



HAL
open science

Ressources et parcours pour l'apprentissage du langage Python : aide à la navigation individualisée dans un hypermédia épistémique à partir de traces

Mahdi Miled

► **To cite this version:**

Mahdi Miled. Ressources et parcours pour l'apprentissage du langage Python : aide à la navigation individualisée dans un hypermédia épistémique à partir de traces. Education. École normale supérieure de Cachan - ENS Cachan; École Nationale des Sciences de l'Informatique (La Manouba, Tunisie), 2014. Français. NNT : 2014DENS0045 . tel-01149191

HAL Id: tel-01149191

<https://theses.hal.science/tel-01149191>

Submitted on 6 May 2015

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.



**THESE DE DOCTORAT
EN COTUTELLE ENTRE L'ECOLE NORMALE SUPERIEURE DE CACHAN
ET L'ECOLE NATIONALE DES SCIENCES DE L'INFORMATIQUE**

Présentée par

Monsieur Mahdi Miled

pour obtenir le grade de

DOCTEUR DE L'ECOLE NORMALE SUPERIEURE DE CACHAN

Domaine :

INFORMATIQUE

Sujet de la thèse :

**RESSOURCES ET PARCOURS POUR L'APPRENTISSAGE DU LANGAGE PYTHON :
AIDE A LA NAVIGATION INDIVIDUALISEE DANS UN HYPERMEDIA EPISTEMIQUE A
PARTIR DE TRACES**

Thèse présentée et soutenue à l'ENS de Cachan le 26 novembre 2014 devant le jury
composé de :

Alain Derycke	Professeur émérite	Président
Sébastien George	Professeur	Rapporteur
Vanda Luengo	Maître de conférence HDR	Rapporteur
Eric Bruillard	Professeur	Directeur de thèse
Henda Ben Ghezela	Professeure	Co-directrice de thèse
Christophe Reffay	Maître de conférences	Encadrant
Mona Laroussi	Maître de conférences	Co-encadrante

Remerciements

Je tiens à remercier Eric Bruillard de m'avoir accueilli au sein du laboratoire STEF à l'ENS de Cachan et d'avoir dirigé mes travaux de recherche. J'ai pu profiter de son expérience de la recherche tout en gardant une liberté dans mes travaux.

Je remercie Henda Hajjami Ben Ghezela d'avoir accepté d'être ma co-directrice de recherche au laboratoire RIADI à l'ENSI dans le cadre de cette thèse en cotutelle.

Mes remerciements vont à Christophe Reffay avec qui j'ai pu travailler d'une façon régulière et échanger, grâce à son suivi rigoureux, tout au long de cette thèse.

Je remercie Mona Laroussi pour sa disponibilité et ses précieux conseils qui m'ont permis de mener ces travaux à terme.

Ma reconnaissance va également à Vanda Luengo et à Sébastien George d'avoir accepté d'évaluer mon travail et d'être rapporteurs de cette thèse. Leurs judicieuses suggestions et propositions m'ont donné la possibilité d'améliorer la qualité de ce présent manuscrit.

J'exprime mes sincères remerciements à Alain Derycke qui m'a fait l'honneur d'être président de ce jury.

Ma gratitude va envers Elisabeth Delozanne pour son suivi et ces repères méthodologiques qui se sont avérés fort utiles.

Je remercie l'Institut Français de Coopération qui a financé en partie mes séjours de recherche à l'ENS de Cachan.

Je n'oublie pas de remercier l'association France-IOI de m'avoir aidé à lancer les différentes expérimentations et plus particulièrement Mathias Hiron et Loïc Février pour leur coopération.

Je souhaite également remercier toutes les personnes qui ont contribué de près ou de loin à couronner ce travail.

