <FileIdentifier>Apophyse zygomatique</FileIdentifier>
            <FileIdentifier>Apophyse mastoïde</FileIdentifier>
            <FileIdentifier>Mandibule</FileIdentifier>
            <FileIdentifier>Lacrymal</FileIdentifier>
            <FileIdentifier>Apophyse styloïde</FileIdentifier>
          </WordChoice>
          <ProvidingWord AlphabeticalOrder="true" OneOccurrence="false">true</ProvidingWord>
        </WordPlace>
      </Illustration>
    </Annotation>
  </WorkingOnIllustration>
</ExercisesStructure>

```

Figure A-4 : Structure d'exercices issue du patron opérationnel « Annotation d'illustration » associé au patron « Travail sur illustration ».

Les contraintes contenues dans cette structure de profils sont si précises qu'elles permettent de ne générer qu'un seul énoncé (cf. Exemple A-17 de l'Annexe D, page 295).

La Figure A-5 fournit un second exemple. Cette structure d'exercices stipule que pour créer un exercice il faut choisir, dans la base de données, une illustration portant sur la symétrie axiale. Les apprenants ne devront placer aucun mot sur cette illustration. Cette structure d'exercices, contrairement à la précédente, permet de créer des exercices très diversifiés au niveau des illustrations proposées (cf. Exemples A-19 et A-20 de l'Annexe D, page 297).

Le patron opérationnel « Copie d'un modèle », présenté Figure A-6, décrit une activité comme un ensemble non vide d'élément *Content* associé au nombre *ContentsNumber*. Ces éléments s'ajoutent à ceux contenus dans le patron « Travail sur illustration ».

```

<ExercisesStructure>
  <Metadata>...</Metadata>
  - <WorkingOnIllustration>
    <Wording>Construit le symétrique de la figure suivante :</Wording>
    - <Annotation IllustrationsNumber="1">
      - <Illustration>
        - <PictureChoice>
          - <PictureConstraints Quantity="1">
            <Keyword>Symetrie axiale</Keyword>
          </PictureConstraints>
        </PictureChoice>
      </Illustration>
    </Annotation>
  </WorkingOnIllustration>
</ExercisesStructure>

```

Figure A-5 : Structure d'exercices issue du patron opérationnel « Annotation d'illustration » associé au patron « Travail sur illustration ».

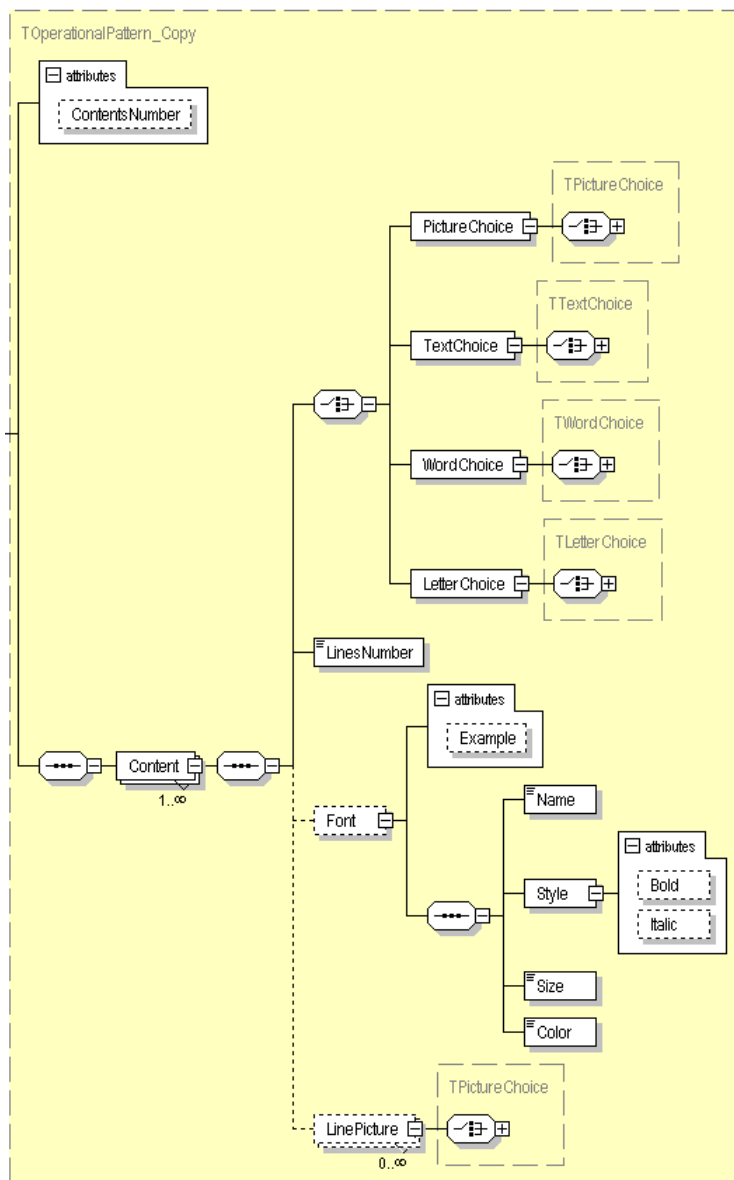


Figure A-6 : Patron opérationnel « Copie d'un modèle ».

Un élément *Content* peut être une image, un texte, un mot ou tout simplement une lettre. Ainsi, chaque élément *Content* contient des indications permettant de choisir un modèle à recopier, *PictureChoice*, *TextPicture*, *WordChoice* ou *LetterChoice*, le nombre de fois qu'il faut recopier le

modèle, *LinesNumber*, et éventuellement une série de figures servant de support à l'apprenant pour recopier le modèle, *LinePicture*.

Les éléments *PictureChoice* (cf. Figure 6-8), *TextPicture* (cf. Figure 6-9) et *WordChoice* (cf. Figure A-3) sont définis comme précédemment.

L'élément *LetterChoice* (cf. Figure A-7) indique comment choisir une lettre : de manière précise, *FileIdentifier*, ou en respectant les contraintes, *LetterConstraints*. Les contraintes permettent de ne sélectionner que des voyelles, que des consonnes, mais aussi de ne sélectionner que des lettres écrites en majuscules ou minuscules.

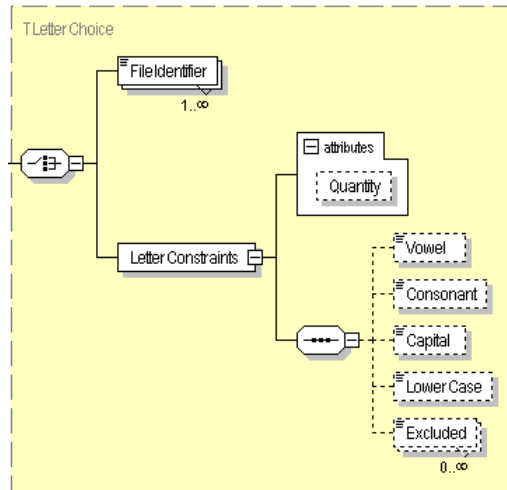


Figure A-7 : Choix d'une lettre dans un patron d'exercices.

L'élément *LineNumber* (cf. Figure A-6) est un nombre entier sans limite de taille. Il permet d'indiquer combien de fois l'apprenant devra recopier le modèle.

L'élément *Font* permet de choisir la police que devra utiliser l'apprenant pour recopier les mots ou les lettres. Le booléen *Example* permet d'indiquer s'il faut ou non fournir la liste des mots ou des lettres en respectant cette police.

L'élément *LinePicture* permet de choisir une illustration servant de support à la recopie par l'apprenant. Cette illustration est déterminée de la même manière que les illustrations associées à un patron (cf. Figure 6-8).

L'élément *ContentsNumber* détermine le nombre de contenus qu'un exercice contiendra. Ce nombre est toujours supérieur à un, et il est inférieur ou égal au nombre d'éléments *Content* contenus dans la structure d'exercices.

La Figure A-8 montre un exemple de structure d'exercices issue du patron opérationnel « Copie d'un modèle ». Cette structure permet de générer des exercices d'écriture pour des élèves de CP (cf. Exemple A-21 de l'Annexe D, page 297). La consigne de l'exercice est fournie dans l'élément *Wording*. L'exercice contiendra deux des quatre *Content* de la structure d'exercices. Chacun des *Content* indique la lettre à recopier, dans quelle police, le nombre de ligne à faire et fournit un modèle de ligne support à la recopie.


```

<ExercisesStructure>
  <Metadata>...</Metadata>
  - <WorkingOnIllustration>
    <Wording>Remplir les lignes d'écriture suivante :</Wording>
    - <Copy ContentsNumber="2">
      - <Content>
        - <LetterChoice>
          <FileIdentifier>a</FileIdentifier>
          </LetterChoice>
          <LinesNumber>1</LinesNumber>
          - <Font Example="true">
            <Name>Cursif</Name>
            <Style Bold="false" Italic="false" />
            <Size>12</Size>
            <Color>Black</Color>
          </Font>
          - <LinePicture>
            <FileIdentifier>13</FileIdentifier>
          </LinePicture>
        </Content>
        ...
      - <Content>
        - <LetterChoice>
          <FileIdentifier>B</FileIdentifier>
          </LetterChoice>
          <LinesNumber>1</LinesNumber>
          + <Font Example="true">
          + <LinePicture>
        </Content>
      </Copy>
    </WorkingOnIllustration>
  </ExercisesStructure>

```

Figure A-8 : Structure d'exercices issue du patron opérationnel « Copie d'un modèle » associé au patron « Travail sur illustration ».

PATRON C – ORGANISATION D'ÉLÉMENTS

Le patron « Organisation d'éléments » permet de créer des activités papier appartenant à la catégorie C de la typologie présentée sur la Figure 6-2. Ce patron, présenté Figure A-9, précise qu'une activité de ce type est formée d'une consigne, *Wording*, du contenu d'un des deux patrons opérationnels rattachés à l'organisation d'éléments, *Classify* ou *Group*, et éventuellement de documents, *Pictures* ou *Texts*.

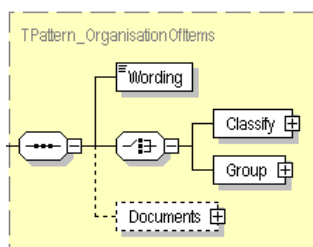


Figure A-9 : Patron d'exercices « Organisation d'éléments ».

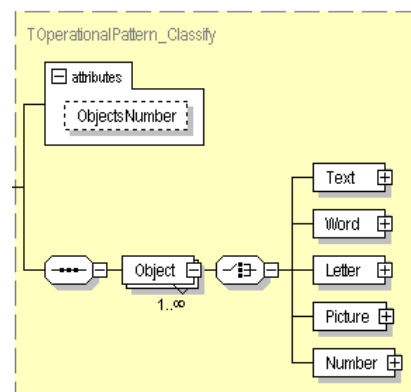


Figure A-10 : Patron opérationnel « Classement d'objets ».

Le patron opérationnel « Classement d'objets », présenté Figure A-10, décrit une activité comme un ensemble non vide d'éléments *Object* associé au nombre *ObjectsNumber*. Ces éléments s'ajoutent à ceux du patron « Organisation d'éléments ».

Chaque élément *Object* peut être un texte, une liste de mots, de lettres, d'images ou de nombres. Dans chaque cas, l'élément *Object* contiendra l'objet à classer et la manière dont il faut le mélanger pour le fournir à l'apprenant.

Le nombre *ObjectsNumber* détermine le nombre d'objets qu'un exercice contiendra. Ce nombre est toujours supérieur à un, et il est inférieur ou égal au nombre d'éléments *Object* contenus dans la structure d'exercices.

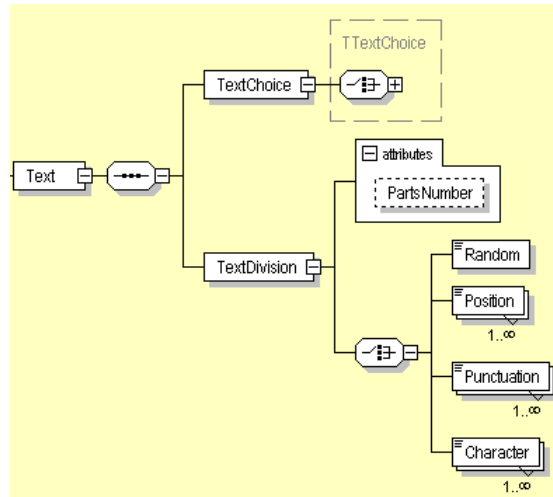


Figure A-11 : Sélection d'un objet de type *Text* dans le patron opérationnel « Classement d'objets ».

Si l'objet est de type *Text*, il est composé, comme le montre la Figure A-11, d'un texte et de la manière de le diviser pour le présenter à l'apprenant. Le texte est déterminé par la façon de le choisir, *TextChoice* (cf. Figure 6-2). Pour le découper, il y a plusieurs possibilités, *TextDivision*. Le texte peut être découpé aléatoirement, *Random*, à certaines positions prédéfinies, *Position*, à l'endroit de certains éléments de ponctuation, *Punctuation*, ou à l'endroit de certains caractères prédéfinis, *Character*. Dans tous les cas, le texte doit être découpé en autant de parties que le stipule le nombre *PartsNumber*.

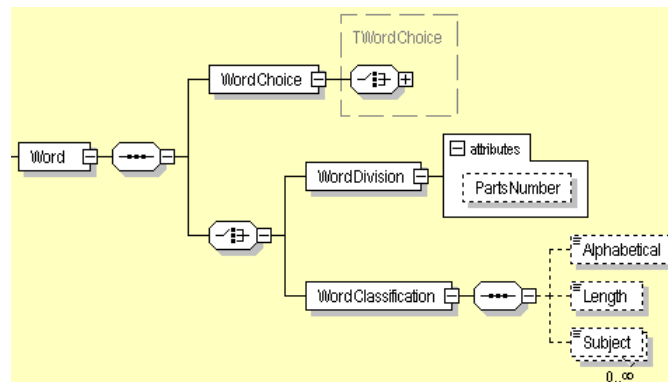


Figure A-12 : Sélection d'un objet de type *Word* dans le patron opérationnel « Classement d'objets ».

Si l'objet est de type *Word*, il est composé, comme le montre la Figure A-12, d'une liste non vide de mots, et soit de la manière de découper un mot, soit de la manière de mélanger la liste de mots. Les mots sont soit prédéfinis, soit choisis selon des contraintes (cf. Figure A-3).

Dans le cas d'une division d'un mot, *WordDivision*, le but pour l'apprenant sera de reconstruire le mot. Il faut alors indiquer en combien de parties le mot doit être découpé, *PartsNumber*. Ce nombre doit être au moins égal au nombre de mots de la liste, pour être sûr que chaque mot est

découpé au moins une fois. Il doit de plus être inférieur au nombre *MaxNumberOfParts* (cf. Équation 9-1) qui indique le nombre maximum de découpe possible.

$$\text{MaxNumberOfParts} = \frac{(\text{Nombre de lettres de tous les mots de la liste}) - (\text{Nombre de mots dans la liste})}{1}$$

Équation 9-1 : Nombre maximum de découpes possibles dans une liste de mots.

Dans le cas d'un classement des mots, *WordClassification*, il faut indiquer dans quel ordre l'apprenant doit les trier : alphabétique ou inversement, selon leur taille ou selon un thème. Ces contraintes de classement peuvent être combinées, pour, par exemple, demander de trier des noms de fruits par couleur, puis au sein d'une couleur, par ordre alphabétique.

Si l'objet est de type *Letter*, il est composé, comme le montre la Figure A-13, d'un ensemble de lettres et de la manière de la classer. Les lettres peuvent être choisies aléatoirement ou être prédéfinies, *LetterChoice* (cf. Figure A-7). Il est nécessaire d'avoir au moins deux lettres pour générer un exercice de classement. Le classement demandé à l'apprenant peut être l'ordre alphabétique et son inverse.

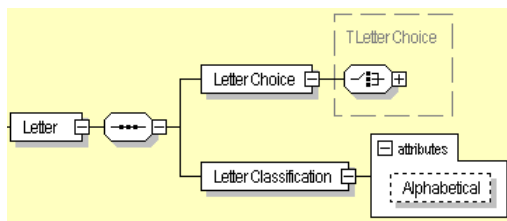


Figure A-13 : Sélection d'un objet de type *Letter* dans le patron opérationnel « Classement d'objets ».

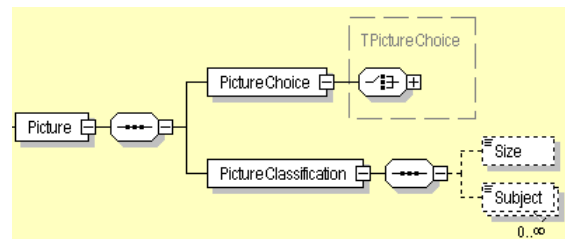


Figure A-14 : Sélection d'un objet de type *Picture* dans le patron opérationnel « Classement d'objets ».

Si l'objet est de type *Picture*, il est composé, comme le montre la Figure A-14, d'un ensemble non vide d'illustrations et de la manière de les classer. Chaque illustration est déterminée par la façon de la choisir, *PictureChoice* (cf. Figure 6-8). Pour les classer, il y a plusieurs possibilités, *PictureClassification*. Les illustrations peuvent être classées par taille, *Size*, et/ou selon un ou plusieurs thèmes, *Subject*.

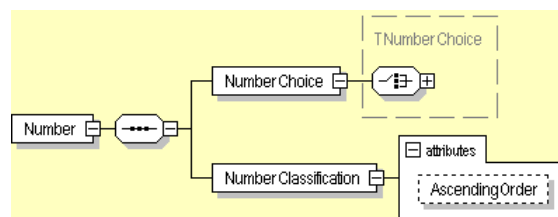


Figure A-15 : Sélection d'un objet de type *Number* dans le patron opérationnel « Classement d'objets ».

Enfin, si l'objet est de type *Number*, il est composé d'une liste d'au moins deux nombres et de la manière de les classer (cf. Figure A-15). Le classement peut se faire par ordre croissant ou décroissant, en précisant le booléen *AscendingOrder*. Les nombres peuvent soit être prédéfinis, *FixNumber*, soit choisis selon des contraintes, *NumberConstraints* (cf. Figure A-16). Ces contraintes peuvent porter sur l'intervalle dans lequel il faut les choisir, *Interval*, et sur le type de nombres, *Integer*, *Decimal*, *Relative* ou *Fraction*. Il est également possible d'interdire le choix de certains nombres, *Excluded*. Quelles que soient les contraintes, il faut préciser la quantité de nombres à choisir, *Quantity*.

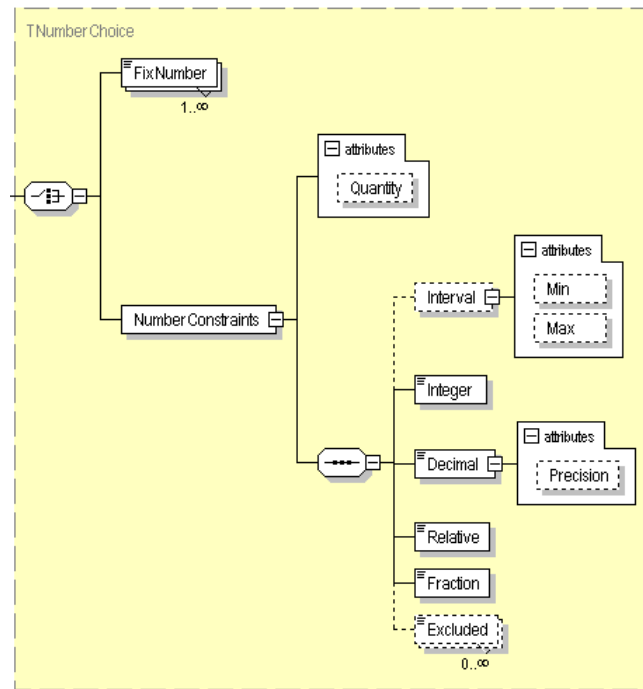


Figure A-16 : Choix d'un nombre dans un patron d'exercices.

```

<ExercisesStructure>
  <Metadata>...</Metadata>
  - <OrganisationOfItems>
    <Wording>Ranger les listes de nombres suivants par ordre croissant :</Wording>
    - <Classify ObjectsNumber="2">
      - <Object>
        - <Number>
          - <NumberChoice>
            - <NumberConstraints Quantity="7">
              <Interval Min="-10" Max="10" />
              <Integer>false</Integer>
              <Decimal>false</Decimal>
              <Relative>true</Relative>
              <Fraction>false</Fraction>
            </NumberConstraints>
          </NumberChoice>
          <NumberClassification AscendingOrder="true" />
        </Number>
      </Object>
      - <Object>
        - <Number>
          - <NumberChoice>
            - <NumberConstraints Quantity="8">
              <Integer>false</Integer>
              <Decimal Precision="0.1">true</Decimal>
              <Relative>true</Relative>
              <Fraction>true</Fraction>
            </NumberConstraints>
          </NumberChoice>
          <NumberClassification AscendingOrder="true" />
        </Number>
      </Object>
    </Classify>
  </OrganisationOfItems>
</ExercisesStructure>

```

Figure A-17 : Structure d'exercices issue du patron opérationnel « Classement d'objets » associé au patron « Organisation d'éléments ».

Regardons à présent deux exemples de structures d'exercices créées avec le patron opérationnel « Classement d'objets ». Le premier, fourni Figure A-17, permet de générer, entre autres, l'Exemple A-23 de l'Annexe D, page 298. La consigne de l'exercice est fournie dans l'élément

Wording. L'élément *ObjectsNumber* indique qu'il y a deux listes d'objets à classer. Ces objets sont des listes de nombres, *Number*. La première liste contient sept nombres compris entre -10 et 10. Ces nombres peuvent être des nombres relatifs, mais ni des fractions, ni des décimaux. La seconde liste contient huit nombres pouvant être des décimaux avec au plus un chiffre après la virgule, *Precision*, mais ils peuvent être également des fractions ou des nombres relatifs. Les deux listes sont à trier par ordre croissant, *AscendingOrder*.

Le second exemple, fourni Figure A-18, est une structure d'exercices permettant de générer des exercices où il est demandé de remettre un texte dans l'ordre. La consigne de l'exercice est fournie dans l'élément *Wording*. L'exercice portera sur un seul texte, comme l'indique l'élément *ObjectsNumber*, qui devra respecter les consignes stipulées dans *TextConstraints* : être écrit en anglais et être une lettre contenant entre 80 et 100 mots. Ce texte sera séparé en onze parties, comme l'indique l'élément *PartsNumber*. La séparation se fera sur les signes de ponctuations « .?! , ». Les Exemples A-25 et A-26 de l'Annexe D, page 298, sont issus de cette structure d'exercices.