Table des matières

REMERCIEMENTS	2
TABLE DES MATIERES	3
TABLE DES ILLUSTRATIONS.....	7
TABLEAUX	9
TABLE DES ALGORITHMES.....	9
INTRODUCTION GENERALE	10
1. <i>Motivations, problématique et objectifs de recherche</i>	<i>10</i>
2. <i>Le contexte de l'apprentissage du langage de programmation Python.....</i>	<i>10</i>
3. <i>Méthodologie et déroulement.....</i>	<i>11</i>
4. <i>Contributions</i>	<i>12</i>
5. <i>Organisation du mémoire</i>	<i>13</i>
PARTIE I : ANCRAGES THÉORIQUES, REVUES DE LA LITTÉRATURE ET POSITIONNEMENT DES TRAVAUX.....	15
INTRODUCTION DE LA PARTIE I :	15
CHAPITRE 1 : LES RESSOURCES NUMERIQUES ET LES PARCOURS D'APPRENTISSAGE.....	16
I. INTRODUCTION	17
II. L'OBJET D'APPRENTISSAGE : DIFFÉRENTES DENOMINATIONS MAIS DES OBJECTIFS CONVERGENTS	17
III. PROPRIÉTÉS DES RESSOURCES D'APPRENTISSAGE ET LEUR INDEXATION	18
1. <i>Quelques propriétés des ressources d'apprentissage numériques.....</i>	<i>18</i>
2. <i>L'indexation de ressources d'apprentissage.....</i>	<i>19</i>
IV. MODÈLES DE DESCRIPTION DES RESSOURCES D'APPRENTISSAGE BASES SUR LA HIERARCHIE ET LA GRANULARITE	21
1. <i>Le modèle CISCO.....</i>	<i>21</i>
2. <i>Le modèle de Heiwy et Ducateau.....</i>	<i>21</i>
3. <i>Le modèle SCORM.....</i>	<i>22</i>
4. <i>Le modèle Learnativity.....</i>	<i>23</i>
5. <i>Le modèle de Verbert et Duval.....</i>	<i>24</i>
VI. L'ÉVALUATION DES OBJETS D'APPRENTISSAGE	25
VII. DES RESSOURCES D'APPRENTISSAGE VERS DES PARCOURS D'APPRENTISSAGE	25
1. <i>La notion de parcours d'apprentissage</i>	<i>25</i>
2. <i>La scénarisation pédagogique</i>	<i>26</i>
3. <i>Le graphe des épistèmes comme ressources et parcours pour le langage Python.....</i>	<i>26</i>
VIII. CONCLUSION	27
CHAPITRE 2 : LES ENVIRONNEMENTS D'ACCES ET D'EDITION DE RESSOURCES NUMERIQUES D'APPRENTISSAGE.....	30
I. INTRODUCTION	31
II. LES ENVIRONNEMENTS D'ACCES AUX RESSOURCES.....	31
1. <i>Les entrepôts de ressources d'apprentissage et les environnements généralistes : centraliser les ressources pour faciliter l'accès.....</i>	<i>32</i>
2. <i>Les LMS</i>	<i>33</i>
3. <i>Les environnements dédiés à l'apprentissage de la programmation.....</i>	<i>34</i>
III. LA CONCEPTION ET L'ÉDITION COLLABORATIVE DE RESSOURCES.....	35
1. <i>L'édition collaborative de modèles.....</i>	<i>36</i>
2. <i>Le Cas MediaWiki.....</i>	<i>36</i>
3. <i>Quelques outils de travail collaboratif (CSCW) et outils-auteurs pour la création de ressources ...</i>	<i>36</i>
4. <i>Services pour l'accès aux ressources et activités collaboratives.....</i>	<i>36</i>
5. <i>Plateformes pour l'accès et la création de ressources d'apprentissage</i>	<i>37</i>
IV. CONCLUSION	40

CHAPITRE 3 : VISUALISATION, NAVIGATION DANS LES RESSOURCES ET TRAÇAGE D'ACTIVITES.....	42
I. INTRODUCTION	43
II. LE CHAMP DE LA VISUALISATION ET LA NAVIGATION	43
1. <i>La visualisation de l'information et la navigation dans les graphes</i>	43
2. <i>La visualisation de l'information et l'aide à la navigation dans les EIAH</i>	44
3. <i>Quelques outils et interfaces de visualisation de graphes</i>	45
III. LES TRACES D'INTERACTIONS.....	49
1. <i>La notion de trace en EIAH</i>	49
2. <i>Traces et indicateurs en EIAH</i>	49
3. <i>Objectifs et finalités</i>	49
4. <i>Traces primitives et traces modélisées</i>	50
5. <i>Indicateurs sur les traces</i>	51
6. <i>L'expérience tracée et l'acquisition dynamique de connaissances</i>	51
7. <i>Les traces déclaratives</i>	51
8. <i>Liens entre traces d'activité et traces déclaratives</i>	51
IV. CONCLUSION	52
CHAPITRE 4 : PERSONNALISATION ET HYPERMEDIA ADAPTATIFS.....	53
I. INTRODUCTION	54
II. GENESE : DE LA VERSATILITE, L'ADAPTABILITE A LA PERSONNALISATION	54
III. PROFIL, MODELE D'UTILISATEUR ET MODELE D'APPRENANT	54
IV. TECHNIQUES D'ADAPTATION DES HYPERMEDIAS.....	55
1. <i>Les techniques de présentation adaptative</i>	55
2. <i>Les techniques de navigation et de support adaptatifs</i>	56
V. SYSTEMES HYPERMEDIA ADAPTATIFS.....	57
1. <i>Tendances de l'adaptation : le web sémantique et la recommandation</i>	58
2. <i>Suivi et personnalisation des objets d'apprentissage et des parcours</i>	59
VI. CONCLUSION	59
CHAPITRE 5 : LA FOUILLE DE DONNEES EN EDUCATION	61
I. INTRODUCTION	62
II. TECHNIQUES ET ALGORITHMES COMMUNS EN FOUILLE DE DONNEES EN EDUCATION	62
1. <i>Apprentissage supervisé</i>	62
2. <i>Apprentissage non supervisé</i>	64
III. QUELLES ANALYSES SONT NECESSAIRES ET QUE RECHERCHER ?.....	66
IV. OUTILS ET PLATEFORMES SPECIFIQUES POUR LA FOUILLE DE DONNEES EN EDUCATION.....	68
V. TENDANCES ET PERSPECTIVES SUR LA FOUILLE DE DONNEES EDUCATIVES.....	69
VI. CONCLUSION	70
CONCLUSION DE LA PARTIE I.....	71
PARTIE II : D'UN GRAPHE STATIQUE DES ÉPISTÈMES VERS UN HYPERMÉDIA ÉPISTÉMIQUE ADAPTATIF	73
INTRODUCTION DE LA PARTIE II :	73
CHAPITRE 1 : APPROCHE GENERALE ET GLOBALE : ENTRE EDITION ET NAVIGATION	74
I. INTRODUCTION	75
II. UN CYCLE DE VIE DES RESSOURCES BASE SUR L'ARTICULATION ENTRE LES ACTIVITES D'EDITION ET DE NAVIGATION : UNE RELATION NON TRIVIALE	75
1. <i>Points de cadrage et contraintes</i>	75
2. <i>Pourquoi lier édition et navigation ?</i>	76
3. <i>Comment lier ces activités ?</i>	76
4. <i>Modélisation générale</i>	77
III. LES APPORTS VISEES DE CETTE APPROCHE.....	78
IV. CONCLUSION	80
CHAPITRE 2 : LE GRAPHE DES EPISTEMES : LE SYSTEME NERVEUX DES RESSOURCES ET DES PARCOURS.....	81
I. INTRODUCTION	82