```

<ExercisesStructure>
  <Metadata>...</Metadata>
  - <OrganisationOfItems>
    <Wording>Remettre cette lettre dans l'ordre :</Wording>
    - <Classify ObjectsNumber="1">
      - <Object>
        - <Text>
          - <TextChoice>
            - <TextConstraints Quantity="1">
              <Language>Anglais</Language>
              <Genre>Lettre</Genre>
              <WordNumber Min="80" Max="100" />
            </TextConstraints>
          </TextChoice>
        - <TextDivision PartsNumber="11">
          <Punctuation>.</Punctuation>
          <Punctuation>?</Punctuation>
          <Punctuation>!</Punctuation>
          <Punctuation>,</Punctuation>
        </TextDivision>
      </Text>
    </Object>
  </Classify>
</OrganisationOfItems>
</ExercisesStructure>

```

Figure A-18 : Structure d'exercices issue du patron opérationnel « Classement d'objets » associé au patron « Organisation d'éléments ».

Le patron opérationnel « Groupement d'objets », présenté Figure A-19, décrit une activité comme un ensemble non vide d'éléments *Object* associé au nombre *ObjectsNumber*. Ces éléments s'ajoutent à ceux du patron « Organisation d'éléments ».

Un élément *Object* contient des indications permettant de choisir les objets à grouper, *TextPicture* (cf. Figure 6-9), *WordChoice* (cf. Figure A-3), *LetterChoice* (cf. Figure A-7), *PictureChoice* (cf. Figure 6-8) ou *NumberChoice* (cf. Figure A-16) ainsi que la façon d'associer des objets d'un même type. De plus, si les objets sont de types différents, ils peuvent être associés de par leur équivalence ou leur thème, *Association*.

Grouper des textes peut se faire en fonction de (cf. Figure A-20) leur auteur, *Author*, la période à laquelle ils ont été écrits, *Period*, leur genre, (conte, poème, lettre, récit, etc.), *Gender*, leur sujet (par exemple chimie organique ou chimie inorganique), *Subject*, leur temps (par exemple présent de narration, imparfait ou futur), *Tense*, ou leur longueur, *Length*.

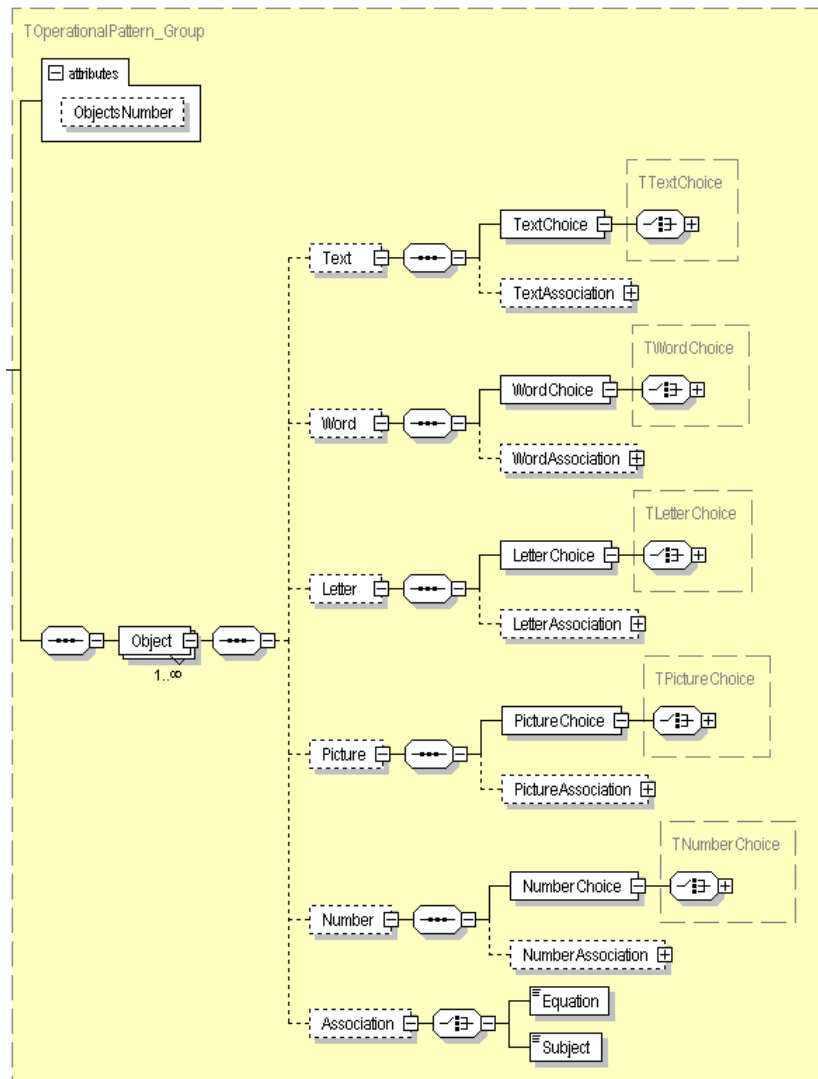


Figure A-19 : Patron opérationnel « Groupement d'objets ».

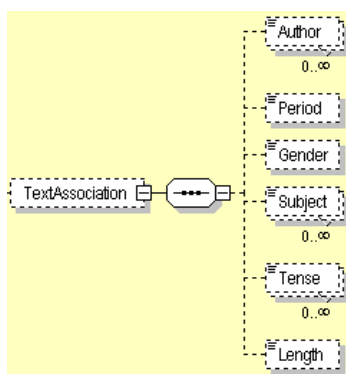


Figure A-20 : Critères possibles pour associer des textes dans le patron opérationnel « Groupement d'objets ».

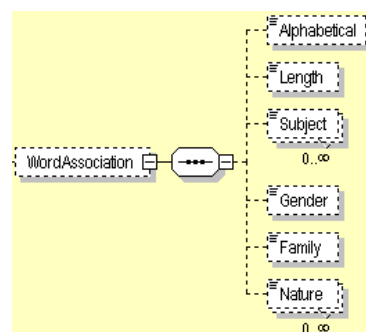


Figure A-21 : Critères possibles pour associer des mots dans le patron opérationnel « Groupement d'objets ».

Grouper des mots peut se faire en fonction de (cf. Figure A-21) leur première lettre, *Alphabetical*, leur longueur, *Length*, leurs thèmes, *Subject*, leur genre (féminin, masculin), *Gender*, leur famille (par exemple « dent », « dentiste », « dentaire », « dentifrice », « dentier » sont de la même famille), *Family*, ou leur nature (nom, adjectif, verbe, etc.), *Nature*.

Grouper des lettres peut se faire selon qu'elles sont des voyelles ou des consonnes, écrites en majuscule ou minuscule (cf. Figure A-22).

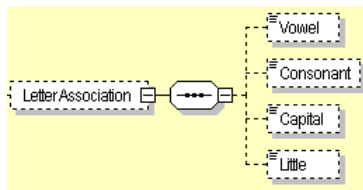


Figure A-22 : Critères possibles pour associer des lettres dans le patron opérationnel « Groupement d'objets ».

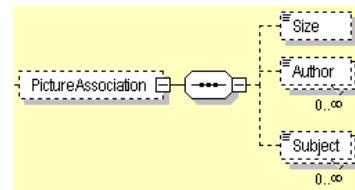


Figure A-23 : Critères possibles pour associer des illustrations dans le patron opérationnel « Groupement d'objets ».

Grouper des images peut se faire en fonction de (cf. Figure A-23) leur taille, leur auteur ou leurs thèmes.

Grouper des nombres peut se faire (cf. Figure A-24) selon l'intervalle auquel ils appartiennent, s'ils sont entiers, décimaux, relatifs ou sous forme de fractions.

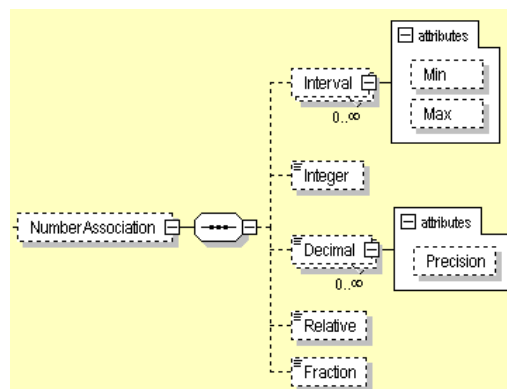


Figure A-24 : Critères possibles pour associer des nombres dans le patron opérationnel « Groupement d'objets ».

Nous allons à présent voir un exemple d'une structure d'exercices créée avec le patron opérationnel « Groupement d'objets ». Cette structure d'exercices, fournie Figure A-25, permet de générer l'Exemple A-28 de l'Annexe D, page 299. La consigne de l'exercice est fournie dans l'élément *Wording*. L'exercice contiendra six mots qui seront choisis parmi des noms de sports et six images selon les mêmes contraintes. L'Association demandée à l'apprenant sera de grouper les mots et les images équivalentes, ce qui veut dire dans cet exemple, grouper l'image avec le mot correspondant.

```

<ExercisesStructure>
  <Metadata>...</Metadata>
  - <OrganisationOfItems>
    <Wording>Reliez le nom d'un sport et le dessin qui le représente :</Wording>
    - <Group ObjectNumber="1">
      - <Object>
        - <Word>
          - <WordChoice>
            - <WordConstraints Quantity="6">
              <Nature>Nom</Nature>
              <Subject>Sport</Subject>
            </WordConstraints>
          </WordChoice>
        </Word>
      - <Picture>
        - <PictureChoice>
          - <PictureConstraints Quantity="6">
            <Keyword>Nom de sport</Keyword>
          </PictureConstraints>
        </PictureChoice>
      </Picture>
    - <Association>
      <Equation>true</Equation>
    </Association>
  </Object>
</Group>
</OrganisationOfItems>
</ExercisesStructure>

```

Figure A-25 : Structure d'exercices issue du patron opérationnel « Groupement d'objets » associé au patron « Organisation d'éléments ».

PATRON D – TABLEAU À DOUBLE ENTRÉE

Le patron « Tableau à double entrée » permet de créer des activités papier appartenant à la catégorie D de la typologie présentée sur la Figure 6-2. Ce patron, présenté Figure A-26, précise qu'une activité de ce type est formée d'une consigne, *Wording*, du contenu d'un des trois patrons opérationnels rattachés au tableau à double entrée, *ConjugationTable*, *MathematicsTable* ou *OtherDoubleEntryTable*, et éventuellement de documents, *Pictures* ou *Texts*.

Le patron opérationnel « Table de conjugaison », présenté Figure A-27, décrit une activité grâce aux éléments *Language*, *ConstraintsConjugation*, *PriorityVerb*, ainsi qu'aux deux entiers *TotalVerbsNumber* et *PriorityVerbsNumber*. Ces éléments s'ajoutent à ceux du patron « Tableau à double entrée ».

L'élément *Language* permet de préciser la langue des verbes concernés.

L'élément *ConstraintsConjugation* contient un ensemble de contraintes permettant de sélectionner les paramètres des tables de conjugaison. Ces contraintes portent sur la classe des verbes et leurs particularités, les personnes pour lesquelles il faudra conjuguer les verbes, et les temps demandés. La classe des verbes, *Class*, peut être « premier groupe, deuxième groupe, troisième groupe » pour la langue française, « régulier, irrégulier » pour la langue anglaise, etc. Les particularités, *VerbsMetadata*, permettent de sélectionner un verbe selon, par exemple, la terminaison qu'il possède, selon les exceptions dans sa table de conjugaison, selon son appartenance ou non à la liste des verbes à connaître pour un niveau scolaire donné, etc. Les personnes, *Person*, peuvent être une ou plusieurs des trois personnes du singulier et des trois personnes du pluriel. Les temps, *Tense*, sont propres à chaque langue : « présent, futur, imparfait... » pour le français, « prétérit, présent perfect... » pour l'anglais, etc.

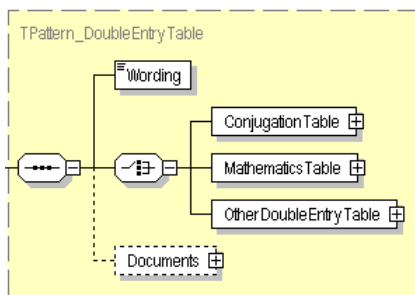


Figure A-26 : Patron d'exercices « Tableau à double entrée ».

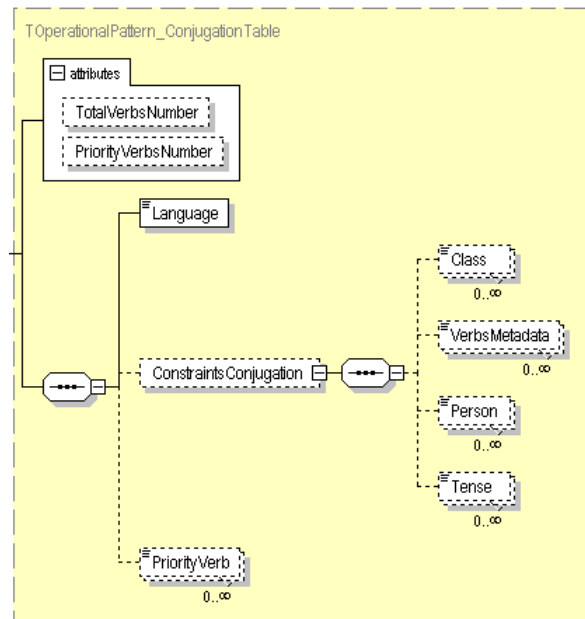


Figure A-27 : Patron opérationnel « Table de conjugaison ».

Chaque élément *PriorityVerb* contient un verbe précis, dit verbe prioritaire. Ainsi, les nombres *TotalVerbsNumber* et *PriorityVerbsNumber* indiquent que l'exercice créé devra contenir autant de verbes que l'indique l'entier *TotalVerbsNumber* et que parmi ces verbes, n verbes doivent être pris dans la liste des verbes prioritaires, n étant l'entier *PriorityVerbsNumber*.

La Figure A-28 contient un exemple de structure d'exercices créé à partir du patron opérationnel « Table de conjugaison ». Cette structure d'exercices permet de créer des exercices comme celui de l'Exemple A-30 de l'Annexe D, page 299.

```

<ExercisesStructure>
  <Metadata>...</Metadata>
  - <DoubleEntryTable>
    <Wording>Conjuguer les verbes ##VERBS## au présent de l'indicatif :</Wording>
    - <ConjugationTable TotalVerbsNumber="3" PriorityVerbsNumber="1">
      <Language>Français</Language>
      - <ConstraintsConjugation>
        <Class>Premier groupe</Class>
        <Class>Troisieme groupe</Class>
        <Person>Toutes</Person>
        <Tense>Présent de l'indicatif</Tense>
      </ConstraintsConjugation>
      <PriorityVerb>Manger</PriorityVerb>
      <PriorityVerb>Voyager</PriorityVerb>
    </ConjugationTable>
  </DoubleEntryTable>
</ExercisesStructure>

```

Figure A-28 : Structure d'exercices issue du patron opérationnel « Table de conjugaison » associé au patron « Tableau à double entrée ».

La consigne de l'exercice est fournie dans l'élément *Wording*. La chaîne de caractère « ##VERBS## » devra être remplacée par la liste des verbes à conjuguer. Il sera demandé à l'apprenant de conjuguer des verbes en français comme indiqué dans *Language*, au présent de l'indicatif spécifié dans *Tense*, et à toutes les personnes notées dans *Person*. L'exercice contiendra trois verbes comme le précise l'élément *TotalVerbsNumber*. Parmi ces trois verbes, un sera issu de la liste des verbes prioritaires, *PriorityVerb* et les deux autres seront choisis en

prenant des verbes du premier et du troisième groupe comme indiqué dans *Class*, sans autre contrainte particulière.

Le patron opérationnel « Table de mathématiques », présenté Figure A-29, décrit une activité grâce aux éléments *Operator*, *ConstraintsMathematics*, *PriorityNumber*, ainsi qu'aux deux entiers *TotalNumbersNumber* et *PriorityNumbersNumber*. Ces éléments s'ajoutent à ceux du patron « Tableau à double entrée ».

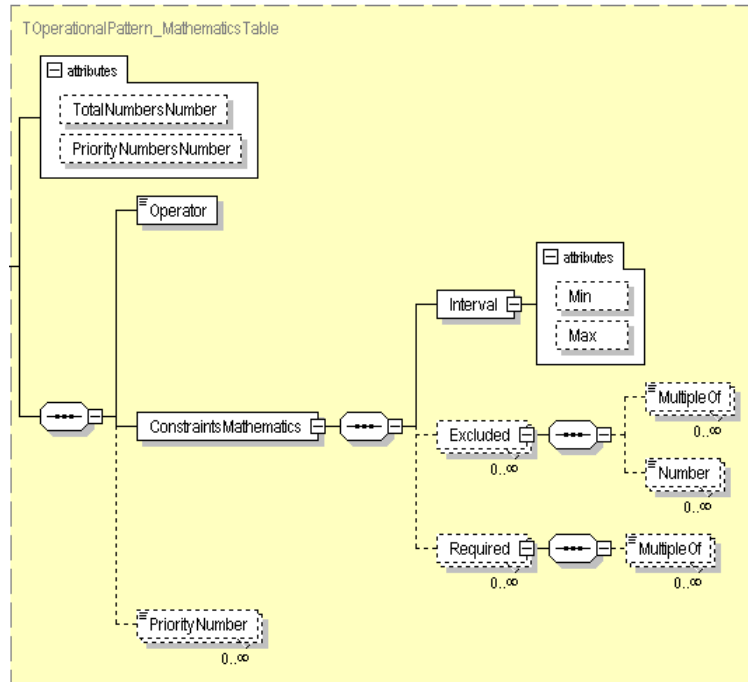


Figure A-29: Patron opérationnel « Table de mathématiques ».

L'élément *Operator* permet de préciser le type de tables demandées, par exemple addition ou multiplication.

L'élément *ConstraintsMathematics* contient un ensemble de contraintes permettant de sélectionner les nombres contenus dans les tables de mathématiques. Ces contraintes portent sur l'intervalle dans lequel les nombres doivent être choisis, *Interval*, mais également sur des informations permettant d'inclure ou d'exclure certains nombres. Ainsi, l'élément *Excluded* permet d'interdire le choix de certains multiples, *MultipleOf*, ou directement de certains nombres, *Number*. De même, l'élément *Required* permet de limiter le choix à certains multiples, *MultipleOf*. Il est ainsi possible de demander la table de multiplication d'un nombre compris entre 1 et 10, qui ne soit pas multiple de 2 mais qui soit multiple de 3. Cela revient à demander la table de 3 ou de 9.

Chaque élément *PriorityNumber* contient un nombre, dit nombre prioritaire. Ainsi, l'exercice créé devra contenir autant de tables que l'indique l'entier *TotalNumbersNumber* et que parmi ces tables, il doit y avoir n tables portant sur des nombres prioritaires, n étant l'entier *PriorityNumbersNumber*.

La Figure A-30 contient un exemple de structure d'exercices créée à partir du patron opérationnel « Table de mathématiques ». Cette structure d'exercices permet de créer des exercices comme celui de l'Exemple A-33 de l'Annexe D, page 300. La consigne de l'exercice est fournie dans l'élément *Wording*. Il sera demandé à l'apprenant de fournir des tables de multiplication comme indiqué dans *Operator*. L'exercice contiendra trois tables, valeur donnée

par *TotalNumbersNumber*. Parmi ces trois tables, une sera issue de la liste des nombres prioritaires, *PriorityNumber*, qui contient les nombres 3, 5 et 7, et les deux autres seront choisis en prenant des nombres inférieurs à 20, *Interval*, qui ne sont pas des multiples de 10, ni les nombres 0 ou 1, informations précisées dans *Excluded*.