II.	DESCRIPTION DU GRAPHE DES EPISTEMES.....	82
1.	<i>Objectifs et motivations</i>	82
2.	<i>La plateforme France-IOI</i>	83
3.	<i>Graphe des concepts et des prérequis</i>	83
III.	ADAPTATION DU GRAPHE DES EPISTEMES POUR LE CAS FRANCE-IOI	86
IV.	CONCLUSION	88
CHAPITRE 3 : CONCEPTION COLLABORATIVE, NAVIGATION DYNAMIQUE ET MODELISATION DES TRAJECTOIRES		89
I.	INTRODUCTION	90
II.	MODELISATION DU PROCESSUS COLLABORATIF.....	90
1.	<i>Du modèle de conception à l'édition de micro-ressources</i>	90
2.	<i>Modèle de l'apprenant, de la tâche et du domaine</i>	92
3.	<i>Modélisation des états d'une unité</i>	93
4.	<i>Modèle de rafraîchissement du graphe des unités selon l'influence des parcours antérieurs</i>	94
5.	<i>Mécanismes automatiques de mise à jour</i>	97
III.	MODELISATION DES TRAJECTOIRES	97
1.	<i>Définitions</i>	98
2.	<i>Comment mesurer et évaluer les trajectoires ?</i>	99
3.	<i>Modélisation des trajectoires avec des réseaux sémantiques</i>	101
IV.	INSTANCIATION DE LA DEPENDANCE ENTRE EDITION ET NAVIGATION	104
V.	CONCLUSION	106
CHAPITRE 4 : ARCHITECTURE GENERALE DES PROTOTYPES.....		107
I.	INTRODUCTION	108
II.	UN OUTIL DE VISUALISATION COMME PIVOT A LA NAVIGATION DYNAMIQUE	108
1.	<i>Objectifs</i>	108
2.	<i>Intégration d'un outil de visualisation</i>	108
3.	<i>Test d'outils de visualisation des graphes</i>	108
III.	UNE MISE EN APPLICATION DES EPISTEMES SUR CLOUDSCORM.....	111
IV.	PROTOTYPE PRECEPTOR	112
1.	<i>Fonctionnalités et utilisateurs concernés</i>	112
2.	<i>Traces liées à l'édition et à la navigation</i>	113
3.	<i>Choix de l'implémentation</i>	114
4.	<i>Accès aux fonctionnalités</i>	116
V.	PROTOTYPE HIPPY	120
1.	<i>Modèles de données et éléments d'implémentation</i>	120
2.	<i>Navigation dans l'hypermédia épistémique</i>	122
3.	<i>Mise en place de l'évaluation épistémique individualisée</i>	124
VI.	CONCLUSION	128
CHAPITRE 5 : VERS DES PROCESSUS UNIFIES D'ANALYSE DE DONNEES EN EDUCATION..		130
I.	INTRODUCTION	131
II.	SELECTION, MODIFICATION ET INTERROGATION DE DONNEES SUR DES REQUETES SIMPLES ET CROISEES 131	
1.	<i>Requêtes sur les macro-trajectoires</i>	131
2.	<i>Requêtes sur les micro-trajectoires</i>	132
3.	<i>Requêtes sur les trajectoires épistémiques</i>	133
III.	VISUALISATION ET EXPLOITATION DE DONNEES PAR DES OUTILS DE FOUILLE DE DONNEES	133
1.	<i>Importation de données issues de tables</i>	134
2.	<i>Adaptation des données</i>	134
3.	<i>Application des algorithmes de fouille</i>	135
IV.	CROISEMENT DE DONNEES ISSUES DE TRACES ET D'UN QUESTIONNAIRE : VERS LA MISE EN RELATION DES DONNEES DECLARATIVES ET OBJECTIVES	137
V.	CONCLUSION	137
CONCLUSION PARTIE II		138
PARTIE III : MISE À L'EPREUVE, EXPERIMENTATION ET RESULTATS		139
INTRODUCTION DE LA PARTIE III :.....		139