```

<ExercisesStructure>
  <Metadata>...</Metadata>
  - <DoubleEntryTable>
    <Wording>Écrire la table de multiplication des nombres :</Wording>
    - <MathematicsTable TotalNumbersNumber="3" PriorityNumbersNumber="1">
      <Operator>x</Operator>
      - <ConstraintsMathematics>
        <Interval Max="20" />
        - <Excluded>
          <MultipleOf>10</MultipleOf>
          <Number>0</Number>
          <Number>1</Number>
        </Excluded>
      </ConstraintsMathematics>
      <PriorityNumber>3</PriorityNumber>
      <PriorityNumber>5</PriorityNumber>
      <PriorityNumber>7</PriorityNumber>
    </MathematicsTable>
  </DoubleEntryTable>
</ExercisesStructure>

```

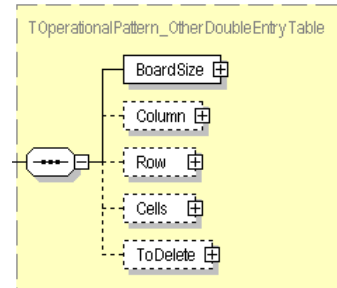


Figure A-30 : Structure d'exercices issue du patron opérationnel « Table de mathématiques » associé au patron « Tableau à double entrée ».

Figure A-31: Patron opérationnel « Tableaux à double entrées ».

Le patron opérationnel « Autre tableau à double entrée », présenté Figure A-31, décrit un tableau grâce à sa taille, *BoardSize*, éventuellement par le contenu de ses colonnes, *Column*, de ses lignes, *Row*, et de ses cellules, *Cells*, ainsi que l'élément *ToDelete*. Ces éléments s'ajoutent à ceux du patron « Tableau à double entrée ».

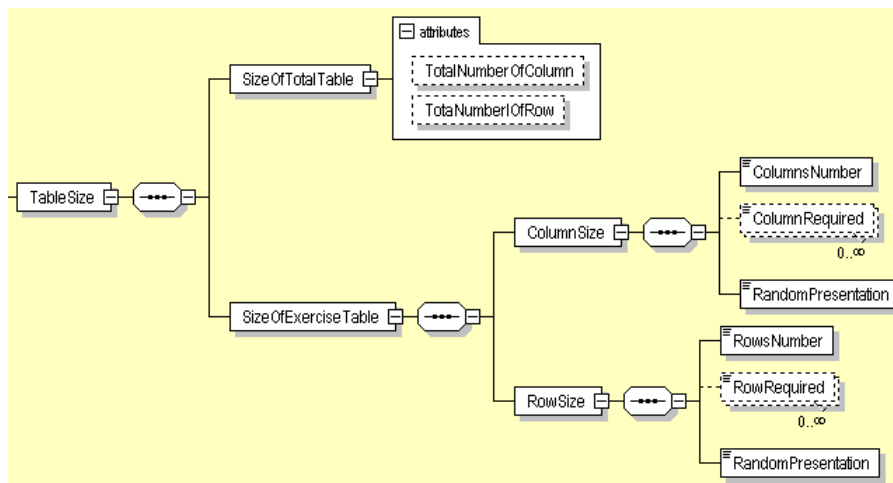


Figure A-32 : Définition de la taille d'un tableau dans le patron opérationnel « Autre tableau à double entrée ».

La taille d'un tableau (cf. Figure A-32) contient deux éléments : sa taille théorique, *SizeOfTotalTable*, et la taille du tableau contenu effectivement dans l'exercice, *SizeOfExerciseTable*. Chacune des deux tailles est définie par deux entiers : le nombre de colonnes, *TotalNumberOfColumn* et *ColumnsNumber*, et le nombre de lignes, *TotalNumberOfRow* et *RowsNumber*. La taille du tableau contenu dans l'exercice, *SizeOfExerciseTable*, peut également contenir les lignes et colonnes qui doivent obligatoirement apparaître dans l'exercice,

ColumnRequired et *RowRequired*, ainsi que deux booléens, *RandomPresentation*, indiquant si l'ordre des lignes et des colonnes peut être modifié.

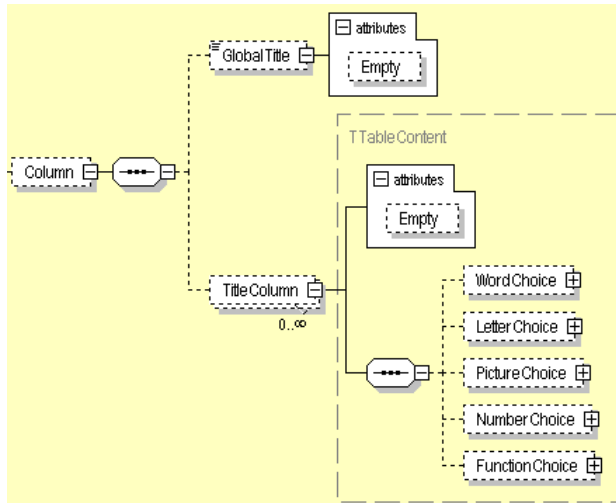


Figure A-33 : Définition des titres des colonnes d'un tableau dans le patron opérationnel « Autre tableau à double entrée ».

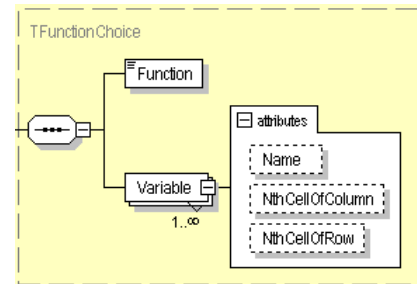


Figure A-34 : Définition d'une expression mathématique dans le patron opérationnel « Autre tableau à double entrée ».

La description des colonnes (cf. Figure A-33) se fait via un ensemble de titres. Le premier titre, *GlobalTitle*, décrit le contenu de toutes les colonnes, pour l'Exemple A-34 de l'Annexe D, page 300, ce titre est « couleurs ». Les autres titres, *TitleColumn*, sont les titres de chacune des colonnes. Il y en a $(TotalNumberOfColumn - 1)$. Ces titres peuvent être des mots (cf. Figure A-3), des lettres (cf. Figure A-7), des images (cf. Figure 6-8), des nombres (cf. Figure A-16), prédéfinis ou choisis selon des contraintes, mais aussi des expressions mathématiques mettant en jeu des variables (cf. Figure A-34). Pour l'Exemple A-34 de l'Annexe D, ces titres sont « ROUGE, BLEU, VERT, JAUNE » associés à une image présentant la couleur.

Les expressions mathématiques, *FunctionChoice*, sont des expressions mettant en jeu des variables définies dans n'importe quelle autre case du tableau. Par exemple, le titre d'une colonne peut être la valeur de la fonction « $12x+3$ », sachant que la valeur de la variable x est donnée dans la troisième case de la première colonne. La seule contrainte est que les cases servant de variables contiennent des nombres, *NumberChoice*.

La description des lignes d'un tableau respecte le même schéma que celle des colonnes (cf. Figure A-35). Il y a autant de *TitleRow* que la valeur de $(TotalNumberOfRow - 1)$.

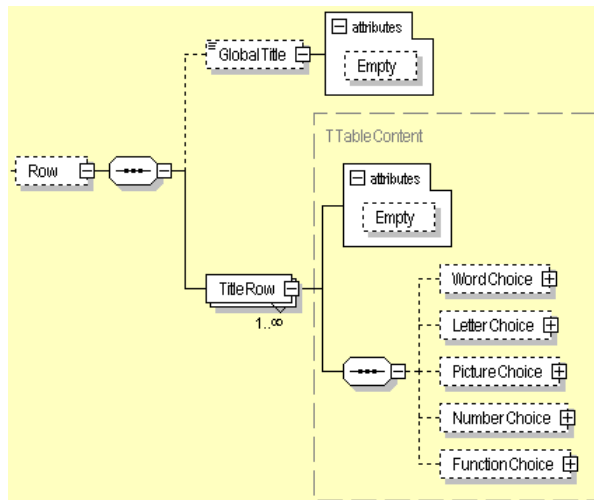


Figure A-35 : Définition des titres des lignes d'un tableau dans le patron opérationnel « Autre tableau à double entrée ».

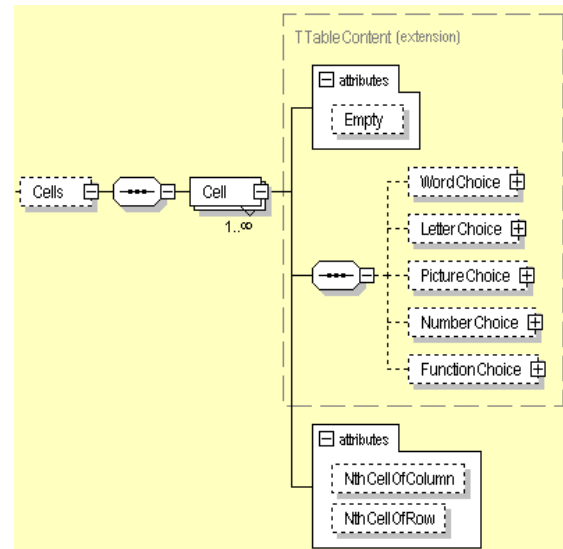


Figure A-36 : Définition des cellules d'un tableau dans le patron opérationnel « Autre tableau à double entrée ».

Un tableau peut de plus contenir la description du contenu de chaque cellule (cf. Figure A-36). Une cellule, *Cell*, est décrite de la même façon qu'un titre de ligne ou de colonne, avec en plus ses coordonnées dans le tableau, *NthCellofColum* et *NthCellofRow*. Le contenu des cellules est soit calculé automatiquement, soit prédéfini.

Chaque titre du tableau et chaque cellule contiennent un booléen *Empty*. Ce booléen indique si la case du titre ou de la cellule sera vide dans l'exercice donné à l'apprenant. Ainsi, il est possible de créer des tableaux dont les titres des lignes et des colonnes sont fournis, l'apprenant devra remplir les cases, mais il est également possible de créer un tableau à « trous », l'apprenant devant compléter les trous, que ce soit des titres ou des cellules. Ces deux exemples sont illustrés par les structures d'exercices des figures A-38 et A-39.

Dans le cas où le booléen *Empty* de tous les titres et de toutes les cellules est à « *false* », c'est-à-dire que toutes les cases du tableau sont remplies par défaut, il est possible de définir l'élément *ToDelete* (cf. Figure A-37). Cet élément permet de créer un tableau à « trous ». Il comprend l'entier *NumberEmpty* qui indique le nombre de cases du tableau à vider. Ce nombre peut être augmenté de deux booléens indiquant s'il faut laisser au moins une case non vidée par ligne, *OnePerRow*, et/ou par colonne, *OnePerColumn*, et par la liste des lignes, *Row*, et/ou des colonnes, *Column*, sur lesquelles il ne faut rien effacer.

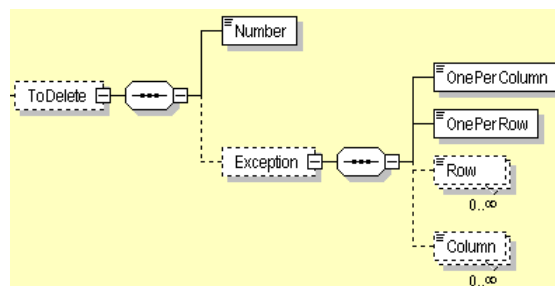


Figure A-37 : Élément *ToDelete* du patron opérationnel « Autre tableau à double entrée ».

La structure d'exercices de la Figure A-38 permet de créer l'Exemple A-34 de l'Annexe D, page 300, qui correspond au cas où les titres des colonnes et des lignes sont fournis et où l'apprenant

doit compléter les cellules du tableau. La consigne de l'exercice est fournie dans *Wording*. Le tableau de l'exercice contiendra 5 colonnes et 4 lignes, comme indiqué dans *TableSize*. Les colonnes contiendront des couleurs comme le précise l'élément *GlobalTitle*, et chaque titre de colonne sera composé du nom de la couleur en majuscules et d'une image de la base d'images représentant une zone de la couleur correspondante. Les lignes contiendront des formes géométriques choisies dans la base d'images. Cette structure d'exercices ne contient pas le contenu des cellules, aucune correction ne pourra donc être proposée à l'enseignant et/ou à l'apprenant.

```

<ExercisesStructure>
  <Metadata>...</Metadata>
  - <DoubleEntryTable>
    <Wording>Complète le tableau.</Wording>
    - <OtherDoubleEntryTable>
      - <TableSize>
        <SizeOfTotalTable TotalNumberOfColumn="5"
          TotalNumberOfRow="4" />
      - <SizeOfExerciseTable>
        - <ColumnSize>
          <ColumnsNumber>5</ColumnsNumber>
          <RandomPresentation>true</RandomPresentation>
        </ColumnSize>
        - <RowSize>
          <RowsNumber>4</RowsNumber>
          <RandomPresentation>true</RandomPresentation>
        </RowSize>
      </SizeOfExerciseTable>
    </TableSize>
  - <Column>
    <GlobalTitle Empty="false">Couleurs</GlobalTitle>
  - <TitleColumn Empty="false">
    - <WordChoice>
      <FileIdentifier>ROUGE</FileIdentifier>
    </WordChoice>
    - <PictureChoice>
      <FileIdentifier>50</FileIdentifier>
    </PictureChoice>
  </TitleColumn>
  + <TitleColumn Empty="false">
  + <TitleColumn Empty="false">
  + <TitleColumn Empty="false">
</Column>
  - <Row>
    <GlobalTitle>Formes</GlobalTitle>
  - <TitleRow Empty="false">
    - <PictureChoice>
      - <PictureConstraints Quantity="1">
        <Keyword>Formes géométriques</Keyword>
      </PictureConstraints>
    </PictureChoice>
  </TitleRow>
  + <TitleRow Empty="false">
  + <TitleRow Empty="false">
</Row>
</OtherDoubleEntryTable>
</DoubleEntryTable>
</ExercisesStructure>

```

Figure A-38 : Structure d'exercices issue du patron opérationnel « Autre tableau à double entrée » associé au patron « Tableau à double entrée ».

La structure d'exercices de la Figure A-39 permet de créer un exercice de français où l'apprenant doit remplir un tableau à « trous ». La consigne de l'exercice est fournie dans *Wording*. Le tableau de l'exercice contiendra 4 colonnes et 12 lignes comme indiqué dans *TableSize*. Les titres des colonnes sont fournis dans *TitleColumn*. Les lignes n'auront pas de titre. Chacune des 12 lignes, sauf la ligne de titre, est composée dans la première colonne d'un nom (cf. ① sur la Figure A-39), dans la seconde d'un verbe (cf. ② sur la Figure A-39), dans la troisième d'un adjectif (cf. ③ sur la Figure A-39) et dans la quatrième d'un adverbe (cf. ④ sur la Figure A-39). Ces mots sont choisis en fonction de contraintes spécifiées dans *WordChoice*. Pour former le tableau à trous, il faut vider 35 cases, comme le stipule l'élément *ToDelete*, en tenant compte des *Exception* : laisser au moins une cellule intacte par ligne, une cellule intacte par colonne et ne vider aucune cellule de la première ligne. Cette structure d'exercices permet de créer divers exercices dont l'Exemple A-36 de l'Annexe D, page 301.

```

<ExercisesStructure>
  <Metadata>...</Metadata>
  - <DoubleEntryTable>
    <Wording>Serais-tu capable d'écrire sur chaque ligne des mots
    de la même famille ?</Wording>
    - <OtherDoubleEntryTable>
      - <TableSize>
        <SizeOfTotalTable TotalNumberOfColumn="4"
        TotalNumberOfRow="12" />
      - <SizeOfExerciseTable>
        - <ColumnSize>
          <ColumnsNumber>4</ColumnsNumber>
          <RandomPresentation>false</RandomPresentation>
        </ColumnSize>
        - <RowSize>
          <RowsNumber>12</RowsNumber>
          <RowRequired>1</RowRequired>
          <RandomPresentation>>true</RandomPresentation>
        </RowSize>
      </SizeOfExerciseTable>
    </TableSize>
    - <Column>
      - <TitleColumn Empty="false">
        - <WordChoice>
          <FileIdentifier>Noms</FileIdentifier> ①
        </WordChoice>
      </TitleColumn>
      - <TitleColumn Empty="false">
        - <WordChoice>
          <FileIdentifier>Verbes</FileIdentifier> ②
        </WordChoice>
      </TitleColumn>
      - <TitleColumn Empty="false">
        - <WordChoice>
          <FileIdentifier>Adjectifs</FileIdentifier> ③
        </WordChoice>
      </TitleColumn>
      - <TitleColumn Empty="false">
        - <WordChoice>
          <FileIdentifier>Adverbes</FileIdentifier> ④
        </WordChoice>
      </TitleColumn>
    </Column>
  </DoubleEntryTable>
  - <Cells>
    - <Cell Empty="false" NthCellOfColumn="1" NthCellOfRow="2">
      <WordChoice>
        - <WordConstraints>
          <Language>français</Language>
          <Nature>nom</Nature>
          <Family>secheresse</Family>
        </WordConstraints>
      </WordChoice>
    </Cell>
    - <Cell Empty="false" NthCellOfColumn="2" NthCellOfRow="2">
      <WordChoice>
        - <WordConstraints>
          <Language>français</Language>
          <Nature>verbe</Nature>
          <Family>secheresse</Family>
        </WordConstraints>
      </WordChoice>
    </Cell>
    - <Cell Empty="false" NthCellOfColumn="3" NthCellOfRow="2">
      <WordChoice>
        - <WordConstraints>
          <Language>français</Language>
          <Nature>adjectif</Nature>
          <Family>secheresse</Family>
        </WordConstraints>
      </WordChoice>
    </Cell>
    - <Cell Empty="false" NthCellOfColumn="4" NthCellOfRow="2">
      <WordChoice>
        - <WordConstraints>
          <Language>français</Language>
          <Nature>adverbe</Nature>
          <Family>secheresse</Family>
        </WordConstraints>
      </WordChoice>
    </Cell>
    ...
  </Cells>
  - <ToDelete>
    <Number>35</Number>
    - <Exception>
      <OnePerColumn>>true</OnePerColumn>
      <OnePerRow>true</OnePerRow>
      <Row>1</Row>
    </Exception>
  </ToDelete>
</OtherDoubleEntryTable>
</DoubleEntryTable>
</ExercisesStructure>

```

Figure A-39 : Structure d'exercices issue du patron opérationnel « Autre tableau à double entrée » associé au patron « Tableau à double entrée ».

Un dernier exemple d'utilisation du tableau à double entrée est la structure d'exercices de la Figure A-40. Cette structure d'exercices permet de créer un exercice de mathématiques. Le tableau à double entrée ne contient aucun trou, l'apprenant devant s'en servir pour identifier les paramètres d'une fonction. L'énoncé de l'exercice est fourni dans *Wording*. Le tableau complet contient trois colonnes et six lignes, informations données dans *TableSize*, mais le tableau fourni dans l'exercice ne contiendra que deux colonnes dont la première, comme indiqué dans *ColumnSize*, et quatre lignes dont la première et la quatrième, comme indiqué dans *RowSize*.

```

<ExercisesStructure>
  <Metadata>...</Metadata>
  - <DoubleEntryTable>
    <Wording>Trouver les valeurs des paramètres a, b, c de la
    fonction décrite dans la deuxième colonne du
    tableau.</Wording>
    - <OtherDoubleEntryTable>
      - <TableSize>
        <SizeOfTotalTable TotalNumberOfColumn="3"
        TotalNumberOfRow="6" />
        - <SizeOfExerciseTable>
          - <ColumnSize>
            <ColumnsNumber>2</ColumnsNumber> ⑤
            <ColumnRequired>1</ColumnRequired>
            <RandomPresentation>false</RandomPresentation>
          </ColumnSize>
          - <RowSize>
            <RowsNumber>4</RowsNumber>
            <RowRequired>1</RowRequired> ⑥
            <RowRequired>4</RowRequired>
            <RandomPresentation>false</RandomPresentation>
          </RowSize>
        </SizeOfExerciseTable>
      </TableSize>
    - <Column>
      - <TitleColumn Empty="false">
        - <WordChoice>
          <FileIdentifier>x</FileIdentifier>
        </WordChoice>
      </TitleColumn>
      - <TitleColumn Empty="false">
        - <WordChoice>
          <FileIdentifier>ax2 + bx + c</FileIdentifier>
        </WordChoice>
      </TitleColumn>
      - <TitleColumn Empty="false">
        - <WordChoice>
          <FileIdentifier>ax2 + bx + c</FileIdentifier>
        </WordChoice>
      </TitleColumn>
    </Column>
  </OtherDoubleEntryTable>
  </DoubleEntryTable>
  - <Cells>
    - <Cell Empty="false" NthCellOfColumn="1" NthCellOfRow="2">
      - <NumberChoice>
        - <NumberConstraints Quantity="1"> ①
          <Interval Min="-10" Max="10" />
          <Integer>true</Integer>
          <Decimal>false</Decimal>
          <Relative>true</Relative>
          <Fraction>false</Fraction>
          <Excluded>0</Excluded>
        </NumberConstraints>
      </NumberChoice>
    </Cell>
    + <Cell Empty="false" NthCellOfColumn="1" NthCellOfRow="3">
    - <Cell Empty="false" NthCellOfColumn="1" NthCellOfRow="4">
      - <NumberChoice>
        <FixNumber>0</FixNumber> ②
      </NumberChoice>
    </Cell>
    + <Cell Empty="false" NthCellOfColumn="1" NthCellOfRow="5">
    + <Cell Empty="false" NthCellOfColumn="1" NthCellOfRow="6">
    - <Cell Empty="false" NthCellOfColumn="2" NthCellOfRow="2">
      - <FunctionChoice>
        <Function>X2 - 2X + 3</Function>
        <Variable Name="X" NthCellOfColumn="1" ③
        NthCellOfRow="2" />
      </FunctionChoice>
    </Cell>
    + <Cell Empty="false" NthCellOfColumn="2" NthCellOfRow="3">
    + <Cell Empty="false" NthCellOfColumn="2" NthCellOfRow="4">
    + <Cell Empty="false" NthCellOfColumn="2" NthCellOfRow="5">
    + <Cell Empty="false" NthCellOfColumn="2" NthCellOfRow="6">
    - <Cell Empty="false" NthCellOfColumn="3" NthCellOfRow="2">
      - <FunctionChoice>
        <Function>3X2 + 4X - 7</Function>
        <Variable Name="X" NthCellOfColumn="1" ④
        NthCellOfRow="2" />
      </FunctionChoice>
    </Cell>
    + <Cell Empty="false" NthCellOfColumn="3" NthCellOfRow="3">
    + <Cell Empty="false" NthCellOfColumn="3" NthCellOfRow="4">
    + <Cell Empty="false" NthCellOfColumn="3" NthCellOfRow="5">
    + <Cell Empty="false" NthCellOfColumn="3" NthCellOfRow="6">
  </Cells>
</OtherDoubleEntryTable>
</DoubleEntryTable>
</ExercisesStructure>
  
```

Figure A-40 : Structure d'exercices issue du patron opérationnel « Autre tableau à double entrée » associé au patron « Tableau à double entrée ».

La description du tableau contenu dans *Column* et *Cells* permet de créer le tableau de la Figure A-41. Les cellules de la première colonne, excepté celle de titre, sont des nombres. Les cellules des lignes 2, 3, 5 et 6 contiennent des nombres entiers ou relatifs, choisis aléatoirement entre -10 et 10 (cf. ① sur la Figure A-40), tandis que la case de la ligne 4 contient forcément le chiffre zéro (cf. ② sur la Figure A-40). Les cellules des deuxième et troisième colonnes, exceptées les cases de titre, sont calculées automatiquement en fonction des valeurs de la première colonne en respectant les fonctions $f(x) = x^2 - 2x + 3$ (cf. ③ sur la Figure A-40) et $g(x) = 3x^2 + 4x - 7$ (cf. ④ sur la Figure A-40).

x	ax ² + bx + c	ax ² + bx + c
-2	11	-3
-1	6	-8
0	3	-7
1	2	0
2	3	13

Figure A-41 : Représentation de la totalité du tableau décrit dans la structure d'exercices de la Figure A-40.

Les contraintes sur le choix des colonnes présentées à l'apprenant permettent de ne fournir qu'une seule des colonnes contenant la description de la fonction à trouver, soit $f(x)$, soit $g(x)$ (cf. © sur la Figure A-40). Les contraintes sur le choix des lignes permettent de ne présenter que quatre lignes dont la ligne de titre et la ligne où x vaut zéro (cf. © sur la Figure A-40). Les deux autres lignes sont choisies aléatoirement. L'Exemple A-37 de l'Annexe D, page 301, peut être créé à partir de cette structure d'exercices.

PATRON H – QUESTIONS / RÉPONSES

Le patron « Questions / réponses » permet de créer des activités papier appartenant à la catégorie H de la typologie présentée sur la Figure 6-2. Ce patron, présenté Figure A-42, précise qu'une activité de ce type est formée d'une consigne, *Wording*, et éventuellement d'une liste de questions, *Questions*, et de documents, *Pictures* ou *Texts*.

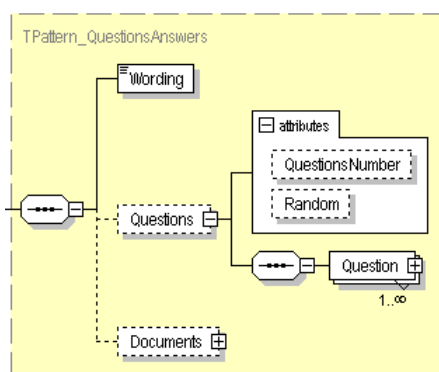


Figure A-42 : Patron d'exercices « Questions / réponses ».

Les structures d'exercices ne contenant qu'un énoncé, *Wording*, permettent de créer des exercices de dissertation. Les structures d'exercices contenant un énoncé, *Wording*, et un ou plusieurs documents, *Pictures* et/ou *Texts*, permettent de créer des exercices de lecture, de récitation, d'apprentissage (cours, poésie, etc.) ou des exercices d'étude de documents sans question complémentaire. Les structures d'exercices contenant une liste de questions, *Questions*, permettent de faire des QCM, des QROC ou des études de documents contenant des questions précises. Des exemples de structures de ce type sont fournis après la description formelle des éléments constituant la liste de questions.

La liste de questions, *Questions*, contient un ensemble non vide d'éléments *Question*, ainsi qu'un entier *QuestionsNumber* permettant de préciser le nombre de questions que contiendront les exercices créés et un booléen *Random* indiquant si les questions doivent être présentées dans l'ordre donné ou non. Cet entier est un entier positif inférieur ou égal au nombre d'éléments *Question* définis. Il est, de cette manière, possible de créer une liste contenant 20 questions, mais de ne fournir que 10 questions à l'apprenant.

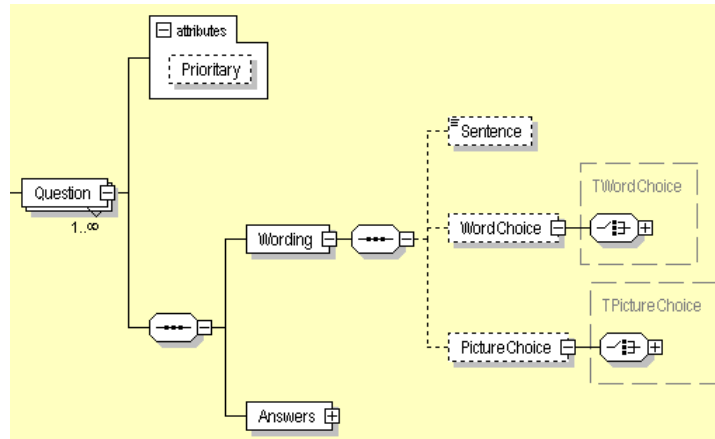


Figure A-43 : Élément *Question* du patron d'exercices « Questions / réponses ».

Chaque élément *Question* (cf. Figure A-43) contient l'énoncé de la question, *Wording*, et une liste d'éléments de réponse à cette question, *Answers*. Le booléen *Prioritary* permet de préciser que cette question doit apparaître prioritairement dans la liste des questions fournies à l'apprenant. L'énoncé de la question, *Wording*, peut être composé d'un ou plusieurs des trois éléments suivants : une phrase, *Sentence* ; un mot prédéfini ou choisi selon des contraintes, *WordChoice* (cf. Figure A-3); une image prédéfinie ou choisie selon des contraintes, *PictureChoice* (cf. Figure 6-8).

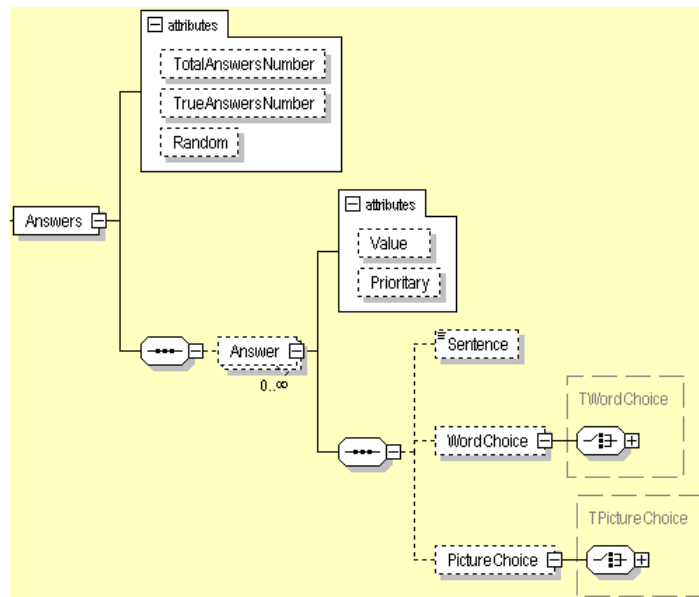


Figure A-44 : Élément *Answers* du patron d'exercices « Questions / réponses ».

La liste d'éléments de réponse *Answers* est composée d'un ensemble non vide de réponses, *Answer*, du booléen *Random* et des deux entiers *TotalAnswersNumber* et *TrueAnswersNumber* (cf. Figure A-44). L'entier *TotalAnswersNumber* permet de préciser le nombre de réponses à fournir à l'apprenant. Il est compris entre zéro, pour les QROC, et le nombre d'éléments *Answer* définis, pour les QCM. L'entier *TrueAnswersNumber* indique le nombre de réponses justes à fournir. Ce nombre peut être égal à zéro. Le booléen *Random* indique si les réponses doivent être présentées dans l'ordre donné.

Une réponse, *Answer*, peut être composée, comme pour l'énoncé d'une question, d'un ou plusieurs des trois éléments suivants : une phrase, *Sentence* ; un mot prédéfini ou choisi selon des contraintes, *WordChoice* ; une image prédéfinie ou choisie selon des contraintes,

PictureChoice. Une réponse est de plus étiquetée comme vraie ou fausse, *Value*, et comme devant ou non être mise prioritairement dans la liste des réponses, *Prioritary*.

```
<ExercicesStructure>
  <Metadata>...</Metadata>
  - <QuestionsAnswers>
    <Wording>Pour la liberté,
      le temps est-il un
      obstacle ou un moyen ?
    </Wording>
  </QuestionsAnswers>
</ExercicesStructure>
```

Figure A-45 : Structure d'exercices issue du patron « Questions / réponses ».

```
<ExercicesStructure>
  <Metadata>...</Metadata>
  - <QuestionsAnswers>
    <Wording>Apprendre la poésie suivant par coeur
      pour la séance prochaine.</Wording>
    - <Documents>
      - <Text>
        - <TextConstraints Quantity="1">
          <Language>Français</Language>
          <Genre>Poesie</Genre>
          <WordNumber Min="100" Max="200" />
        </TextConstraints>
      </Text>
    </Documents>
  </QuestionsAnswers>
</ExercicesStructure>
```

Figure A-46 : Structure d'exercices issue du patron « Questions / réponses ».

La structure d'exercices présentée sur la Figure A-45 permet de créer un exercice de dissertation dont l'énoncé est fourni dans l'élément *Wording*. La structure d'exercices de la Figure A-46 permet de créer des exercices demandant d'apprendre une poésie, comme le stipule l'élément *Wording*. Le poème contenu dans l'exercice est choisi en fonction des contraintes contenues dans *TextConstraints* : un poème en français contenant entre 100 et 200 mots.

La structure d'exercices de la Figure A-47 permet de créer des exercices comme celui de l'Exemple A-49 de l'Annexe D, page 303. Il s'agit d'un QCM où chaque énoncé de question correspond à l'image d'un animal choisie en fonction de mots clefs, *PictureChoice*. La structure d'exercices contient quatre questions alors que les QCM créés n'auront que trois questions, *QuestionsNumber*. Seule la première question est une question prioritaire, *Prioritary*, elle sera donc présente dans tous les QCM créés. L'ordre des questions doit être le même que celui de la structure d'exercices, *Random*. Les questions ont toutes la même structure, et pour chacune d'elles, quatre éléments de réponse sont donnés. Un élément de réponse correspond à un mot prédéfini, *WordChoice*. Sur les quatre réponses, une seule est juste, *Value*. Dans les QCM créés, il n'y aura que trois éléments de réponse, *TotalAnswersNumber*, dont un sera forcément la réponse juste, *TrueAnswersNumber*. Les éléments de réponse sont présentés dans un ordre aléatoire, *Random*.