CHAPITRE 1 : ANALYSE COMPARATIVE D'ALGORITHMES DE VISUALISATION	140
I. INTRODUCTION	141
II. PROTOCOLE ET METHODOLOGIE SUIVIS	141
III. MESURES DES TEMPS D'EXECUTION	142
IV. COMPARATIFS ET RESULTATS	142
V. CONCLUSION	145
CHAPITRE 2 : TERRAIN ET EXPERIMENTATION DE L'HYPERMEDIA EPISTEMIQUE HIPPY	146
I. INTRODUCTION	147
II. TERRAIN EXPERIMENTAL	147
III. METHODOLOGIE	147
1. <i>Hypothèses de départ</i>	147
2. <i>Scénarios et tâches</i>	148
3. <i>Utilisabilité et utilité en EIAH</i>	148
4. <i>Moyens et mise en œuvre</i>	149
5. <i>Contexte et étapes de l'expérimentation</i>	149
IV. CONCLUSION	150
CHAPITRE 3 : RESULTATS ET EVALUATIONS	151
I. INTRODUCTION	152
II. EVALUATION ANALYTIQUE AVEC LES CRITERES IHM	152
III. RESULTATS IDENTIFIES A PARTIR DES TRACES	152
1. <i>Données générales sur les trajectoires</i>	152
2. <i>Résultats des algorithmes de fouille</i>	156
3. <i>Constats et discussion</i>	161
4. <i>Une première mise à jour des ressources à partir des données de navigation</i>	161
IV. RESULTATS DU QUESTIONNAIRE	162
1. <i>Un questionnaire destiné aux apprenants</i>	162
V. COMBINAISON DE RESULTATS ISSUS DU QUESTIONNAIRE AVEC LES DONNEES SUR LES TRACES	169
VI. CONCLUSION	171
CONCLUSION PARTIE III.....	172
CONCLUSION GENERALE	173
INDEX.....	176
REFERENCES BIBLIOGRAPHIQUES.....	178
ANNEXES	193
I. MODÈLE DE DONNÉES POUR LE GRAPHE DES ÉPISTÈMES SOUS DHOSTON SPRINGY	194
II. MODÈLE DE DONNÉES SOUS JSON (ADAPTÉ AU FRAMEWORK JIT).....	200
III. FICHIER IMSMANIFEST.XML (ARCHIVE SCORM).....	218
IV. QUELQUES FICHIERS ZEND FRAMEWORK (MODELS, VIEWS ET CONTROLLERS)	224
1. <i>Configuration</i>	224
2. <i>Organisation d'une application</i>	225
3. <i>Quelques fichiers de l'application PRECEPTOR</i>	225
V. MODÈLE DE TRAJECTOIRES EN RDF	255
VI. FICHIERS PHP DU PROTOTYPE HIPPY	263
1. <i>Extraits du Fichier Episteme.inc.php</i>	263
VII. EXEMPLE DE PROCESSUS D'ANALYSE EN XML (GÉNÉRÉ PAR RAPIDMINER)	265
1. <i>Weka Hierarchical Clustering</i>	265
VIII. QUESTIONNAIRE	267
RESUME	270
ABSTRACT.....	270

Table des illustrations

Figure 1 : liens entre les chapitres de la partie I.....	15
Figure 2 : cycle de vie des ressources selon différentes approches (Catteau, 2008a).....	20
Figure 3 : modèle CISCO.....	21
Figure 4 : modèle de ressources d'apprentissage (Heiwy et Ducateau, 2003).....	22
Figure 5 : modèle SCORM.....	23
Figure 6 : Learnativity content model (d'après Duval et Hodgins 2003).....	24
Figure 7 : modèle de Verbert et Duval.....	24
Figure 8 : cycle de vie selon l'approche ARIADNE.....	32
Figure 9 : page d'accueil de MERLOT.....	33
Figure 10 : Code Academy.....	35
Figure 11 : page d'accueil de Connexions.....	38
Figure 12 : un exemple d'activité (sur les variables) dans Khanacademy.....	39
Figure 13 : graphes radiaux.....	44
Figure 14 : Treemaps.....	44
Figure 15 : Arbor.js.....	46
Figure 16 : Canviz.....	46
Figure 17 : Js-graph.it.....	47
Figure 18 : Pearltrees.....	48
Figure 19 : transformation de M-traces (Djouad et al., 2009).....	50
Figure 20 : techniques d'adaptation (d'après Brusilovsky).....	57
Figure 21 : cycle d'adaptation (Brusilovsky, 1996).....	58
Figure 22 : arbre de décision induit par C4.5 (d'après Wu & Kumar, 2010).....	63
Figure 23 : résultat de la convergence du clustering (Wu & Kumar, 2010).....	64
Figure 24 : architecture en couches de l'hypermédia épistémique.....	72
Figure 25 : liens entre les chapitres de la partie II.....	73
Figure 26 : dépendances entre conception et utilisation de ressources.....	77
Figure 27 : relation entre activité d'édition et de navigation.....	77
Figure 28 : illustration d'une portion du graphe des épistèmes (D'après P. Ortiz).....	84
Figure 29 : explicitation du sous-graphe des notions utiles pour l'exercice « Labyrinthe »....	84
Figure 30 : le sujet de l'exercice "Labyrinthe".....	85
Figure 31 : l'épistème "Afficher".....	86
Figure 32 : construction du graphe résultant France-IOI.....	87
Figure 33 : processus de conception et d'édition collaborative.....	91
Figure 34 : diagramme de cas d'utilisation.....	91
Figure 35 : diagramme de collaboration.....	92
Figure 36 : exercice et épistèmes.....	93
Figure 37 : états des unités.....	94
Figure 38 : structure d'une micro-trajectoire.....	98
Figure 39 : ensemble ordonné d'exercices, thèmes, objectif et exemple de macro-trajectoire de l'apprenant A ₁	99
Figure 40 : composants d'une trajectoire épistémique.....	99
Figure 41 : structure des trajectoires à 3 niveaux.....	100
Figure 42 : modélisation d'un parcours effectif.....	101
Figure 43 : réseau sémantique d'une trajectoire avec Protégé.....	101