```

<ExercisesStructure>
  <Metadata>...</Metadata>
  - <QuestionsAnswers>
    <Wording>Dans ce quiz, trouver la bonne orthographe :</Wording>
  - <Questions QuestionsNumber="3" Random="false">
    - <Question Priority="true">
      - <Wording>
        - <PictureChoice>
          - <PictureConstraints Quantity="1">
            <Keyword>abeille</Keyword>
          </PictureConstraints>
        </PictureChoice>
      </Wording>
    - <Answers TotalAnswersNumber="3" TrueAnswersNumber="1" Random="true">
      - <Answer Value="true">
        - <WordChoice>
          <FileIdentifier>Une abeille</FileIdentifier>
        </WordChoice>
      </Answer>
      - <Answer Value="false">
        - <WordChoice>
          <FileIdentifier>Une abeye</FileIdentifier>
        </WordChoice>
      </Answer>
      - <Answer Value="false">
        - <WordChoice>
          <FileIdentifier>Une abaille</FileIdentifier>
        </WordChoice>
      </Answer>
      - <Answer Value="false" Priority="true">
        - <WordChoice>
          <FileIdentifier>Une abbeille</FileIdentifier>
        </WordChoice>
      </Answer>
    </Answers>
  </Question>
  + <Question>
  + <Question>
  + <Question>
</Questions>
</QuestionsAnswers>
</ExercisesStructure>

```

Figure A-47 : Structure d'exercices issue du patron « Questions / réponses ».

ANNEXE G. LES GÉNÉRATEURS D'EXERCICES DE L'APPROCHE GEPPETO_P

Dans la section 6.3.3.d, nous avons présenté le générateur d'exercices correspondant à la catégorie « Travail sur texte » de notre typologie d'activités papier (cf. Figure 6-2, page 120). Dans cette annexe, nous présentons les autres générateurs correspondant à cette typologie, excepté le générateur « Démonstration » pour lequel nous n'avons pas instancié l'architecture générique.

GÉNÉRATEUR « TRAVAIL SUR ILLUSTRATION »

Ce générateur doit permettre de fournir des exercices demandant d'annoter ou de recopier une illustration. Ces exercices portent sur une ou plusieurs illustrations qui sont choisies semi-automatiquement en respectant les contraintes définies par l'utilisateur du générateur. Les actions à faire sur les illustrations sont diverses : placer des mots fournis, compléter l'illustration sans aucun indice, recopier un modèle. Nous présentons maintenant les différentes bases de connaissances nécessaires à ce générateur, ainsi que les différentes étapes du processus de génération.

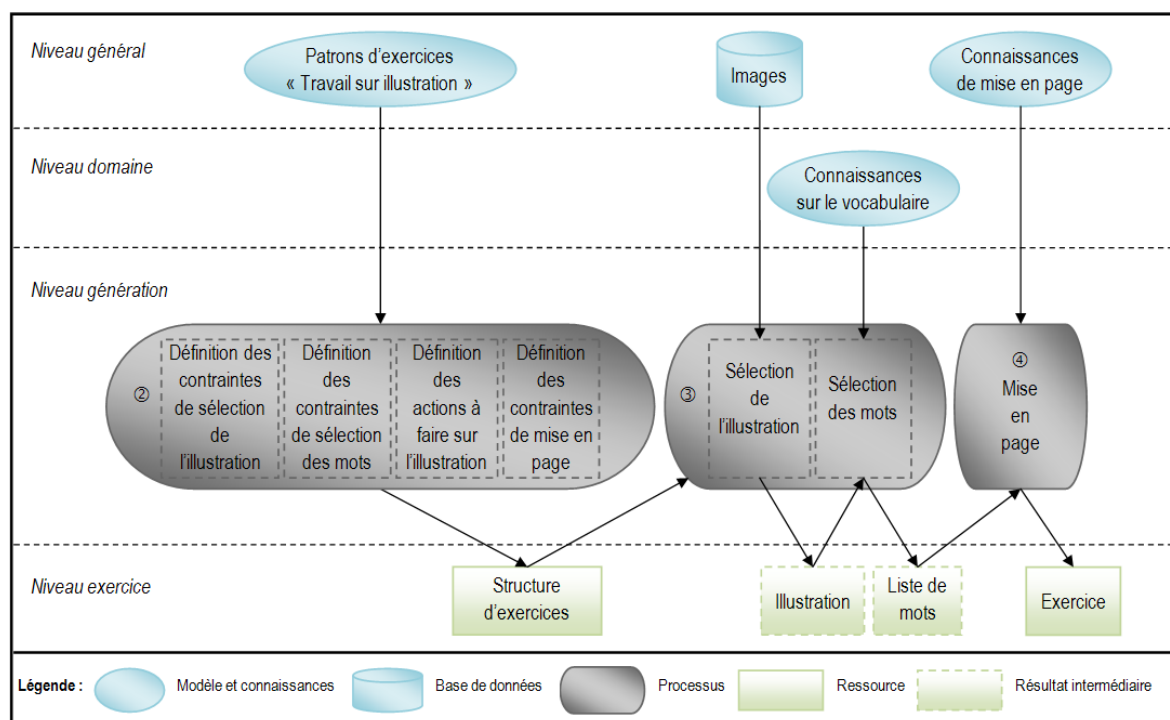


Figure A-48 : Architecture du générateur « Travail sur illustration ».

NIVEAU GÉNÉRAL : LES BASES DE CONNAISSANCES INDÉPENDANTES DU DOMAINE

Patron d'exercices. Il s'agit du patron d'exercices « Travail sur illustration » présenté dans l'Annexe F page 307.

Images. La base de données des illustrations est organisée, comme celle des textes, grâce à des métadonnées (auteurs, date de création, taille, mots clefs).

Connaissances de mise en page. Ces connaissances permettent de présenter un exercice en fournissant l'énoncé à l'apprenant et la correction à l'utilisateur du générateur.

NIVEAU DOMAINE : LES BASES DE CONNAISSANCES DÉPENDANTES DU DOMAINE

Connaissances sur le vocabulaire. Ces connaissances sont les mêmes que celles du générateur « Travail sur texte ». Elles contiennent la description technique et sémantique des mots.

L'ÉTAPE 2 DU PROCESSUS DE GÉNÉRATION : LA DÉFINITION DES CONTRAINTES

Définition des contraintes de sélection de l'illustration. La sélection d'une illustration peut se faire de trois manières : l'importation de l'illustration, la sélection dans la base d'illustrations ou la définition de contraintes de sélection dans la base d'illustrations. Les contraintes peuvent porter sur chacune des métadonnées permettant de décrire une illustration. Dans le cas de l'importation ou de la sélection dans la base d'illustrations, l'utilisateur peut modifier l'illustration pour ajouter des champs à remplir, positionner des flèches, etc. À ces contraintes de sélection d'illustrations, il faut ajouter le nombre d'illustrations à fournir à l'apprenant dans l'exercice.

Définition de contraintes de sélection des mots. Pour les exercices d'annotation, l'utilisateur indique comment choisir les mots à placer sur l'illustration. Dans le cas de l'importation ou de la sélection précise d'une illustration dans la base, il peut indiquer quels mots précisément seront à placer. Mais l'utilisateur peut aussi demander au générateur de sélectionner automatiquement des mots selon des contraintes portant sur les métadonnées décrivant les mots dans les connaissances sur le vocabulaire. Dans le cas de la sélection automatique d'une illustration, les mots seront choisis automatiquement en fonction de l'illustration et des mots clefs définissant cette illustration. Dans tous les cas, l'utilisateur doit indiquer le nombre de mots que l'apprenant devra placer sur l'illustration.

Définition des actions à faire sur l'illustration. L'utilisateur choisit s'il veut créer un exercice d'annotation ou de recopie.

Définition des contraintes de mise en page. Ces contraintes dépendent du type d'exercices à créer. Pour les exercices d'annotation, il faut indiquer si les mots à placer sont fournis à l'apprenant et sous quelle forme (tous les mots ou pas, dans quel ordre, etc.). Pour les exercices de recopie, il faut indiquer combien de fois doit être recopié le modèle et éventuellement sur quel type de ligne (ligne droite, partition musicale, etc.) et dans quelle police.

L'ÉTAPE 3 DU PROCESSUS DE GÉNÉRATION : LA GÉNÉRATION DE L'EXERCICE ET DE SA SOLUTION

Sélection de l'illustration. Le choix peut se faire aléatoirement parmi les illustrations fournies par l'utilisateur ou dans la base d'illustrations en respectant les contraintes de sélection contenues dans la structure d'exercices. Il y a autant de choix à faire que l'exercice contiendra d'illustrations.

Sélection des mots. Le système choisit les mots à placer sur une illustration ou à recopier dans la liste fournie par l'utilisateur ou dans la base de mots selon les contraintes contenues dans la structure d'exercices.

L'ÉTAPE 4 DU PROCESSUS DE GÉNÉRATION : LA MISE EN PAGE DE L'EXERCICE

Mise en page. Cette étape dépend de l'exercice que l'on souhaite créer et correspond à la mise en œuvre des contraintes de mise en page. Le système affiche la consigne de l'exercice, l'illustration et si besoin la liste des mots à placer ou les lignes permettant à l'apprenant de

compléter des lignes d'écriture. Le système fournit également la correction pour l'utilisateur du générateur.

Lors de la création d'un exercice, toutes les étapes du processus de génération ne sont pas forcément nécessaires, cela dépend du type d'exercices à créer. Ainsi, l'étape de sélection des mots est utile pour les exercices d'annotation ou de recopie de mots mais pas pour les autres exercices de recopie (par exemple, recopier une clef de sol).

Des exemples de structures d'exercices et d'exercices issus de ces structures ont été fournis lors de la définition du patron d'exercices « Travail sur illustration » (cf. Annexe F, page 307).

GÉNÉRATEUR « ORGANISATION D'ÉLÉMENTS »

Ce générateur doit permettre de fournir des exercices de classement ou de groupement d'objets. Ces exercices portent sur un ou plusieurs objets qui sont choisis semi-automatiquement en respectant les contraintes définies par l'utilisateur du générateur. Les objets peuvent être des textes, des mots, des lettres, des nombres ou des illustrations. Nous présentons ici les différentes bases de connaissances nécessaires à ce générateur, ainsi que les différentes étapes du processus de génération.

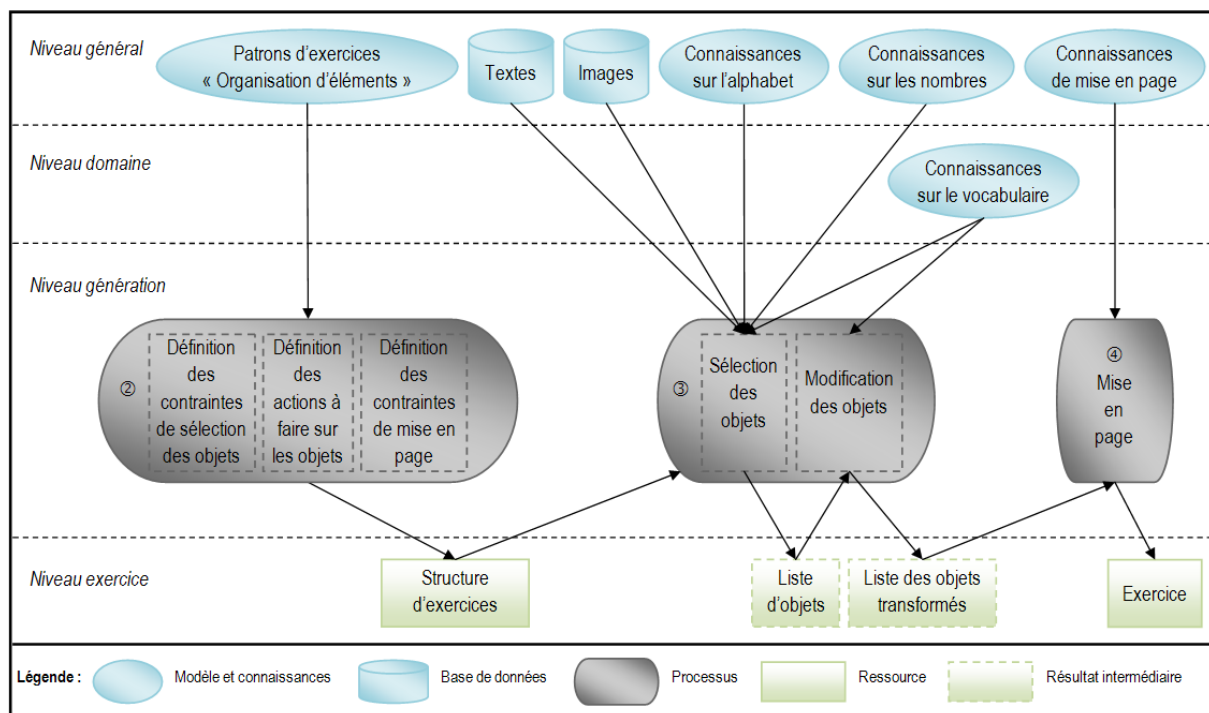


Figure A-49 : Architecture du générateur « Organisation d'éléments ».

NIVEAU GÉNÉRAL : LES BASES DE CONNAISSANCES INDÉPENDANTES DU DOMAINE

Patron d'exercices. Il s'agit du patron d'exercices « Organisation d'éléments » présenté dans l'Annexe F, page 307.

Images et Textes. Les bases de données des textes et des illustrations sont indexées grâce à des métadonnées que nous avons décrites lors de la présentation des générateurs précédents.

Connaissances sur l'alphabet. Ces connaissances permettent de comparer les lettres de l'alphabet.

Connaissances sur les nombres. Ces connaissances permettent de comparer les nombres.

Connaissances de mise en page. Ces connaissances permettent de présenter un exercice en fournissant l'énoncé à l'apprenant et la correction à l'utilisateur du générateur.

NIVEAU DOMAINE : LES BASES DE CONNAISSANCES DÉPENDANTES DU DOMAINE

Connaissances sur le vocabulaire. Ces connaissances sont les mêmes que celles du générateur « Travail sur texte ». Elles contiennent la description technique et sémantique des mots.

L'ÉTAPE 2 DU PROCESSUS DE GÉNÉRATION : LA DÉFINITION DES CONTRAINTES

Définition des contraintes de sélection de l'objet. La sélection d'un objet peut se faire de quatre manières : la saisie, l'importation, la sélection dans la base ou la définition de contraintes de sélection dans la base. Selon le type d'objets (texte, mot, lettre, nombre ou illustration), toutes les sélections d'objets ne sont pas possibles. Par exemple, une illustration ne pourra être saisie, un nombre ne pourra être importé, etc. Les contraintes de sélection automatique dans les bases de données portent sur les métadonnées permettant de décrire les objets. De la même manière, les métadonnées ne sont pas les mêmes pour tous les types d'objets. Un texte sera choisi en fonction de son auteur, son titre, l'année de sa parution, son genre, la langue dans laquelle il est écrit et le nombre de mots alors qu'un nombre sera choisi en fonction de l'intervalle auquel il appartient et de son type (entier, relatif, décimal, fraction). Pour les illustrations, il est possible de définir une illustration et d'indiquer comment la dupliquer, en changeant par exemple sa taille ou son orientation. Pour les mots, les lettres et les nombres, il est également possible de n'en choisir qu'un et d'indiquer comment les dupliquer (changement de la police, de la casse, de la taille d'écriture, de la couleur, etc.). À ces contraintes de sélection d'objets, il faut ajouter le nombre d'objets à fournir à l'apprenant dans l'exercice.

Définition des actions à faire sur les objets. L'utilisateur choisit s'il veut créer un exercice de classement ou de groupement et indique le ou les critères permettant ce tri (classement par ordre alphabétique, par ordre croissant, groupement par thèmes, découpage du texte à certains endroits et lesquels, etc.).

Définition des contraintes de mise en page. Ces contraintes indiquent dans quel ordre fournir les objets et selon quelle disposition, par exemple à la suite les uns des autres ou en colonne.

L'ÉTAPE 3 DU PROCESSUS DE GÉNÉRATION : LA GÉNÉRATION DE L'EXERCICE ET DE SA SOLUTION

Sélection de l'objet. Le choix d'un objet peut se faire aléatoirement parmi ceux fournis par l'utilisateur ou dans les bases de données en respectant les contraintes de sélection. Il y a autant de choix à faire que l'exercice contiendra d'objets.

Modification des objets. Si l'exercice porte sur des textes, le système découpe les textes selon les contraintes spécifiées ou aux endroits indiqués dans les textes fournis, pour atteindre le nombre de parties de texte voulues. Si l'exercice porte sur des images à dupliquer, le système modifie celles-ci selon les contraintes de l'utilisateur. Si l'exercice porte sur des mots, des lettres ou des nombres à dupliquer, il les duplique et modifie leur apparence (couleur, police, etc.) selon les contraintes de l'utilisateur.

L'ÉTAPE 4 DU PROCESSUS DE GÉNÉRATION : LA MISE EN PAGE DE L'EXERCICE

Mise en page. Le système mélange les objets choisis avant de proposer l'exercice pour l'apprenant et la correction pour l'utilisateur du générateur.

Des exemples de structures d'exercices et d'exercices issus de ces structures ont été fournis lors de la définition du patron d'exercices « Organisation d'éléments » (cf. Annexe F, page 307).

GÉNÉRATEUR « TABLEAU À DOUBLE ENTRÉE »

Ce générateur doit permettre de fournir des exercices portant sur des tables de conjugaison, de mathématiques ou d'autres tableaux à double entrée. Le contenu des tableaux est choisi semi-automatiquement en respectant les contraintes définies par l'utilisateur du générateur. Nous présentons maintenant les différentes bases de connaissances nécessaires à ce générateur, ainsi que les différentes étapes du processus de génération.

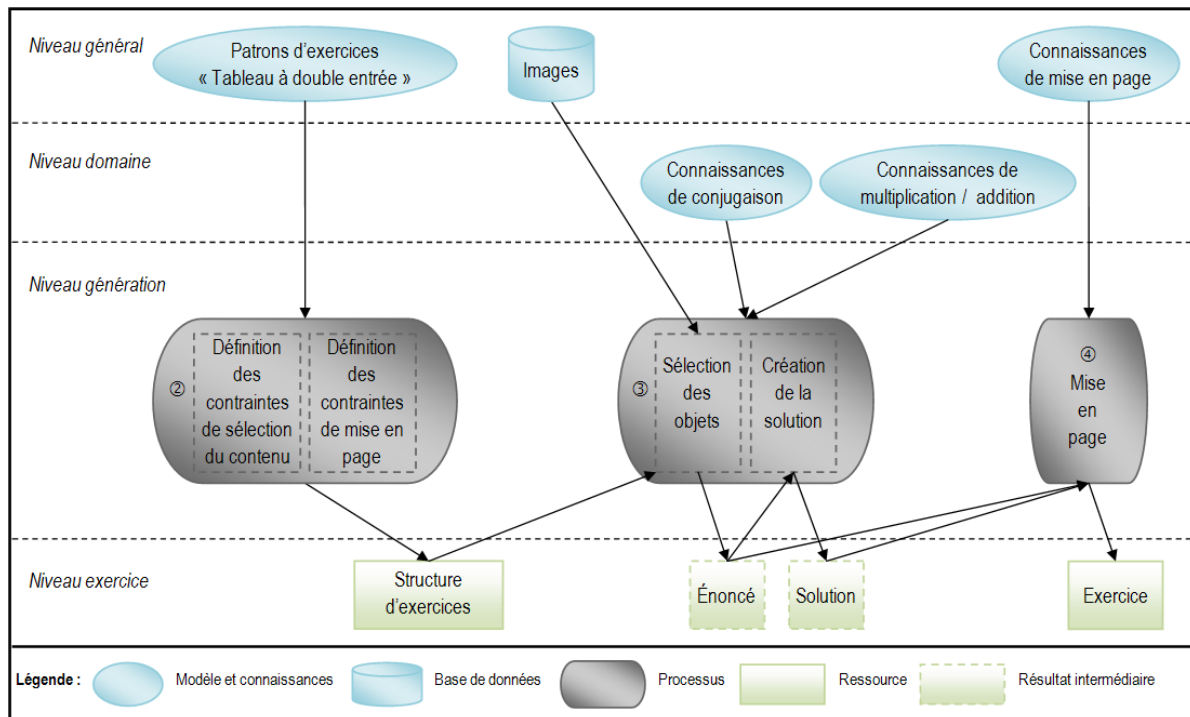


Figure A-50 : Architecture du générateur « Tableau à double entrée ».

NIVEAU GÉNÉRAL : LES BASES DE CONNAISSANCES INDÉPENDANTES DU DOMAINE

Patron d'exercices. Il s'agit du patron d'exercices « Tableau à double entrée » présenté dans l'Annexe F, page 307.

Connaissances de mise en page. Ces connaissances permettent de présenter un exercice en fournissant l'énoncé à l'apprenant et la correction à l'utilisateur du générateur.

NIVEAU DOMAINE : LES BASES DE CONNAISSANCES DÉPENDANTES DU DOMAINE

Connaissances de conjugaison. Cette base de connaissances est propre à la langue choisie pour faire travailler l'apprenant. Elle contient les verbes avec leur conjugaison à tous les temps et tous les modes.

Connaissances de multiplication / addition. Communes à toutes les langues, ces bases de connaissances contiennent les règles de multiplication / addition.

L'ÉTAPE 2 DU PROCESSUS DE GÉNÉRATION : LA DÉFINITION DES CONTRAINTES

Définition des contraintes de sélection du contenu. L'utilisateur choisit le type d'exercices et les contraintes de génération de tables (choix du temps et des personnes pour les tables de

conjugaison, de l'intervalle des nombres pour les tables de multiplication ou d'addition, le contenu des colonnes, lignes et cellule pour les tableaux à double entrée). Il peut également préciser des objets (verbes, nombres, etc.) prioritaires, c'est-à-dire des objets qu'il souhaite avoir en priorité dans les exercices qui seront générés. Il définit enfin le nombre de tableaux (verbes, nombres, tableaux à double entrée) et les cellules du ou des tableaux à demander à l'apprenant.

Définition des contraintes de mise en page. Ces contraintes indiquent si le but de l'exercice est de remplir les tables (conjugaison des verbes, résultat des calculs mathématiques, résultat de l'application du contenu d'une colonne sur une ligne, par exemple application d'une couleur sur une forme) ou d'identifier l'intitulé d'une ligne, d'une colonne ou du tableau (par exemple, identifier le verbe conjugué ou la fonction dont les valeurs sont fournies dans les cellules).

L'ÉTAPE 3 DU PROCESSUS DE GÉNÉRATION : LA GÉNÉRATION DE L'EXERCICE ET DE SA SOLUTION

Sélection de l'objet. Le système sélectionne le nombre total d'objets (verbes, nombres, cellules, tableaux) à mettre dans la liste d'objets de l'exercice, ainsi que le nombre d'objets à prendre dans la liste des objets prioritaires. Trois cas de figure se posent alors. Si ces deux nombres sont égaux, la liste d'objets de l'exercice correspond à la liste des objets prioritaires. Si l'on a plus d'objets prioritaires que d'objets à sélectionner, le système effectue un choix aléatoire parmi les objets prioritaires. Sinon, dans le cas où l'on doit sélectionner plus d'objets que le nombre d'objets prioritaires, le système prend ces objets prioritaires et les complète avec des objets choisis aléatoirement selon les autres contraintes contenues dans la structure d'exercices. Ces contraintes dépendent du domaine d'application.

Création de la solution. Selon la liste d'objets déterminée par le système, celui-ci cherche les tables correspondantes dans la base de connaissances propre aux objets.

L'ÉTAPE 4 DU PROCESSUS DE GÉNÉRATION : LA MISE EN PAGE DE L'EXERCICE

Mise en page. Le système présente la consigne de l'exercice suivie, soit de la liste des objets (verbes ou nombres) dont il faut fournir les tables, soit du ou des tableaux à remplir ou à utiliser. Le système fournit également la correction pour l'utilisateur du générateur.

Des exemples de structures d'exercices et d'exercices issus de ces structures ont été fournis lors de la définition du patron d'exercices « Tableau à double entrée » (cf. Annexe F, page 307).

GÉNÉRATEUR « EXPRESSION ALGÈBRE »

Ce générateur doit permettre de fournir des exercices portant sur des expressions algébriques. Pour pouvoir fournir la solution des exercices créés, le générateur utilise le résolveur de problèmes SYRCLAD [Guin 1997]. Nous présentons ici les différentes bases de connaissances nécessaires à ce générateur, ainsi que les différentes étapes du processus de génération.

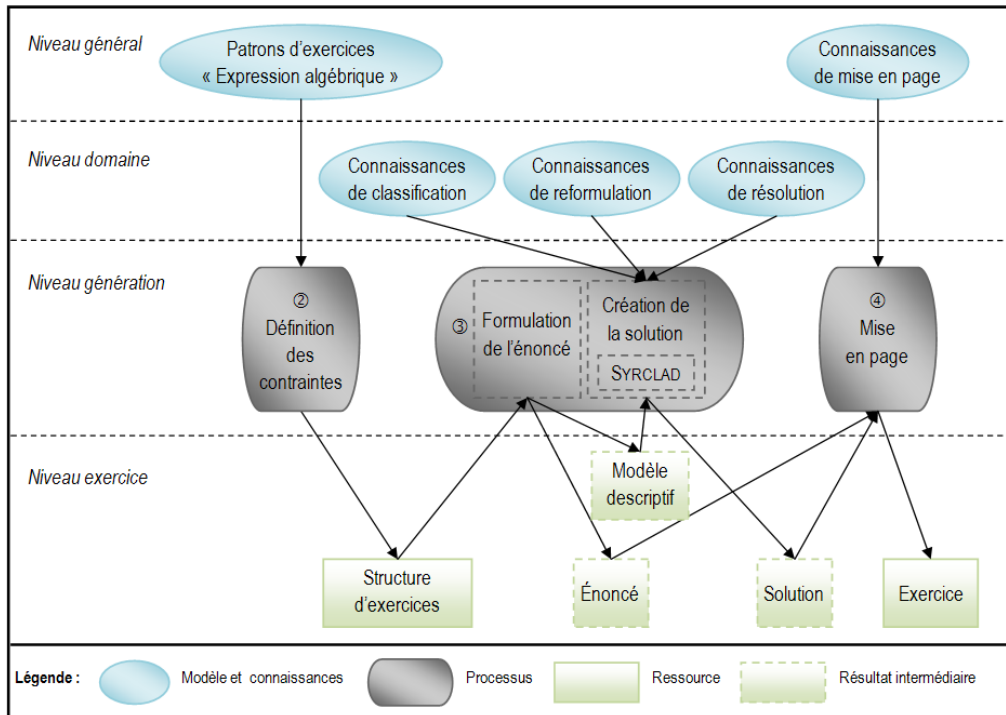


Figure A-51 : Architecture du générateur « Expression algébrique ».

NIVEAU GÉNÉRAL : LES BASES DE CONNAISSANCES INDÉPENDANTES DU DOMAINE

Patron d'exercices. Il s'agit du patron d'exercices « Expression algébrique » présenté dans l'Annexe F, page 307.

Connaissances de mise en page. Ces connaissances permettent de présenter un exercice en fournissant l'énoncé à l'apprenant et la correction à l'utilisateur du générateur.

NIVEAU DOMAINE : LES BASES DE CONNAISSANCES DÉPENDANTES DU DOMAINE

Connaissances de classification. Pour que SYRCLAD puisse résoudre des problèmes portant sur des expressions algébriques, un expert doit fournir à SYRCLAD et donc par extension au générateur, un graphe de classification des expressions. Pour cela, l'expert peut s'inspirer de la hiérarchie de patrons d'expressions algébriques mise en place dans le système APLUSIX [Bouhineau et al. 2005b]. Cette hiérarchie de patrons d'exercices permet de décrire six types d'expressions : calculer, développer, factoriser, résoudre une équation, résoudre une inéquation, résoudre un système. Pour chacun de ces types, il existe plusieurs classes de problèmes selon les méthodes de résolution qui leur sont associées.

Connaissances de reformulation. Les attributs discriminants intervenant dans le graphe de classification sont des attributs relevant de la théorie du domaine. La plupart du temps, ces attributs ne sont pas présents dans les modèles descriptifs des problèmes. Il est donc nécessaire d'avoir des connaissances permettant de déterminer la valeur de ces attributs. Ces connaissances permettent alors de passer d'un modèle descriptif à un modèle opérationnel, plus adapté au choix d'une méthode de résolution dans SYRCLAD. Pour que SYRCLAD puisse utiliser ces connaissances de reformulation, l'expert doit les exprimer par des règles « si-alors ».

Connaissances de résolution. Pour choisir la méthode de résolution à employer, il faut savoir quelle méthode est adaptée pour un problème d'une classe donnée.

L'ÉTAPE 2 DU PROCESSUS DE GÉNÉRATION : LA DÉFINITION DES CONTRAINTES

Définition des contraintes. Selon la classe de l'expression qu'il a choisie, l'utilisateur va préciser les contraintes de génération. Ces contraintes peuvent porter sur les nombres utilisés (entiers, décimaux, fractions, radicaux), sur leur signe, leur intervalle de valeur... Mais elles peuvent aussi porter sur les règles utilisées pour construire l'expression : les opérateurs utilisés, le nombre de termes, l'utilisation ou non de la retenue pour les additions/soustractions, l'utilisation ou non de parenthèses, le degré des équations, le nombre d'inconnues dans les systèmes ou les équations... Ces contraintes vont permettre de construire l'expression algébrique.

L'ÉTAPE 3 DU PROCESSUS DE GÉNÉRATION : LA GÉNÉRATION DE L'EXERCICE ET DE SA SOLUTION

Formulation de l'énoncé. Les nombres contenus dans l'expression sont choisis en respectant les contraintes définies par l'utilisateur. Le générateur produit également le modèle descriptif de l'expression pour le résolveur SYRCLAD.

Création de la solution. Le générateur utilise le résolveur de problèmes SYRCLAD pour trouver une solution à l'expression algébrique créée.

L'ÉTAPE 4 DU PROCESSUS DE GÉNÉRATION : LA MISE EN PAGE DE L'EXERCICE

Mise en page. Le système propose l'expression algébrique à l'apprenant et la solution fournie par SYRCLAD à l'utilisateur du générateur.

GÉNÉRATEUR « PROBLÈME SCIENTIFIQUE »

Le générateur de problèmes scientifiques a pour but de générer des problèmes mais également leur solution. Pour cela, nous utilisons le résolveur de problèmes SYRCLAD [Guin 1997]. Ce résolveur permet de résoudre les problèmes d'un domaine uniquement si une classification des problèmes dans ce domaine peut être définie par les experts du domaine. En acceptant cette limite dans la définition des exercices de type « Problème scientifique » (cf. F sur la Figure 6-2), nous pouvons dire qu'un exercice sera constitué d'un énoncé et de sa correction.

Le générateur de problèmes scientifiques est un générateur qui inclut le générateur GENAMBRE [Duclosson et al. 2005a, Jean-Daubias et al. 2009b] en lui ajoutant une interface générique pour que l'utilisateur puisse définir ses contraintes de génération d'exercices. Il utilise, comme nous venons de le voir, le résolveur de problèmes SYRCLAD [Guin 1997] pour avoir la solution du problème proposé par GENAMBRE. Enfin, l'étape de mise en page permet d'avoir un exercice ayant une présentation homogène avec les exercices créés par les autres générateurs suivant l'architecture générique des générateurs d'activités papier de l'approche GEPPETO_P. Nous présentons maintenant les différentes bases de connaissances nécessaires à ce générateur, ainsi que les différentes étapes du processus de génération.

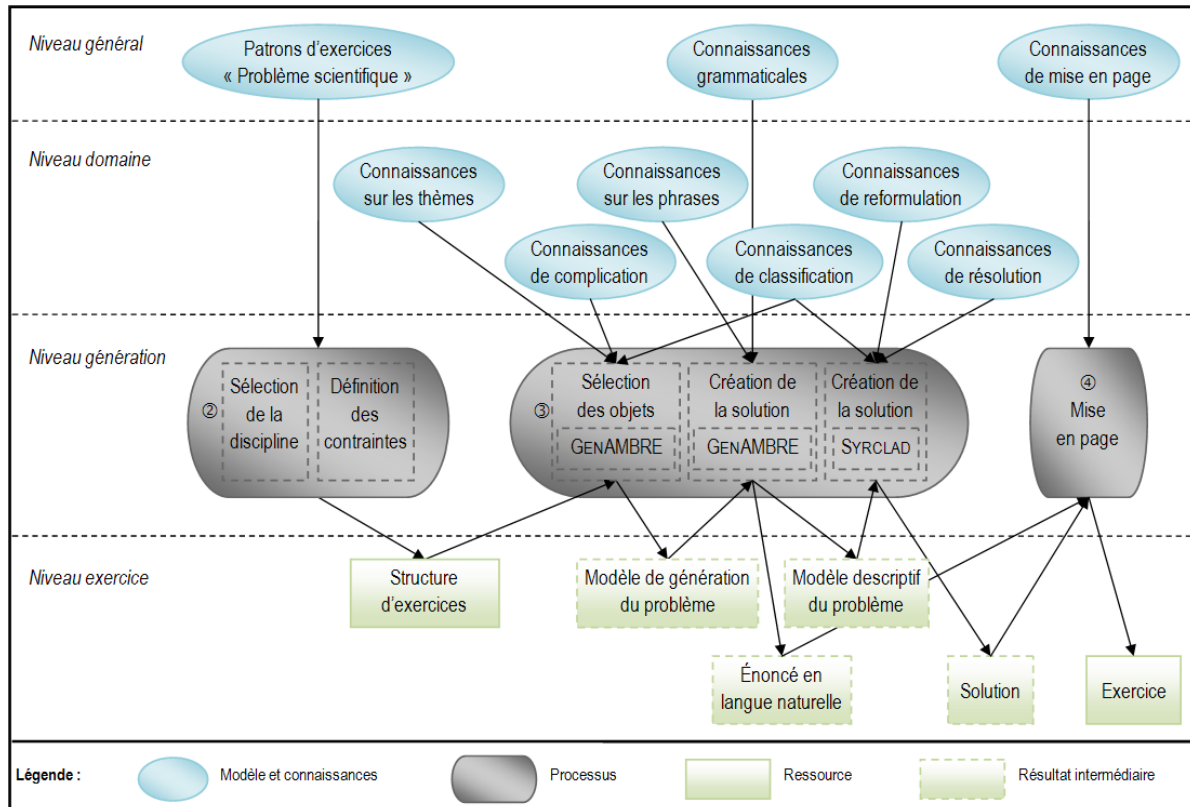


Figure A-52 : Architecture du générateur « Problème scientifique ».

NIVEAU GÉNÉRAL : LES BASES DE CONNAISSANCES INDÉPENDANTES DU DOMAINE

Patron d'exercices. Il s'agit du patron d'exercices « Problème scientifique » présenté dans l'Annexe F, page 307.

Connaissances grammaticales. L'expert fournit une grammaire au système de génération, c'est-à-dire un ensemble de structures de phrases qui pourront être utilisées. Les unités syntaxiques de base sont les groupes sujets, les verbes et les compléments.

Connaissances de mise en page. Ces connaissances permettent de présenter un exercice en fournissant l'énoncé à l'apprenant et la correction à l'utilisateur du générateur.

NIVEAU DOMAINE : LES BASES DE CONNAISSANCES DÉPENDANTES DU DOMAINE

Connaissances sur les thèmes. Pour pouvoir générer un problème, il est nécessaire de connaître le thème concerné et les traits de surface qui lui sont associés (par exemple les objets, les personnages, les actions). De telles connaissances sont dépendantes du domaine pour lequel elles sont utilisées, mais leur représentation est indépendante du domaine. Les connaissances sur les thèmes sont fournies par l'expert.

Connaissances de complication. Pour pouvoir compliquer l'énoncé de l'exercice, il est nécessaire de fournir des connaissances de complication au système de génération. Ces connaissances vont permettre, par exemple, de changer l'ordre des phrases ou d'ajouter des phrases inutiles.

Connaissances sur les phrases. Les énoncés générés, notamment la structure de leurs phrases, dépendent de la classe à laquelle le problème appartient. Il est donc nécessaire, pour pouvoir générer l'énoncé en langue naturelle, de savoir quelles structures de phrases contenues dans les connaissances grammaticales pourront être utilisées pour le problème à générer. Ce

sont ces informations que contiennent les connaissances sur les phrases. Elles permettent d'associer à la classe du problème les structures de phrases utilisables et les éléments du problème qui seront associés à ces structures.

Connaissances de classification. Dans chaque domaine d'application, un expert fournit au résolveur SYRCLAD, et donc par extension au générateur GENAMBRE, un graphe de classification des problèmes. Ce graphe est utilisé dans le résolveur pour classer le problème. C'est une hiérarchie de classes dépendante du domaine, mais dont la représentation est la même quel que soit le domaine.

Connaissances de reformulation. Comme pour les expressions algébriques, les attributs discriminants intervenant dans le graphe de classification sont des attributs relevant de la théorie du domaine. La plupart du temps, ces attributs ne sont pas présents dans les modèles descriptifs des problèmes. Il est donc nécessaire d'avoir des connaissances permettant de déterminer la valeur de ces attributs. Ces connaissances permettent alors de passer d'un modèle descriptif à un modèle opérationnel, plus adapté au choix d'une méthode de résolution dans SYRCLAD. Pour que SYRCLAD puisse utiliser ces connaissances de reformulation, l'expert les exprime par des règles « si-alors ».

Connaissances de résolution. Comme pour les expressions algébriques, pour choisir la méthode de résolution à employer, il faut savoir quelle méthode est adaptée pour un problème d'une classe donnée.

L'ÉTAPE 2 DU PROCESSUS DE GÉNÉRATION : LA DÉFINITION DES CONTRAINTES

Sélection de la discipline. L'utilisateur choisit, en fonction des bases de connaissances disponibles, la discipline (problème additif, thermodynamique, etc.) pour laquelle il souhaite générer un exercice. En effet, GENAMBRE et SYRCLAD sont respectivement un générateur et un résolveur de problèmes pouvant être utilisés dans plusieurs domaines, mais il faut pour cela qu'un expert fournisse les connaissances propres à chaque domaine.

Définition des contraintes. L'utilisateur définit les contraintes de génération du problème (intervalle des nombres utilisés, ajout de phrases inutiles, etc.) utiles à GENAMBRE pour créer une structure d'exercices.

L'ÉTAPE 3 DU PROCESSUS DE GÉNÉRATION : LA GÉNÉRATION DE L'EXERCICE ET DE SA SOLUTION

Génération du modèle du problème. À partir des connaissances sur les thèmes, sur les phrases, des connaissances de complication et des contraintes entrées par l'utilisateur, GENAMBRE génère le modèle de génération du problème. Ce modèle est en fait un modèle descriptif étendu, puisqu'il contient le modèle descriptif du problème, mais précise également quelques informations absentes du modèle descriptif, telles que la classe du problème ou encore son thème. Pour construire ce modèle, le générateur complète les contraintes définies par l'utilisateur et fournit à GENAMBRE, en choisissant notamment aléatoirement des valeurs pour celles qui n'ont pas été précisées.

Génération de l'énoncé du problème. Pour générer un énoncé en langue naturelle, GENAMBRE utilise les connaissances sur les phrases et la grammaire du domaine, ainsi que le modèle de génération du problème créé auparavant. Les connaissances sur les phrases permettent à GENAMBRE de prendre les décisions d'ordre conceptuel (« décider quoi dire »). Le système spécifie ainsi la structure des phrases qui seront générées en précisant pour chaque phrase les éléments qui la composeront (issus du modèle du problème : objets, personnages...). Une fois les décisions d'ordre conceptuel prises, GENAMBRE passe à l'étape de génération de

texte, et met en place différents mécanismes linguistiques : traitements syntaxiques (« décider comment le dire »), lexicaux et morphologiques (« décider comment l'écrire »).

Création de la solution. Le générateur utilise le résolveur de problèmes SYRCLAD pour trouver une solution au problème créé.

L'ÉTAPE 4 DU PROCESSUS DE GÉNÉRATION : LA MISE EN PAGE DE L'EXERCICE

Mise en page. Le système uniformise la présentation des exercices et fournit l'énoncé du problème généré par GENAMBRE à l'apprenant et la solution générée par SYRCLAD à l'utilisateur.

GÉNÉRATEUR « QUESTIONS / RÉPONSES »

Ce générateur permet de fournir une liste de questions portant sur aucun, un ou plusieurs documents (textes et/ou images). Les questions sont soit fermées et l'on peut fournir la correction, soit ouverte et la correction ne sera fournie que lorsque l'utilisateur l'aura définie. Ce générateur va permettre de créer des QCM (questions à choix multiples), des QROC (questions à réponses ouvertes courtes), des études de documents, des sujets de dissertation, etc. Nous présentons ici les différentes bases de connaissances nécessaires à ce générateur, ainsi que les différentes étapes du processus de génération.

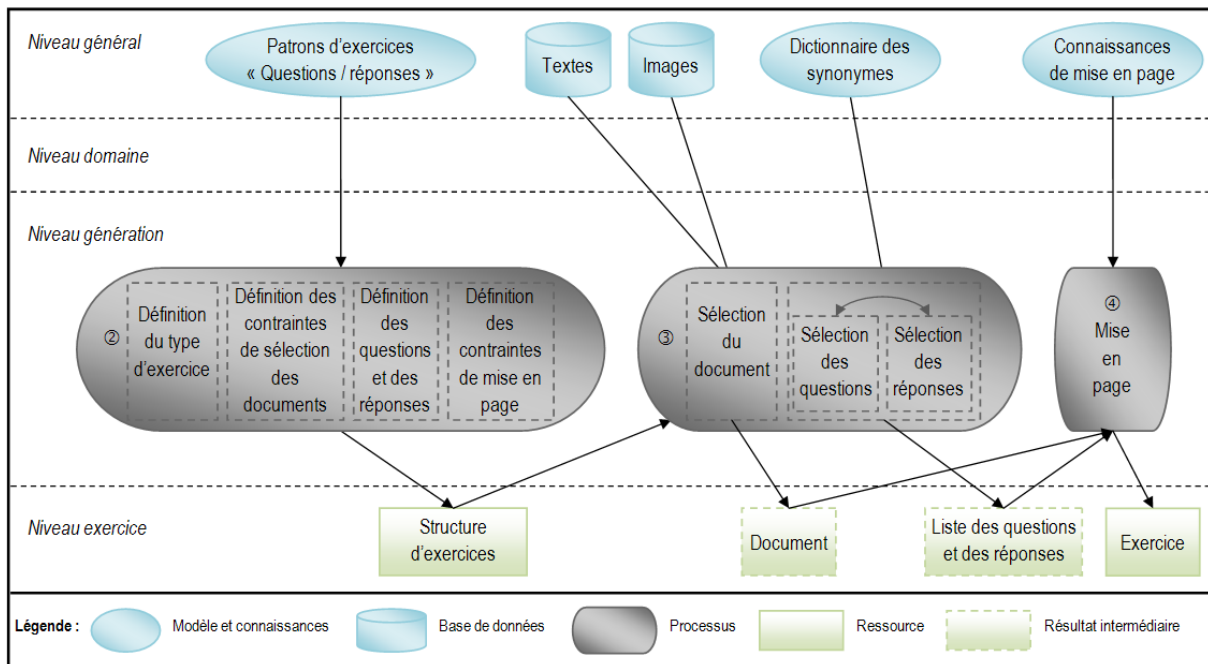


Figure A-53 : Architecture du générateur « Questions / réponses ».

NIVEAU GÉNÉRAL : LES BASES DE CONNAISSANCES INDÉPENDANTES DU DOMAINE

Patron d'exercices. Il s'agit du patron d'exercices « Questions / réponses » présenté dans l'Annexe F, page 307.

Images et Textes. Les bases de données des textes et des illustrations sont organisées grâce à des métadonnées que nous avons fournies lors de la présentation des générateurs précédents.

Dictionnaire des synonymes. Ce dictionnaire permet de faire varier les termes d'un énoncé pour augmenter la diversité des énoncés générés.

Connaissances de mise en page. Ces connaissances permettent de présenter un exercice en fournissant l'énoncé à l'apprenant et la correction à l'utilisateur du générateur.

NIVEAU DOMAINE : LES BASES DE CONNAISSANCES DÉPENDANTES DU DOMAINE

Ce générateur ne possède pas de connaissance spécifique à un domaine d'application.

L'ÉTAPE 2 DU PROCESSUS DE GÉNÉRATION : LA DÉFINITION DES CONTRAINTES

Définition du type d'exercice. L'utilisateur choisit la consigne générale et donc le type d'exercice à créer, ainsi que le nombre de questions que contiendra l'exercice.

Définition des contraintes de sélection des documents. L'utilisateur fournit des documents précis ou indique des contraintes pour les choisir automatiquement (auteur, type image/texte, langue, etc.).

Définition des questions et des réponses associées. L'utilisateur indique les questions pouvant se trouver dans l'exercice, ainsi que les réponses à ces questions. Il peut donner plus de questions que n'en contiendra l'exercice. Il doit de plus indiquer, pour chaque question, le nombre de réponses à proposer à l'apprenant en précisant les nombres X et Y de telle façon que la question présente dans l'exercice aura X réponses justes parmi Y propositions de réponses, sachant que $X \leq Y$ et que Y est au moins inférieur au nombre total de réponses définies.

Définition des contraintes de mise en page. Ces contraintes précisent la mise en page de l'exercice (une réponse par ligne, toutes les réponses sur une ligne, le document affiché avant ou après les questions...) et indique s'il faut faire varier ou non le vocabulaire des questions.

L'ÉTAPE 3 DU PROCESSUS DE GÉNÉRATION : LA GÉNÉRATION DE L'EXERCICE ET DE SA SOLUTION

Sélection des documents. Les documents peuvent être choisis parmi les documents fournis par l'utilisateur ou dans la base de textes et d'images en respectant les contraintes de sélection contenues dans la structure d'exercices.

Sélection des questions et des réponses. Le système choisit les questions aléatoirement parmi celles contenues dans la structure d'exercices. Le système choisit ensuite, pour chaque question, les propositions de réponses en respectant les contraintes données par l'utilisateur.

L'ÉTAPE 4 DU PROCESSUS DE GÉNÉRATION : LA MISE EN PAGE DE L'EXERCICE

Mise en page. Cette étape dépend de l'exercice que l'on souhaite créer. Selon les contraintes de mise en page choisies par défaut ou modifiées par l'enseignant, le système crée l'exercice avec le(s) document(s), les questions choisies par le système et lorsque cela est demandé les propositions de réponses.

Des exemples de structures d'exercices et d'exercices issus de ces structures ont été fournis lors de la définition du patron d'exercices « Questions / réponses » (cf. Annexe F, page 307).

ANNEXE H. EXEMPLE DE CONTENU DE FICHIERS DE CONFIGURATION ET IDENTIFICATION DES INVARIANTS DU MÉTA-MODÈLE AKEPI

Cette annexe présente quelques exemples permettant de déterminer comment découvrir les invariants dans les fichiers de configuration et les interfaces ayant un impact sur le paramétrage du système.

Premier exemple, l'EIAH AMBRE-ADD [Nogry et al. 2004] possède un générateur d'exercices permettant d'agir sur la création d'une **activité**. Le **fichier de configuration** (voir un extrait en Figure A-54) du générateur contient entre autre la valeur de chaque paramètre nécessaire à la création des exercices. Ce fichier de configuration est constitué de codes spécifiques permettant d'adapter l'EIAH AMBRE-ADD.