Figure 44 : graphe conceptuel partiel d'une trajectoire	103
Figure 45 : processus de révision à partir des trajectoires.....	105
Figure 46 : une portion du graphe des épistèmes avec Dhoston Springy.....	109
Figure 47 : une portion du graphe des épistèmes avec ForceDirected.....	110
Figure 48 : une portion du graphe des épistèmes avec HyperTree	110
Figure 49 : une portion du Graphe des épistèmes avec RGraph	111
Figure 50 : liste d'unités épistémiques sur l'environnement CloudScorm.....	112
Figure 51 : processus de génération des traces d'édition et de navigation	114
Figure 52 : structure de l'application Preceptor.....	115
Figure 53 : page d'accueil du prototype PRECEPTOR.....	116
Figure 54 : requête Ajax de traçage	116
Figure 55 : une vue du graphe des épistèmes sur le prototype PRECEPTOR	117
Figure 56 : édition du modèle JSON.....	118
Figure 57 : interface d'évaluation d'une unité	119
Figure 58 : édition de métadonnées sur les unités.....	120
Figure 59 : modèle conceptuel de données de HiPPY	122
Figure 60 : exemple de graphe non-orienté tâche	123
Figure 61 : sous-graphe de la tâche "Labyrinthe"	124
Figure 62 : envoi de l'identifiant de l'épistème ouvert	125
Figure 63 : exemple d'enregistrement de traces	125
Figure 64 : adaptation de la navigation et diagnostic épistémique.....	126
Figure 65 : requête concernant les macro-trajectoires.....	131
Figure 66 : requête croisée sur les micro-trajectoires.....	133
Figure 67 : requête donnant les épistèmes non utilisés	133
Figure 68 : aperçu quantitatif sur les événements de la trajectoire épistémique.....	133
Figure 69 : transformation des attributs 2,3 et 7 (idUser, idEpisteme et idContext) de numériques en nominaux	134
Figure 70 : exécution de Simple Kmeans dans Weka.....	135
Figure 71 : exemple de processus avec RapidMiner.....	136
Figure 72 : exemple de processus avec Knime	136
Figure 73: exemple de requête de croisement entre traces et questionnaire	137
Figure 74 : liens entre les chapitres de la partie III	139
Figure 75 : étapes menant au benchmark	141
Figure 76 : temps d'exécution selon le type de visualisation	142
Figure 77 : temps d'exécution pour ForceDirected	143
Figure 78 : temps d'exécution pour HyperTree	143
Figure 79 : temps d'exécution pour RGraph.....	144
Figure 80 : nombre d'utilisateurs en fonction du nombre d'exercices résolus.....	153
Figure 81 : extrait d'une trajectoire épistémique pour l'utilisateur 21372	154
Figure 82 : extrait de micro-trajectoire de l'utilisateur 21372.....	154
Figure 83 : extrait de micro-trajectoire de l'utilisateur 5379 pour la tâche 1869 (Hello World !)	155
Figure 84 : temps passés sur les différents onglets pour trois utilisateurs (21134, 21867 et 21909).....	155
Figure 85 : résultat de Kmeans fast avec n=4	157
Figure 86 : arbre partiel de décision généré par Knime	159
Figure 87 : motifs récurrents dans les micro-trajectoires	160
Figure 88 : évaluation de la navigation dynamique (n=121 : tous).....	164
Figure 89 : évaluation de la navigation dynamique (à gauche n=58 : garçons hors ISN, à droite n=32 : garçons bacheliers hors ISN).....	164

Figure 90 : évaluation de la navigation dynamique (n=14, programmeurs non débutants)...	165
Figure 91 : évaluation du diagnostic épistémique individualisé (à gauche n=12 : tous, à droite n=58 : garçons hors ISN)	165
Figure 92 : évaluation du diagnostic épistémique individualisé (n=26 : débutants hors ISN)	166
Figure 93 : évaluation de l'utilité de l'onglet « Notions associées » (à gauche n=121 : tous, à droite n=26 : bonnes notions mais pas de pratique)	166
Figure 94 : nuage de mots-clés issu des réponses ouvertes.....	168
Figure 95 : graphe de dépendances des mots-clés issus des réponses ouvertes	169
Figure 96 : relation entre l'utilité déclarée et l'usage effectif	170
Figure 97 : organisation d'une application Zend Framework.....	225

Tableaux

Tableau 1 : comparatif de quelques bibliothèques de visualisation dynamique	48
Tableau 2 : acteurs et interventions sur les ressources d'apprentissage.....	79
Tableau 3 : indicateurs sur les épistèmes et les parcours	94
Tableau 4 : actions de rafraîchissement sur la ressource (selon le niveau et les acteurs concernés).....	96
Tableau 5: couches supportées par les prototypes	128
Tableau 6 : quelques chiffres sur FUOE, FUTO et TUOE	170

Table des algorithmes

Algorithme 1 : génération du modèle individualisée du graphe	126
Algorithme 2 : calcul de la fermeture transitive (Warshall).....	127
Algorithme 3 : construction adaptée de la fermeture transitive du graphe	127
Algorithme 4 : calcul des exercices sautés	132
Algorithme 5 : calcul des exercices inversés	132

Introduction générale

Si la question de partage et d'édition de ressources d'apprentissage a déjà été abordée, cette question est renouvelée avec le contexte du web 2.0 et leur application dans le monde de l'éducation est particulièrement prometteuse. Dans ce contexte participatif et contributif, les utilisateurs ne sont plus seulement consommateurs de contenus mais ils en sont également producteurs.

1. Motivations, problématique et objectifs de recherche

Inscrit dans le champ des EIAH, l'objectif de nos travaux est de proposer, d'une part, des mécanismes visant à soutenir l'apprenant dans son parcours d'apprentissage à travers un réseau d'unités d'apprentissage et à assister, dans une moindre mesure, le concepteur ou l'enseignant dans l'édition du contenu de ces unités. Cela nous amène, de ce fait, à réfléchir à la relation (qui n'est pas forcément triviale) entre conception/édition d'objets d'apprentissage et utilisation/navigation dans les objets d'apprentissage. Celle-ci nous permettra ainsi d'expérimenter le cas de l'apprentissage du langage de programmation Python.

Plus spécifiquement, les orientations de nos recherches sont principalement ciblées sur l'exploration de moyens de visualisation et de navigation individualisée dans un graphe de ressources appelé graphe des épistèmes (voir la section VII.3 du chapitre 1 de la partie I) pour pouvoir procurer à l'apprenant un bilan de connaissances personnalisé. Ces questions font intervenir les éléments suivants :

- la visualisation globale ou localisée des ressources selon le point de vue de l'acteur (étudiant, enseignant/concepteur, tuteur),
- les parcours conseillés (ou parcours types) et les objectifs selon les profils,
- et les traces individuelles et/ou groupées des apprenants (parcours effectifs, degré de satisfaction aux objectifs d'apprentissage...).

2. Le contexte de l'apprentissage du langage de programmation Python

Python est un langage de programmation développé par Guido Van Rossum¹. C'est un langage à typage dynamique à haut niveau de structures de données.

Même si l'on peut considérer que Python manque parfois de promotion, certaines caractéristiques le rendent plutôt attractif :

- il possède une syntaxe simple, c'est le langage à apprendre s'il on veut débiter la programmation par rapport au C jugé parfois plus rebutant ;
- il existe très peu de mots-clés à retenir ;
- le nombre de bibliothèques incluses ou à importer permet d'arriver très rapidement à un résultat concret ;
- dans les autres langages de programmation à apprendre, on commence par progresser très vite puis la progression générale diminue ;

¹ <http://www.python.org/~guido/>

- l'évolution constante du langage permet de disposer de nouvelles fonctionnalités demandées par la communauté de façon rapide et réfléchie.

Parmi les domaines d'application du langage nous pouvons citer :

- le développement web et internet,
- l'accès aux bases de données,
- les interfaces graphiques de bureau,
- les applications scientifiques et numériques,
- la programmation réseau (socket),
- le développement de logiciels,
- et les jeux et graphiques 3D.

Python pourrait être un candidat susceptible de succéder à Perl, surtout sur les aspects de développement web (framework django).

Bien qu'il n'ait pas eu une progression fulgurante, Python fait partie des rares langages avec C# à montrer une progression constante depuis les dix dernières années, alors que les langages leaders (Java, C, C++, Visual Basic, PHP) ont tous tendance à baisser.

Il existe divers contenus pour promouvoir l'apprentissage du langage Python. Quick associates² est un jeu de rôles consistant à développer en Python en tant que prestataires de services (immersion dans le monde de l'entreprise).

Ainsi, notre cas d'application sera l'apprentissage de la programmation (le langage Python en particulier). Notre démarche vise à aider à concevoir des ressources régies par le « vécu » des utilisateurs en tenant compte de l'historique et des interactions pour mieux mettre à jour les ressources appropriées. Ceci aura pour rôle de créer une dynamique entre les activités d'utilisation et d'édition et d'apporter un aspect auto-évolutif à partir des traces de parcours antérieurs servant à la recommandation de nouveaux parcours.