```
appel_generation([[ 'c_comparaison_dif', 'de plus'], [jeu],
[ 'bille', 'bille jaune', 'bille jaune'], [], [0,0,1,0], [5,15], [],
[intervalle,1], [0,2,0,1,0,0]], apercu) .
```

Figure A-54 : Extrait du fichier de configuration pour la génération d'exercices dans l'EIAH AMBRE-ADD.

Si l'on prend le premier *code* du fichier de configuration ['c_comparaison_dif', 'de plus'], il correspond en réalité, pour le premier champ, à la valeur d'un paramètre « classe de problème », de type *scale_list* dont les valeurs possibles (fournies explicitement à l'utilisateur dans l'environnement graphique) sont « réunion, changement, **comparaison** » concaténée à la valeur du paramètre associé « place de l'inconnue » de type *scale_list* dont les valeurs possibles sont « résultat, opérande, opérateur, **différence**, un des deux - min, un deux- max » et pour le second champ, à la valeur du paramètres « variation », également de type *scale_list* et dont les valeurs possibles sont « **augmentation**, diminution ». On voit ici que sous l'apparente spécificité du fichier de configuration, on retrouve un paramétrage finalement assez simple consistant à choisir des valeurs dans des listes prédéfinies. Nous avons mis en **gras** les valeurs des listes correspondant au code retenu dans le fichier de configuration.

De la même façon, si l'on prend le cinquième paramètre du fichier de configuration [0,0,1,0], il correspond à la valeur des paramètres « utiliser une seule sorte d'objets », « utiliser plusieurs sortes d'objets », « affecter les objets » et « affecter les personnages » tous de type *scale_boolean*. Le lecteur aura compris facilement que vrai est représenté par un 1 et faux par un 0.

Pour terminer cet exemple avec un autre type de paramètre, nous choisissons le sixième paramètre du fichier de configuration [5,15]. Il correspond à la valeur du paramètre « intervalles pour les valeurs », de type *scale_interval* et de valeurs possibles [LowerBound = 0..∞, UpperBound = 0..∞].

Second exemple, dans le logiciel APLUSIX [Nicaud et al. 2003], l'apprenant peut obtenir une **séquence d'activités** via une carte de tests qui est représentée par un tableau à double entrée. Cette carte de test est disponible depuis l'**interface** de l'EIAH. Les lignes du tableau correspondent aux types de calculs pouvant être : « calcul numérique (A), développement et résolution (B), factorisation (C), résolution d'équation (D), résolution d'inéquation (E), résolution de système (F) » et les colonnes correspondent au niveau de difficulté entre 1 et 9. Certaines cases du tableau sont inactives, comme A6, A7, A8 et A9. L'apprenant choisit ensuite une case du tableau et l'EIAH génère les exercices correspondant au type de calcul et au niveau demandé.

Cette carte de tests peut être décrite avec le paramètre « type de calcul », de type *scale_list* dont les valeurs possibles sont « A, B, C, D, E, F » et le paramètre « niveau », de type *scale_numerical* de valeurs comprises entre 1 et 9. Ces deux paramètres sont contraints par un ensemble de règles du type « Si {Valeur(type de calcul) = A} Alors {DomaineVal(niveau) = 1..5} ».

Troisième exemple, dans le logiciel ANDES [VanLehn et al. 2005], l'apprenant peut demander à faire une **activité** en choisissant dans les menus de l'**interface** le sujet parmi ceux d'une discipline puis le numéro de l'activité.

Ces menus peuvent être décrits grâce à trois paramètres. Le premier est le paramètre « discipline », de type *scale_list* dont les valeurs sont « mechanics, electricity and magnetism ». Le second est le paramètre « sujet », de type *scale_list* dont un aperçu des valeurs est « vectors, translational kinematics, free body diagrams, statics, ... ». Le troisième est le paramètre « numéro d'exercice », de type *scale_list*, dont un aperçu des valeurs est « vect1a, ..., relvec3a, ..., mirror1, mirror2, ... ». Ces trois paramètres sont contraints par un ensemble de règles du type « Si {Valeur(sujet) = vectors} Alors {DomaineVal(numéro d'exercice) = « vect1a, ..., relvec3a »} ».

Dernier exemple, dans le logiciel EXOMATIKS [Exomatiks], les **fonctionnalités** et l'**interface** du logiciel (langue, options graphiques, les outils, les utilisateurs...) sont définies à partir d'un **fichier de configuration** (voir un extrait en Figure A-55).