3. Méthodologie et déroulement

Après avoir cerné quelques environnements d'apprentissage (LMS, réservoirs de ressources, sites généralistes et dédiés à une discipline spécifique...) et leurs potentialités de co-édition de ressources, nous avons établi différentes modélisations permettant de décrire les acteurs, les états de validation d'une unité et l'articulation entre édition et navigation. Ce dernier élément d'articulation nous a permis d'identifier une représentation cyclique plus fine entre les activités de révision ou d'édition avec la recommandation de parcours. L'étape qui a mené vers la prospection des outils permettant de visualiser et de manipuler des graphes interactifs, nous a permis de voir quels types de visualisation étaient les plus adaptés aux situations que nous voulions intégrer. Il s'agit, en l'occurrence, d'une situation orientée apprenant : ceci comprend la navigation libre et la navigation suggérée (parcours prescrit) et une situation orientée-enseignant/tuteur/concepteur : ceci rassemble les activités d'édition intra-unité, intragraphe (ou et intergraphe). D'après les résultats des tests de performances sur les simulations effectuées, nous avons pu constater qu'il était plus adapté de corréler :

- à une situation d'apprentissage :

² <http://elc.fhda.edu/quicksoft/index.html>

- le type de visualisation Rgraph ou HyperTree à la situation d'apprentissage de navigation libre (graphe entier des unités) ;
- le type ForceDirected à une navigation guidée (parcours représentant un sous-graphe) ;
- à une situation d'édition :
 - le type de visualisation ForceDirected pour l'édition intragraphe ;
 - le type de visualisation HyperTree/Rgraph pour une édition intergraphe.

Nous pensons qu'une navigation adaptative et interactive peut aider l'utilisateur (apprenant ou tuteur) à mieux se représenter les différentes unités et leurs dépendances qu'il est parfois difficile de percevoir dans une navigation statique. De plus, ceci peut apporter des éléments d'adaptation concernant la présentation et faciliter les opérations d'édition et de mise à jour du graphe. Prenant en compte les types de visualisations cités précédemment, nous avons conçu un prototype web proposant un accès différencié (étudiant, tuteur, enseignant/concepteur) baptisé PRECEPTOR (Plateforme d'utilisation et de gestion paRticipativE de ressourCEs PyThon et de parcOuRs). Un aspect de l'architecture de PRECEPTOR, montre le processus de génération des traces d'édition (en termes de copie/dérivation, évaluation/révision et édition de métadonnées) et de navigation (par rapport aux états des unités et des parcours effectifs des apprenants).

Un autre prototype d'hypermédia épistémique adaptatif HiPPY (an epistemic HyPermedia to learn PYthon) a été mis à l'épreuve dans la plateforme France-IOI³, l'objectif étant de pouvoir étudier, de caractériser les parcours des apprenants utilisant le langage Python et d'évaluer les aspects de navigation individualisée.

Nous avons ainsi pu élaborer les premières conclusions pour pouvoir éventuellement généraliser certaines propriétés, modélisations ou orientations dans d'autres langages de programmation ou dans d'autres disciplines.

4. Contributions

Nos principales contributions sont d'ordre conceptuel, informatique, méthodologique et expérimental:

- conceptuel : par l'introduction d'une approche corrélant les activités d'édition avec les activités de navigation ;
- informatique :
 - à travers la proposition d'un prototype d'hypermédia web basé sur une navigation individualisée ;
 - par la présentation de démarches d'extraction de connaissances utiles à partir de traces (utilisation, composition et adaptation d'algorithmes dans des chaînes de fouilles de données) mettant en évidence une différenciation dans les trajectoires;
 - nos apports en informatique se présentent de façon plus détaillée aux niveaux :
 - de l'interface homme/machine (impact de l'interface sur le comportement des utilisateurs) ;

³ www.france-ioi.com, association qui prépare aux olympiades internationales d'informatique. Cette plateforme en ligne propose des exercices d'entraînement en programmation et en algorithmique dans différents langages (C, C++, Python, Pascal, Java, JavaScool et OCaml).

- de l'ingénierie logicielle : architecture en 3-tiers, MVC⁴, technologies AJAX pour le traçage;
 - de l'ingénierie des connaissances et de la fouille de données.
- méthodologique et expérimental : à partir du processus d'élaboration d'une expérimentation à large échelle et exploitation de données.

5. Organisation du mémoire

Ce mémoire s'articule en trois parties :

- Partie I: Ancrages théoriques, revues de la littérature et positionnement des travaux

La 1^{ère} partie, subdivisée en cinq chapitres, a pour objectif de donner les fondements théoriques et les liens avec les objectifs visés et les questions posées dans nos recherches dans le domaine des EIAH. Différents modèles, réalisations et méthodes sont soit analysés, soit énumérés pour délimiter notre champ de recherche et ancrer notre positionnement scientifique.

- Le premier chapitre traite des **ressources d'apprentissage** et analyse la notion de parcours d'apprentissage dans les EIAH.
- Les différentes typologies d'**environnements d'accès aux ressources** d'apprentissage indépendamment des disciplines ciblées sont répertoriées dans le 2^{ème} chapitre. L'accent est également mis sur la conception et l'**édition** de ces ressources.
- Le 3^{ème} chapitre a pour finalité de montrer l'importance du champ de la **visualisation** et la **navigation** dans les ressources d'apprentissage ainsi que la nécessité des **traces** et les visées derrière ce mécanisme dans les EIAH.
- Le 4^{ème} chapitre explicite les aspects théoriques de la personnalisation et les hypermédias **adaptatifs**.
- Le 5^{ème} chapitre donne des rappels des techniques de **fouille de données** et des perspectives d'utilisation dans les environnements d'apprentissage.