```
[generales]
langue=0
...
[Internet]
utiliser_ie=1
...
[outils]
executer=-1
auto_mutiplier=1
auto_reduction=0
activer_coloration=1
...
```

Figure A-55 : Extrait du fichier de configuration pour l'interface du logiciel EXOMATIKS.

Si l'on prend le code « `langue=0` », il correspond à la valeur du paramètre « langue », de type *scale_list* dont les valeurs sont « français, anglais, ... » et où la valeur du fichier indique l'index de la liste énumérée.

ANNEXE I. MODÈLE OKEP/ABALECT

Cette annexe regroupe les différentes parties du modèle OKEP/ABALECT défini pour le logiciel ABALECT [Chevé 2005] ainsi que les documents permettant sa mise en œuvre dans le logiciel Adapte [Lefevre et al. 2009a].

PROPRIÉTÉS PÉDAGOGIQUES DU MODÈLE OKEP/ABALECT AU FORMAT XSD

```
<?xml version="1.0" encoding="UTF-8"?>
<PedagogicalProperties xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="XXX\Metamodel_AKEPI.xsd">
  <PedagogicalContent>
    <TypeOfActivity ID="A001">
      <Name>Exercice</Name>
      <Parameters>
        <Parameter ID="P001">
          <Name>Niveau</Name>
          <AssociatedCompetence>Classe de l'élève</AssociatedCompetence>
          <ScaleList>
            <Variable>>false</Variable>
            <Ordered>>false</Ordered>
            <MultipleSelection>>false</MultipleSelection>
            <Value>
              <Name>CP</Name>
            </Value>
            <Value>
              <Name>CE1</Name>
            </Value>
            <Value>
              <Name>CE2</Name>
            </Value>
            <Value>
              <Name>CM1</Name>
            </Value>
            <Value>
              <Name>CM2</Name>
            </Value>
          </ScaleList>
        </Parameter>
        <Parameter ID="P002">
          <Name>Titre du texte</Name>
          <ScaleList>
            <Variable>>false</Variable>
            <Ordered>>false</Ordered>
            <MultipleSelection>>false</MultipleSelection>
            <Value>
              <Name>L'anniversaire</Name>
            </Value>
            <Value>
              <Name>L'arrivée du loup</Name>
            </Value>
            <Value>
              <Name>L'inondation</Name>
            </Value>
            <Value>
              <Name>Où est mon chat</Name>
            </Value>
            <Value>
              <Name>Une bonne pêche</Name>
            </Value>
            <Value>
              <Name>Une maladie formidable</Name>
            </Value>
          </ScaleList>
        </Parameter>
      </Parameters>
    </TypeOfActivity>
  </PedagogicalContent>
</PedagogicalProperties>
```

```
<Value>
  <Name>La vipère</Name>
</Value>
<Value>
  <Name>L'avion</Name>
</Value>
<Value>
  <Name>Le cheval venu de la mer</Name>
</Value>
<Value>
  <Name>Le hamster</Name>
</Value>
<Value>
  <Name>Nourrir les oiseaux en hiver</Name>
</Value>
<Value>
  <Name>Pirlipi, la chauve-souris</Name>
</Value>
<Value>
  <Name>Recette de la mousse</Name>
</Value>
<Value>
  <Name>Du cadran solaire à la montre atomique</Name>
</Value>
<Value>
  <Name>La fuite de Saddam Hussein</Name>
</Value>
<Value>
  <Name>La mante religieuse</Name>
</Value>
<Value>
  <Name>La prise du château fort</Name>
</Value>
<Value>
  <Name>Le chene</Name>
</Value>
<Value>
  <Name>Les fourmis ont du flair</Name>
</Value>
<Value>
  <Name>Les Gaulois</Name>
</Value>
<Value>
  <Name>Les hirondelles</Name>
</Value>
<Value>
  <Name>Un élevage de lapins angoras</Name>
</Value>
<Value>
  <Name>Un perroquet bien particulier</Name>
</Value>
<Value>
  <Name>Un vétérinaire pour reptiles</Name>
</Value>
<Value>
  <Name>Veux-tu élever un chat</Name>
</Value>
<Value>
  <Name>La sécurité du président</Name>
</Value>
<Value>
  <Name>L'aventure du basket</Name>
</Value>
<Value>
  <Name>Le tir à l'arc</Name>
</Value>
<Value>
  <Name>L'écureuil</Name>
```

```

    </Value>
    <Value>
      <Name>Les tremblements de terre</Name>
    </Value>
    <Value>
      <Name>Qui était Louis XIV</Name>
    </Value>
    <Value>
      <Name>Trois femmes agricultrice</Name>
    </Value>
    <Value>
      <Name>Un emploi du temps dans une école japonaise</Name>
    </Value>
    <Value>
      <Name>Un tournoi au Moyen Age</Name>
    </Value>
    <Value>
      <Name>Côte d'Ivoire</Name>
    </Value>
    <Value>
      <Name>Des logiciels pour apprendre le civisme</Name>
    </Value>
    <Value>
      <Name>La fonte des glaciers</Name>
    </Value>
    <Value>
      <Name>Le blaireau</Name>
    </Value>
    <Value>
      <Name>L'eau courante</Name>
    </Value>
    <Value>
      <Name>Les marées vertes de l'océan</Name>
    </Value>
    <Value>
      <Name>Les trous noirs de l'espace</Name>
    </Value>
    <Value>
      <Name>Petite histoire du vélo</Name>
    </Value>
    <Value>
      <Name>Une drôle de pêche</Name>
    </Value>
    <Value>
      <Name>Une journée en classe en 1882</Name>
    </Value>
    <Value>
      <Name>Une maison pour l'hiver</Name>
    </Value>
  </ScaleList>
</Parameter>
<Parameter ID="P003">
  <Name>Type d'exercice</Name>
  <ScaleList>
    <Variable>>false</Variable>
    <Ordered>>false</Ordered>
    <MultipleSelection>>false</MultipleSelection>
    <Value>
      <Name>Comprendre le texte</Name>
    </Value>
    <Value>
      <Name>Ordre</Name>
      <AssociatedCompetence>cohérence</AssociatedCompetence>
    </Value>
    <Value>
      <Name>Vocabulaire</Name>
    </Value>
  </ScaleList>
</Parameter>

```



```
        <Name>Mémoire / Rapidité / Destination</Name>
    </Value>
    <Value>
        <Name>Orthographe</Name>
    </Value>
</ScaleList>
</Parameter>
<Parameter ID="P004">
    <Name>Nom de l'exercice</Name>
    <ScaleList>
        <Variable>>false</Variable>
        <Ordered>>false</Ordered>
        <MultipleSelection>>true</MultipleSelection>
        <Value>
            <Name>Questionnaire 1</Name>
        </Value>
        <Value>
            <Name>Questionnaire 2</Name>
        </Value>
        <Value>
            <Name>Texte masqué</Name>
        </Value>
        <Value>
            <Name>Closures</Name>
        </Value>
        <Value>
            <Name>Mots en desordre</Name>
        </Value>
        <Value>
            <Name>Mots inversés</Name>
        </Value>
        <Value>
            <Name>Lignes en desordre</Name>
        </Value>
        <Value>
            <Name>Phrases en desordre</Name>
        </Value>
        <Value>
            <Name>Texte à trous</Name>
        </Value>
        <Value>
            <Name>Dictionnaire</Name>
        </Value>
        <Value>
            <Name>Synonymes</Name>
        </Value>
        <Value>
            <Name>Contraires</Name>
        </Value>
        <Value>
            <Name>Masculin / Féminin</Name>
        </Value>
        <Value>
            <Name>Thèmes 1</Name>
        </Value>
        <Value>
            <Name>Thèmes 2</Name>
        </Value>
        <Value>
            <Name>Photographier</Name>
        </Value>
        <Value>
            <Name>Mémoriser la série</Name>
        </Value>
        <Value>
            <Name>Mot tronqué</Name>
        </Value>
        <Value>
```

```

        <Name>Poursuite</Name>
    </Value>
    <Value>
        <Name>Trouve-moi</Name>
    </Value>
    <Value>
        <Name>L'absent</Name>
    </Value>
    <Value>
        <Name>Le pendu</Name>
    </Value>
    <Value>
        <Name>Mot coupé</Name>
    </Value>
    <Value>
        <Name>Ecrire un mot</Name>
    </Value>
    <Value>
        <Name>Je corrige</Name>
    </Value>
    <Value>
        <Name>Ecrire une phrase</Name>
    </Value>
    <Value>
        <Name>Ordre alphabétique</Name>
    </Value>
</ScaleList>
</Parameter>
</Parameters>
</TypeOfActivity>
</PedagogicalContent>
<PedagogicalOrganization>
    <Sequence>
        <TypeOfContent>
            <TypeOfActivity>A001</TypeOfActivity>
            <MinNumberOfActivities>1</MinNumberOfActivities>
            <MaxNumberOfActivities>100</MaxNumberOfActivities>
            <Duplication>>false</Duplication>
        </TypeOfContent>
    </Sequence>
    <OrderedSequence>>true</OrderedSequence>
</PedagogicalOrganization>
</PedagogicalProperties>

```

EXTRAIT DU FICHER DE CORRESPONDANCE ENTRE LES TERMES DU MODÈLE XSD ET LES TERMES DES PRÉDICATS PROLOG

```

<CorrespondanceProlog NameOfFILE="Abalect">
    <element modele="CP" prolog="cp"/>
    <element modele="CE1" prolog="ce1"/>
    <element modele="CE2" prolog="ce2"/>
    <element modele="CM1" prolog="cm1"/>
    <element modele="CM2" prolog="cm2"/>
    <element modele="L'anniversaire" prolog="l_anniversaire"/>
    <element modele="L'arrivée du loup" prolog="l_arrivée_du_loup"/>
    <element modele="L'inondation" prolog="l_inondation"/>
    <element modele="Où est mon chat" prolog="ou_est_mon_chat"/>
    <element modele="Une bonne pêche" prolog="une_bonne_peche"/>
    <element modele="Une maladie formidable" prolog="une_maladie_formidable"/>
    <element modele="La vipère" prolog="la_vipere"/>
    <element modele="L'avion" prolog="l_avion"/>
    <element modele="Le cheval venu de la mer" prolog="le_cheval_venu_de_la_mer"/>
    <element modele="Le hamster" prolog="le_hamster"/>
    <element modele="Nourrir les oiseaux en hiver"
prolog="nourrir_les_oiseaux_en_hiver"/>
    <element modele="Pirlipi, la chauve-souris" prolog="pirlipi_la_chauve_souris"/>
    <element modele="Recette de la mousse" prolog="recette_de_la_mousse"/>

```

```

    <element modele="Du cadran solaire à la montre atomique"
prolog="du_cadran_solaire_a_la_monstre_atomique"/>
    <element modele="La fuite de Saddam Hussein"
prolog="la_fuite_de_saddam_hussein"/>
    <element modele="La mante religieuse" prolog="la_mante_religieuse"/>
    <element modele="La prise du château fort" prolog="la_prise_du_chateau_fort"/>
    <element modele="Le chene" prolog="le_chene"/>
    <element modele="Les fourmis ont du flair" prolog="les_fourmis_ont_du_flair"/>
    <element modele="Les Gaulois" prolog="les_gaulois"/>
    <element modele="Les hirondelles" prolog="les_hirondelles"/>
    <element modele="Un élevage de lapins angoras"
prolog="un_elevage_de_lapins_angoras"/>
    <element modele="Un perroquet bien particulier"
prolog="un_perroquet_bien_particulier"/>
    <element modele="Un vétérinaire pour reptiles"
prolog="un_veterinaire_pour_reptiles"/>
    ...
    <element modele="Comprendre le texte" prolog="comprendre_le_texte"/>
    <element modele="Ordre" prolog="ordre"/>
    <element modele="Vocabulaire" prolog="vocabulaire"/>
    <element modele="Mémoire / Rapidité / Destination"
prolog="memoire_rapidite_destination"/>
    <element modele="Orthographe" prolog="orthographe"/>
    <element modele="Questionnaire 1" prolog="questionnaire_1"/>
    <element modele="Questionnaire 2" prolog="questionnaire_2"/>
    <element modele="Texte masqué" prolog="texte_masque"/>
    <element modele="Closures" prolog="closures"/>
    <element modele="Mots en desordre" prolog="mots_en_desordre"/>
    <element modele="Mots inversés" prolog="mots_inverses"/>
    <element modele="Lignes en desordre" prolog="lignes_en_desordre"/>
    <element modele="Phrases en desordre" prolog="phrases_en_desordre"/>
    ...
</CorrespondanceProlog>

```

PROPRIÉTÉS PÉDAGOGIQUES DU MODÈLE OKEP/ABALECT TRADUITES EN PRÉDICATS PROLOG

```

:-dynamic fait/1.

fait(propriete(p001)).
fait(accessible(p001, oui)).
fait(domaine_val(p001, scale_list, [cp, ce1, ce2, cm1, cm2])).

fait(propriete(p002)).
fait(accessible(p002, oui)).
fait(domaine_val(p002, scale_list, [l_anniversaire, l_arrivée_du_loup,
    l_inondation,ou_est_mon_chat, une_bonne_peche, une_maladie_formidable,
    la_vipere, l_avion, le_cheval_venu_de_la_mer, le_hamster,
    nourrir_les_oiseaux_en_hiver, pirlipi_la_chaue_souris,
    recette_de_la_mousse, du_cadran_solaire_a_la_monstre_atomique,
    la_fuite_de_saddam_hussein, la_mante_religieuse, la_prise_du_chateau_fort,
    le_chene, les_fourmis_ont_du_flair, les_gaulois, les_hirondelles,
    un_elevage_de_lapins_angoras, un_perroquet_bien_particulier,
    un_veterinaire_pour_reptiles, veux_tu_elever_un_chat, la_securite_du_president,
    l_aventure_du_basket, le_tir_a_l_arc, l_ecureuil, les_tremblements_de_terre,
    qui_etait_louis_xiv, trois_femmes_agricultrice,
    un_emploi_du_temps_dans_une_ecole_japonaise, un_tournoi_au_moyen_age,
    cote_d_ivoire, des_logiciels_pour_apprendre_le_civisme, la_fonte_des_glaciers,
    le_blaireau, l_eau_courante, les_marees_vertes_de_l_ocean,
    les_trous_noirs_de_l_espace, petite_histoire_du_velo, une_drole_de_peche,
    une_journee_en_classe_en_1882, une_maison_pour_l_hiver])).

fait(propriete(p003)).
fait(accessible(p003, oui)).
fait(domaine_val(p003, scale_list, [comprendre_le_texte, ordre, vocabulaire,
    memoire_rapidite_destination, orthographe])).

```

```

fait(propriete(p004)).
fait(accessible(p004, oui)).
fait(domaine_val(p004, scale_list, [questionnaire, questionnaire_2, texte_masque,
  closures, mots_en_desordre, mots_inverses, lignes_en_desordre,
  phrases_en_desordre, texte_a_trous, dictionnaire, synonymes, contraires,
  masculin_feminin, themes_1, themes_2, photographe, memoriser_la_serie,
  mot_tronque, poursuite, trouve_moi, l_absent, le_pendu, mot_coupe,
  ecrire_un_mot, je_corrige, ecrire_une_phrase, ordre_alphabetique])).

```

RÈGLES PÉDAGOGIQUES DU MODÈLE OKEP/ABALECT AU FORMAT XSD

```

<PedagogicalRules>
  <Rule ID="R1">
    <Condition>
      <Parameter ID="P001" TypeOfCondition="Value">
        <Value>CP</Value>
      </Parameter>
    </Condition>
    <Conclusion>
      <Parameter ID="P002" TypeOfConclusion="Domain">
        <Value>L'anniversaire</Value>
        <Value>L'arrivée du loup</Value>
        <Value>L'inondation</Value>
        <Value>Où est mon chat</Value>
        <Value>Une bonne pêche</Value>
        <Value>Une maladie formidable</Value>
      </Parameter>
    </Conclusion>
  </Rule>
  <Rule ID="R2">
    <Condition>
      <Parameter ID="P001" TypeOfCondition="Value">
        <Value>CE1</Value>
      </Parameter>
    </Condition>
    <Conclusion>
      <Parameter ID="P002" TypeOfConclusion="Domain">
        <Value>La vipère</Value>
        <Value>L'avion</Value>
        <Value>Le cheval venu de la mer</Value>
        <Value>Le hamster</Value>
        <Value>Nourrir les oiseaux en hiver</Value>
        <Value>Pirlipi, la chauve-souris</Value>
        <Value>Recette de la mousse</Value>
      </Parameter>
    </Conclusion>
  </Rule>
  <Rule ID="R3">
    <Condition>
      <Parameter ID="P001" TypeOfCondition="Value">
        <Value>CE2</Value>
      </Parameter>
    </Condition>
    <Conclusion>
      <Parameter ID="P002" TypeOfConclusion="Domain">
        <Value>Du cadran solaire à la montre atomique</Value>
        <Value>La fuite de Saddam Hussein</Value>
        <Value>La mante religieuse</Value>
        <Value>La prise du châtelet fort</Value>
        <Value>Le chene</Value>
        <Value>Les fourmis ont du flair</Value>
        <Value>Les Gaulois</Value>
        <Value>Les hirondelles</Value>
        <Value>Un élevage de lapins angoras</Value>
        <Value>Un perroquet bein particulier</Value>
        <Value>Un vétérinaire pour reptiles</Value>
        <Value>Veux-tu élever un chat</Value>
      </Parameter>
    </Conclusion>
  </Rule>

```

```

    </Parameter>
  </Conclusion>
</Rule>
<Rule ID="R4">
  <Condition>
    <Parameter ID="P001" TypeOfCondition="Value">
      <Value>CM1</Value>
    </Parameter>
  </Condition>
  <Conclusion>
    <Parameter ID="P002" TypeOfConclusion="Domain">
      <Value>La sécurité du président</Value>
      <Value>L'aventure du basket</Value>
      <Value>Le tir à l'arc</Value>
      <Value>L'écureuil</Value>
      <Value>Les tremblements de terre</Value>
      <Value>Qui était Louis XIV</Value>
      <Value>Trois femmes agricultrice</Value>
      <Value>Un emploi du temps dans une école japonnaise</Value>
      <Value>Un tournoi au Moyen Age</Value>
    </Parameter>
  </Conclusion>
</Rule>
<Rule ID="R5">
  <Condition>
    <Parameter ID="P001" TypeOfCondition="Value">
      <Value>CM2</Value>
    </Parameter>
  </Condition>
  <Conclusion>
    <Parameter ID="P002" TypeOfConclusion="Domain">
      <Value>Côte d'Ivoire</Value>
      <Value>Des logiciels pour apprendre le civisme</Value>
      <Value>La fonte des glaciers</Value>
      <Value>Le blaireau</Value>
      <Value>L'eau courante</Value>
      <Value>Les marées vertes de l'océan</Value>
      <Value>Les trous noirs de l'espace</Value>
      <Value>Petite histoire du vélo</Value>
      <Value>Une drôle de pêche</Value>
      <Value>Une journée en classe en 1882</Value>
      <Value>Une maison pour l'hiver</Value>
    </Parameter>
  </Conclusion>
</Rule>
<Rule ID="R6">
  <Condition>
    <Parameter ID="P003" TypeOfCondition="Value">
      <Value>Comprendre le texte</Value>
    </Parameter>
  </Condition>
  <Conclusion>
    <Parameter ID="P004" TypeOfConclusion="Domain">
      <Value>Questionnaire</Value>
      <Value>Questionnaire 2</Value>
      <Value>Texte masqué</Value>
      <Value>Closures</Value>
    </Parameter>
  </Conclusion>
</Rule>
<Rule ID="R7">
  <Condition>
    <Parameter ID="P003" TypeOfCondition="Value">
      <Value>Ordre</Value>
    </Parameter>
  </Condition>
  <Conclusion>
    <Parameter ID="P004" TypeOfConclusion="Domain">

```

```

    <Value>Mots en desordre</Value>
    <Value>Mots inversés</Value>
    <Value>Lignes en desordre</Value>
    <Value>Phrases en desordre</Value>
  </Parameter>
</Conclusion>
</Rule>
<Rule ID="R8">
  <Condition>
    <Parameter ID="P003" TypeOfCondition="Value">
      <Value>Vocabulaire</Value>
    </Parameter>
  </Condition>
  <Conclusion>
    <Parameter ID="P004" TypeOfConclusion="Domain">
      <Value>Texte à trous</Value>
      <Value>Dictionnaire</Value>
      <Value>Synonymes</Value>
      <Value>Contraires</Value>
      <Value>Masculin / Féminin</Value>
      <Value>Thèmes 1</Value>
      <Value>Thèmes 2</Value>
    </Parameter>
  </Conclusion>
</Rule>
<Rule ID="R9">
  <Condition>
    <Parameter ID="P003" TypeOfCondition="Value">
      <Value>Mémoire / Rapidité / Destination</Value>
    </Parameter>
  </Condition>
  <Conclusion>
    <Parameter ID="P004" TypeOfConclusion="Domain">
      <Value>Photographier</Value>
      <Value>Mémoriser la série</Value>
      <Value>Mot tronqué</Value>
      <Value>Poursuite</Value>
      <Value>Trouve-moi</Value>
      <Value>L'absent</Value>
    </Parameter>
  </Conclusion>
</Rule>
<Rule ID="R10">
  <Condition>
    <Parameter ID="P003" TypeOfCondition="Value">
      <Value>Orthographe</Value>
    </Parameter>
  </Condition>
  <Conclusion>
    <Parameter ID="P004" TypeOfConclusion="Domain">
      <Value>Le pendu</Value>
      <Value>Mot coupé</Value>
      <Value>Ecrire un mot</Value>
      <Value>Je corrige</Value>
      <Value>Ecrire une phrase</Value>
      <Value>Ordre alphabétique</Value>
    </Parameter>
  </Conclusion>
</Rule>
</PedagogicalRules>

```

RÈGLES PÉDAGOGIQUES DU MODÈLE OKEP/ABALECT TRADUITES EN PRÉDICATS PROLOG

```

regle(r1):-
  si([propiete(p001),
     valeur(p001, cp)]),

```

```
alors([modifier(domain_val(p002, _, _),
  domaine_val(p002, scale_list, [l_anniversaire, l_arrivée_du_loup,
  l_inondation, ou_est_mon_chat, une_bonne_peche, une_maladie_formidable]))]).

regle(r2):-
  si([propriete(p001),
  valeur(p001, ce1)]),
  alors([modifier(domain_val(p002, _, _),
  domaine_val(p002, scale_list, [la_vipere, l_avion, le_cheval_venu_de_la_mer,
  le_hamster, nourrir_les_oiseaux_en_hiver, pirlipi_la_chauve_souris,
  recette_de_la_mousse]))]).

regle(r3):-
  si([propriete(p001),
  valeur(p001, ce2)]),
  alors([modifier(domain_val(p002, _, _),
  domaine_val(p002, scale_list, [du_cadran_solaire_a_la_montre_atomique,
  la_fuite_de_saddam_husseini, la_mante_religieuse, la_prise_du_chateau_fort,
  le_chene, les_fourmis_ont_du_flair, les_gaulois, les_hirondelles,
  un_elevage_de_lapins_angoras, un_perroquet_bien_particulier,
  un_veterinaire_pour_reptiles, veux_tu_elever_un_chat]))]).

regle(r4):-
  si([propriete(p001),
  valeur(p001, cm1)]),
  alors([modifier(domain_val(p002, _, _),
  domaine_val(p002, scale_list, [la_securite_du_president, l_aventure_du_basket,
  le_tir_a_l_arc, l_ecureuil, les_tremblements_de_terre, qui_etait_louis_xiv,
  trois_femmes_agricultrice, un_emploi_du_temps_dans_une_ecole_japonaise,
  un_tournoi_au_moyen_age]))]).

regle(r5):-
  si([propriete(p001),
  valeur(p001, cm2)]),
  alors([modifier(domain_val(p002, _, _),
  domaine_val(p002, scale_list, [cote_d_ivoire,
  des_logiciels_pour_apprendre_le_civisme, la_fonte_des_glaciers,
  le_blaireau, l_eau_courante, les_marees_vertes_de_l_ocean,
  les_trous_noirs_de_l_espace, petite_histoire_du_velo, une_drole_de_peche,
  une_journee_en_classe_en_1882, une_maison_pour_l_hiver]))]).

regle(r6):-
  si([propriete(p003),
  valeur(p003, comprendre_le_texte)]),
  alors([modifier(domain_val(p004, _, _),
  domaine_val(p004, scale_list, [questionnaire_1, questionnaire_2,
  texte_masque, closures]))]).

regle(r7):-
  si([propriete(p003),
  valeur(p003, ordre)]),
  alors([modifier(domain_val(p004, _, _),
  domaine_val(p004, scale_list, [mots_en_desordre, mots_inverses,
  lignes_en_desordre, phrases_en_desordre]))]).

regle(r8):-
  si([propriete(p003),
  valeur(p003, vocabulaire)]),
  alors([modifier(domain_val(p004, _, _),
  domaine_val(p004, scale_list, [texte_a_trous, dictionnaire, synonymes,
  contraires, masculin_feminin, themes_1, themes_2]))]).

regle(r9):-
  si([propriete(p003),
  valeur(p003, memoire_rapidite_destination)]),
  alors([modifier(domain_val(p004, _, _),
  domaine_val(p004, scale_list, [photographier, memoriser_la_serie,
  mot_tronque, poursuite, trouve_moi, l_absent]))]).
```

```

regle(r10):-
  si([propiete(p003),
     valeur(p003, orthographe)],
     alors([modifier(domaine_val(p004, _, _),
                        domaine_val(p004, scale_list, [le_pendu, mot_coupe,
                        ecrire_un_mot, je_corrige, ecrire_une_phrase, ordre_alphabetique]))]).

```

PROPRIÉTÉS TECHNIQUES

```

<TechnicalProperties NameOfFile="Abalect">
  <Application>
    <Folder>C:\Program Files\Abalect\</Folder>
    <Executable>Abalect.exe</Executable>
  </Application>
  <ToCreate>
    <File ID="F001">NomEleve.bil</File>
  </ToCreate>
  <Folders>
    <Folder>CP\</Folder>
    <Folder>CE1\</Folder>
    <Folder>CE2\</Folder>
    <Folder>CM1\</Folder>
    <Folder>CM2\</Folder>
  </Folders>
  <Menus>
    <Item Level="1" Titre="Enseignant"/>
    <Item Level="1" Titre="Elève"/>
    <Item Level="1" Titre="Quitter"/>
  </Menus>
</TechnicalProperties>

```

RÈGLES TECHNIQUES

```

<TechnicalRules>
  <FileCreation ID="F001">
    <FileName>
      <Var>EleveEprofilea</Var>
      <Text>.bil</Text>
    </FileName>
    <FileContent Empty="true"/>
  </FileCreation>
  <PedagogicalContent ID="A001">
    <Interface>
      <XSL>exercice.xsl</XSL>
      <RulesXSL>
        <Text>Pour le texte "</Text>
        <Var>P002</Var>
        <Text>", faites les exercices suivants : </Text>
        <Return/>
        <Var Separator="#32#,">P004</Var>
        <Text>.</Text>
        <Return/>
      </RulesXSL>
    </Interface>
  </PedagogicalContent>
  <PedagogicalOrganization>
    <Interface>
      <XSL>sequence.xsl</XSL>
      <RulesXSL>
        <Text>Selectionner le menu Elève.</Text>
        <Return/>
        <Text>Choisissez votre nom dans la liste.</Text>
        <Return/>
        <PedagogicalContent ID="A001" min="1" max="100">
          </PedagogicalContent>

```



```

    </Return/>
  </RulesXSL></Interface>
</PedagogicalOrganization>
</TechnicalRules>

```

FEUILLE DE STYLE EXERCICE.XSL DU MODÈLE OKEP/ABALECT

```

<xsl:stylesheet version='1.0' xmlns:xsl='http://www.w3.org/1999/XSL/Transform'>
  <xsl:output method="text"/>
  <xsl:template match="/">
    <xsl:for-each select="/eiah/caracterisation/Param">
      <xsl:if test="@ID="p002">
        <xsl:text>Pour le texte "</xsl:text>
        <xsl:value-of select="@valeur"/>
        <xsl:text>",</xsl:text> faites les exercices suivants :</xsl:text>
        <xsl:for-each select="/eiah/caracterisation/Param">
          <xsl:if test="@ID="p004">
            <xsl:text> "</xsl:text>
            <xsl:value-of select="@valeur"/>
            <xsl:text>"</xsl:text>
          </xsl:if>
        </xsl:for-each>
        <xsl:text>.</xsl:text>
      </xsl:if>
    </xsl:for-each>
  </xsl:template>
</xsl:stylesheet>

```

FEUILLE DE STYLE SEQUENCE.XSL DU MODÈLE OKEP/ABALECT

```

<xsl:stylesheet version='1.0' xmlns:xsl='http://www.w3.org/1999/XSL/Transform'>
  <xsl:output method="text"/>
  <xsl:template match="/">
    <xsl:text>Selectionner le menu Elève.#</xsl:text>
    <xsl:text>Choisissez votre nom dans la liste.#</xsl:text>
    <xsl:for-each select="/Sequence/Exercice/enonce">
      <xsl:value-of select="."/>
      <xsl:text>#</xsl:text>
    </xsl:for-each>
  </xsl:template>
</xsl:stylesheet>

```

ANNEXE J. TECHNOLOGIES UTILISÉES LORS DE LA MISE EN ŒUVRE DES PROCESSUS DE GEPPETO_s

Dans cette annexe, nous situons sur les architectures des processus GEPPETO_s, présentées dans la section 6.4.3, page 153, les technologies utilisées.

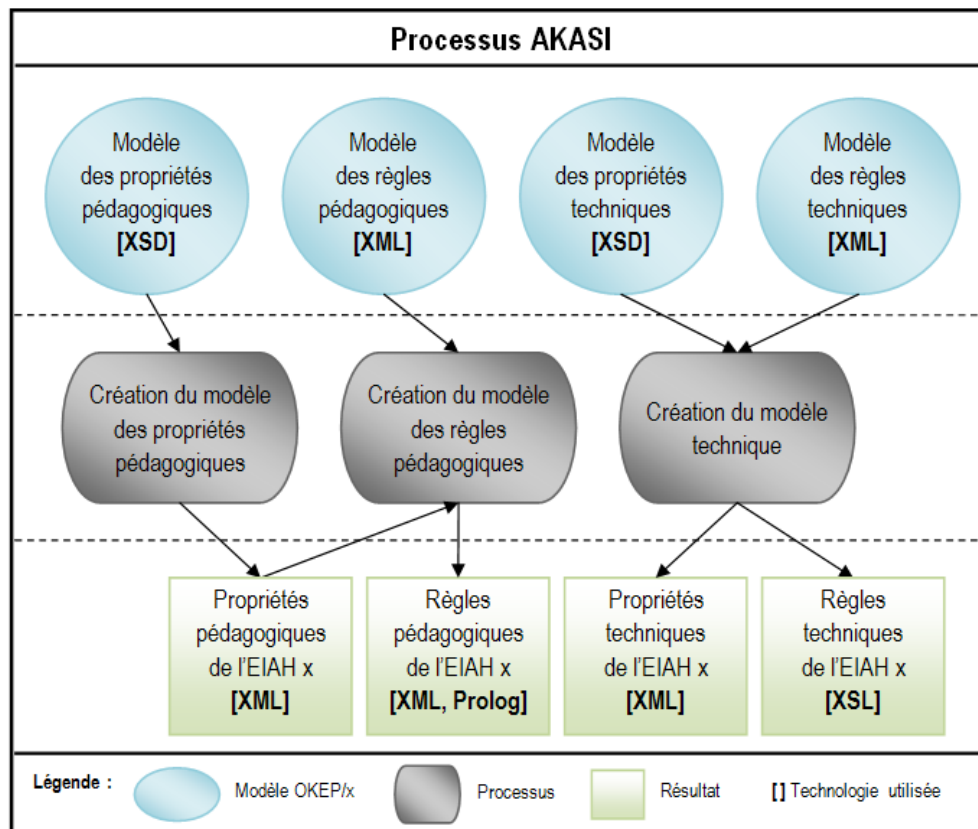


Figure A-56 : Technologies utilisées dans le processus AKASI.

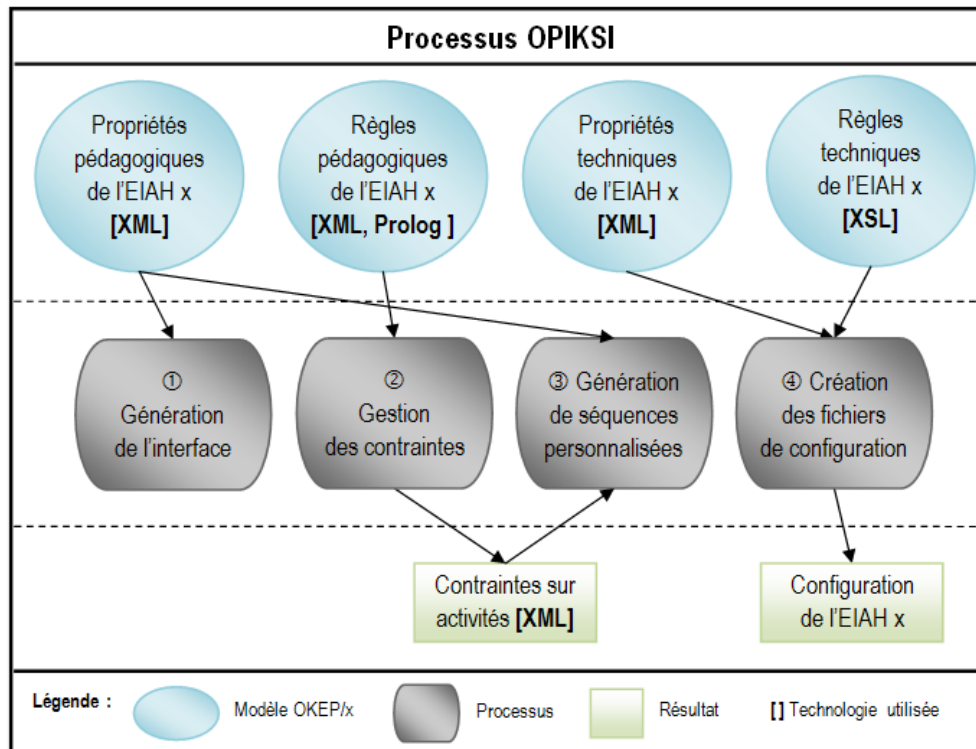


Figure A-57 : Technologies utilisées dans le processus OPIKSI.

ANNEXE K. QUESTIONNAIRE SUR LES PRATIQUES ÉDUCATIVES

Dans le cadre de nos recherches, nous souhaitons obtenir des informations sur vos pratiques d'enseignement. Merci de prendre de prendre 15 minutes pour nous aider.

Les réponses au questionnaire seront traitées de façon anonyme.

Vous trouverez une version électronique de ce questionnaire à l'adresse :

<http://liris.cnrs.fr/stephanie.jean-daubias/projets/p-perlea.html>

UTILISATION DE PROFILS D'ÉLÈVES

Nous abordons tout d'abord l'utilisation des profils d'élèves. Nous entendons par profil d'élèves un ensemble d'informations (connaissances, compétences, etc. sous forme de notes, d'appréciation...) sur un élève. On parle aussi de bilan de compétences.

Q1 : Quel usage faites-vous des profils, des notes, des résultats de vos élèves :

- Création de groupes, dans quel but ?
- Visualisation par les élèves
- Visualisation par d'autres personnes (précisez) :
- Personnalisation des activités fournies aux élèves
- Adaptation de votre enseignement
- Fixation d'objectifs pour inciter les élèves à progresser
- Travail / discussion avec les élèves sur leur profil
- Travail / discussion avec les élèves sur le profil de classe / groupe
- Autres (précisez) :

Q2 : Créez-vous des profils pour :

- Chacun de vos élèves
- Des groupes d'élèves au sein d'une classe
- Chacune de vos classes

Q3 : Dans quel but :

- Visualisation des profils
- Travail sur les profils
- Proposition d'activités adaptées aux profils

Q4 : Récupérez-vous des informations concernant le parcours de vos élèves les années précédentes ? Oui / Non

Si oui, lesquelles, dans quel but ?

Q5 : Possédez-vous des référentiels de compétences (liste de compétences traitées dans un enseignement) ? Oui / Non

Si oui :

- Les utilisez-vous ? Oui / Non
- Dans quel but ?
- Ces référentiels évoluent-ils dans le temps ? Oui / Non
- S'ils évoluent, quel est le taux de modification d'une année scolaire à l'autre (par exemple, environ 30% du référentiel est modifié chaque année) :

Q6 : Utilisez-vous le logiciel J'ADE ? Oui / Non

Q7 : À qui montrez-vous les profils ou résultats de vos élèves, sous quelle forme et dans quel but ? Est-ce une obligation ?

	Forme	But	Obligation
Élève			
Famille			
Collègue			
Institution scolaire			
Autres (précisez) :			
.....			
.....			

UTILISATION DE LOGICIELS PEDAGOGIQUES

Nous abordons ici l'utilisation des logiciels pédagogiques, c'est-à-dire tous les logiciels que vous ou vos élèves utilisent dans un cadre éducatif.

Q8 : Utilisez-vous des logiciels pédagogiques ? Oui / Non

Si oui, quel est leur nom ? Permettent-ils de personnaliser les activités proposées aux élèves ? Produisent-ils des profils d'élèves ou de groupes, des synthèses de résultats ? Utilisez-vous ces profils ?

Nom du logiciel	Personnalisation d'activités		Création de profils		Utilisation des profils	
	Manuellement (par qui et comment ?)	Automatiquement	Pour chaque élève	Pour un groupe	Actuellement	Dans l'idéal

PERSONNALISATION DE L'ENSEIGNEMENT

Nous abordons maintenant la personnalisation de l'enseignement, c'est-à-dire le fait de proposer des activités pédagogiques adaptées à chaque élève ou groupe d'élèves.

Q9 : Détaillez les situations (travail en classe, travail à la maison, travail en salle informatique, remédiation...) pour lesquelles vous faites ou vous souhaitez mettre en place une personnalisation (comme dans l'exemple de la 1^{ère} colonne). Utilisez autant de colonnes que vous avez de situations différentes.

SITUATION				
Contexte	Révision brevet			
Niveau	Collège 3 ^{ème}			
Durée	1h			
Périodicité	2 fois par semaine			
Lieu	Salle informatique			
Effectif	2 groupes de 10 élèves			
PERSONNALISATION				
Par groupe / élève	Groupe			
Faite / souhaitée	Faite			
À toutes les séances	Non			
Remarques	Encadré par un professeur de la discipline			

Q10 : Actuellement, que personnalisez-vous :

- la longueur des séquences d'activités
- le contenu
- le niveau de difficulté
- autres (précisez) :

Q11 : Êtes-vous intéressé par :

- des exercices adaptés à vos élèves
- des logiciels pédagogiques avec sessions personnalisées

Q12: Est-ce que l'on vous demande de proposer :

- des exercices adaptés à vos élèves
- des logiciels pédagogiques avec sessions personnalisées

Q13 : Êtes-vous prêt à utiliser un logiciel pour vous aider lors de la personnalisation des activités proposées aux élèves ? Oui / Non

Q14 : Lors de la création d'activités personnalisées, comment choisissez-vous quelle activité sera faite par quel apprenant ? Selon quel critère ?
.....
.....

Q15 : Combien de temps êtes-vous prêt à consacrer à la personnalisation de votre enseignement :

- pour mettre en place la personnalisation, ponctuellement dans l'année : heures
- pour adapter les activités, chaque semaine :
 - pour obtenir des feuilles d'exercices à imprimer : heures
 - pour avoir des logiciels pédagogiques avec des sessions personnalisées : heures

QUESTIONS GÉNÉRALES

Pour terminer, merci de répondre à ces questions pour nous permettre de mieux vous connaître.

Prénom : Nom (facultatif) :

Âge : ans Sexe : Féminin / Masculin

À quel niveau enseignez-vous :

- Maternelle
- Primaire
- Collège
- Lycée
- Université
- Formation continue
- Autres (précisez) :

Discipline(s) :

Établissement(s) :

Disposez-vous d'une salle informatique au sein de votre (vos) établissement(s) ? Oui / Non

Si oui :

- De combien de postes :
- Pendant des créneaux de : min
- Avec quelle fréquence :
-
-

Êtes-vous restreint dans votre (vos) enseignement(s) quant à l'utilisation de l'ordinateur par vos élèves ? Oui / Non

Si oui, pourquoi ?
.....
.....

Pour préparer vos enseignements, vous utilisez des ordinateurs de manière :

Quotidienne Hebdomadaire Mensuelle Moins souvent Jamais

Dans votre vie quotidienne, vous utilisez des ordinateurs de manière :

Quotidienne Hebdomadaire Mensuelle Moins souvent Jamais

Nous vous remercions de votre participation. Rendez ce questionnaire à la personne qui vous l'a fourni ou renvoyez-le nous à l'adresse ci-dessous.

Marie Lefevre, Stéphanie Jean-Daubias

LIRIS – Bâtiment Nautibus – 23-25 avenue Pierre de Coubertin – 69100 Villeurbanne

marie.lefevre@liris.cnrs.fr, stephanie.jean-daubias@liris.cnrs.fr

<http://liris.cnrs.fr/stephanie.jean-daubias/projets/p-perlea.html>

RÉSUMÉ

Cette thèse en informatique se situe dans le domaine des Environnements Informatiques pour l'Apprentissage Humain (EIAH). Dans ce cadre, nous avons abordé la question de la personnalisation de l'apprentissage.

Nos travaux de recherche consistaient à identifier un processus qui permette à la fois de personnaliser des séances de travail sur papier et des séances de travail sur des logiciels pédagogiques. Nous souhaitions que ce processus permette de prendre en compte les spécificités de chaque apprenant en s'appuyant sur son profil, mais qu'il prenne également en compte les buts et les habitudes pédagogiques des enseignants. Enfin, nous souhaitions que ce processus soit implémentable dans un système externe aux logiciels à personnaliser.

Notre problématique s'est donc décomposée en trois points : comment exploiter les profils d'apprenants pour prendre en compte les individualités des apprenants ? Comment adapter une activité pour prendre en compte les besoins et habitudes pédagogiques d'un enseignant ? Et enfin, comment attribuer une activité à un apprenant ?

Pour répondre au premier point, nous avons proposé le modèle cPMDL. Ce complément du langage de modélisation des profils PMDL permet de contraindre les profils des apprenants afin de sélectionner ceux ayant les caractéristiques requises par les contraintes. cPMDL nous permet donc d'exploiter les informations contenues dans les profils au sein du processus de personnalisation.

Pour répondre au deuxième point, nous avons proposé l'approche GEPPETO. Cette approche s'appuie sur des modèles et des processus génériques permettant d'adapter les activités en fonction des intentions pédagogiques des enseignants grâce à la définition de contraintes sur les activités. Nous avons décliné cette approche pour permettre l'adaptation des activités papier (GEPPETO_p), ainsi que pour l'adaptation des activités logicielles et de la configuration des environnements qui les supportent (GEPPETO_s).

Pour répondre au troisième point, nous avons proposé le modèle PERSUA2 qui permet de lier les contraintes sur profils de cPMDL aux contraintes sur activités de GEPPETO. Ces liens, nommés règles d'affectation, sont ensuite hiérarchisés selon leur degré de priorité pour former une stratégie pédagogique qui sera associée à un ou plusieurs contextes d'utilisation.

Nous avons mis en œuvre ces différentes contributions théoriques dans Adapte, un module de l'environnement informatique du projet PERLEA. Le rôle de cet environnement est d'assister l'enseignant dans la gestion de profils créés par l'enseignant lui-même ou issus de logiciels pédagogiques. Adapte est l'une des exploitations possibles des profils : le module réalisé permet de fournir à chaque apprenant des activités adaptées à son profil tout en respectant les choix pédagogiques de son enseignant. Ces activités peuvent être des activités papier proposées par le système ou des activités logicielles personnalisées par Adapte, mais effectuées dans un autre EIAH. Ce module, pleinement opérationnel, a montré la faisabilité technique de nos contributions théoriques et nous a permis de conduire des mises à l'essai auprès d'enseignants.

MOTS-CLÉS

Environnements Informatiques pour l'Apprentissage Humain (EIAH), processus unifié de personnalisation des activités pédagogiques, méta-modèles et modèles de connaissances, outil d'assistance pour l'enseignant.

ABSTRACT

This thesis in computer science belongs to the field of Interactive Learning Environments (ILE). In this context, we have addressed the issue of personalization of learning.

Our research has consisted in identifying a process allowing one to personalize both paper working sessions and working sessions on educational software. Our goal was to design a process able to take into account the specificities of each learner, based on their profiles, but also to take into account the pedagogical goals and habits of teachers. Moreover, we had to design this process such as it could be easily implemented in a software external to the system being personalized.

Our problem was therefore decomposed into three points: how to use learners' profiles to take the individuality of learners into account? How to adapt a pedagogical activity to take the teaching needs and habits of a teacher into account? And finally, how to assign an activity to a learner?

To answer the first point, we have proposed the cPMDL model. This complement of the profiles modelling language PMDL allows one to constrain the learners' profiles to select those with the characteristics required by the constraints. cPMDL allows us to exploit the information contained in the profiles during the personalization process.

To answer the second point, we have proposed the GEPPETO approach. This approach relies on generic models and processes to adapt activities according to the teachers' intentions, by defining constraints on activities. We have instantiated this approach to enable on the one hand the adaptation of paper activities (GEPPETO_P) and on the other hand to enable the adaptation of software activities and the adaptation of configuration of environments that support them (GEPPETO_S).

To address the third point, we have proposed the PERSUA2 model which links cPMDL constraints on profiles with GEPPETO constraints on activities. Next, these links, called assignment rules, are organized into a hierarchy according to their priority degree in order to form a pedagogical strategy. This pedagogical strategy is then associated with one or more contexts of use.

We have implemented these different theoretical contributions in Adapte, a module of the environment associated to the PERLEA project. The role of this environment is to assist teachers in the management of profiles created by themselves or coming from pedagogical software. Adapte is one of the possible uses of profiles: the module developed provides each learner with activities suited to their profiles, while respecting the teaching choices of the teacher. These activities may be paper activities proposed by the system or software activities personalized by Adapte but made within another ILE. This module, fully operational, has demonstrated the technical feasibility of our theoretical contributions and has allowed us to conduct experiment with teachers.

KEYWORDS

Interactive Learning Environments (ILE), Technology Enhanced Learning (TEL), unified process for personalization of learning activities, meta-models and models of knowledge, assistance tool for teachers.