L'ensemble des unités épistémiques (formalisé par un DAG⁵) constitue une **méta-ressource d'apprentissage** pour le langage Python où chaque unité épistémique (ou épistème) est une micro-ressource. L'utilisation et l'exploitation de cet ensemble de ressources génèrent des **parcours effectifs** qui sont parfois divergents des **parcours prescrits**.

La méta-ressource peut être exploitée et intégrée dans un **environnement d'accès aux ressources** (dans un LMS, un environnement dédié ou un environnement supportant des fonctionnalités de révision et d'**édition**).

Outre une navigation statique, la méta-ressource peut se décliner en un hypermédia utilisant des formes de **visualisation** et de **navigation dynamique** à base de graphes.

Le recours aux **traces** et leur modélisation a pour objet d'octroyer à l'hypermédia des propriétés d'**adaptation** (au profil de l'apprenant, en fonction de son cheminement dans l'apprentissage) en termes de contenu, de navigation et d'évaluation "épistémique".

⁴ Modèle-View-Contrôleur

⁵ Directed acyclic graph : graphe orienté acyclique

L'extraction des données et des connaissances se fait selon des approches classiques de **fouilles de données** à finalité plutôt descriptive que prédictive.

- Partie II: D'un graphe statique des épistèmes vers un hypermédia épistémique adaptatif

La 2^{ème} partie pose les propositions retenues et justifiées en termes de conceptualisations, de modèles et de choix de conceptions pour aider à la réalisation d'outils appropriés.

- Le 1^{er} chapitre introduit l'approche générale originale adoptée, consistant à lier les activités d'édition/conception aux activités d'utilisation et de navigation. Nos travaux ne traitent qu'une partie de cette modélisation.
- Le 2^{ème} chapitre recueille les propositions de modélisation et de transformation dans le but de transformer le graphe des épistèmes en ressources navigables et traçables : modèles de navigation adoptés selon la situation d'édition ou de navigation.
- Le 3^{ème} chapitre détaille les différentes modélisations formelles et informelles proposées: cycle de conception des ressources et édition collaborative ressources, états des unités et leur validation, typologie de parcours, modèle de rafraîchissement du graphe ainsi qu'une modélisation multi-échelle de la trajectoire (niveau macro, micro et épistémique).
- L'objet du 4^{ème} chapitre est de retracer les modèles des prototypes construits, les modèles de description et explicitation des traces, les modèles du domaine et de l'apprenant.
- Le 5^{ème} chapitre s'intéresse à la perspective de processus unifiés d'analyse des données éducatives (workflow d'analyse des trajectoires et le lien avec la réingénierie de l'hypermédia, ou encore métaheuristiques et métamodèles). Ce chapitre s'interrogera aussi sur une piste de recherche féconde : la complémentarité entre métamodèles et fouille de données.

- Partie III: Mise à l'épreuve, expérimentation et résultats

La 3^{ème} partie se propose de mettre à l'épreuve nos modèles proposés et les outils connexes. Une première mise à l'épreuve d'un nombre réduit d'épistèmes sur un environnement similaire à un LMS (CloudScorm) est présentée. Une expérimentation à large échelle dans un milieu écologique y est explicitée (France-IOI).

- Le 1^{er} chapitre décrit une analyse comparative des outils de visualisation open-source ainsi que les différents algorithmes testés.
- Le 2^{ème} chapitre illustre le terrain expérimental et les différentes hypothèses à valider concernant l'intégration de l'hypermédia épistémique dans la plateforme France-IOI.
- Le 3^{ème} chapitre retrace les évaluations et les principaux résultats exploitant les processus de fouille de données.

PARTIE I : ANCRAGES THÉORIQUES, REVUES DE LA LITTÉRATURE ET POSITIONNEMENT DES TRAVAUX

Introduction de la partie I :

Comme nous l'avons énoncé précédemment, un axe d'exploration concerne l'étude des trajectoires des apprenants dans un hypermédia adaptatif ou individualisée. Pour y parvenir, nous avons visé les objectifs intermédiaires suivants :

- pouvoir construire un hypermédia adaptatif sur le web à partir de ressources et de parcours existants basé sur une aide à la navigation individualisée ;
- installer une série d'observables pour rendre l'hypermédia « traçable » ;
- expérimenter l'utilisation de l'hypermédia et récolter les traces d'usages ;
- analyser les traces et les retours d'usage (par questionnaire) ;
- et proposer des mécanismes d'aide à la réingénierie à partir des traces.

Il convient de préciser que ces objectifs font intervenir les concepts de ressources d'apprentissage, de parcours, d'indicateurs à base de traces, de visualisation et de navigation et de fouilles de données. Ces concepts et leurs exploitations informatiques pratiques font l'objet de cette première partie. Les liens entre les chapitres de cette partie sont présentés dans la Figure 1. Dans tout ce qui va suivre, nous donnerons un sens équivalent au graphe des épistèmes, graphe des unités, graphe des unités épistémiques et graphe épistémique. Une tâche aura comme équivalent un exercice.

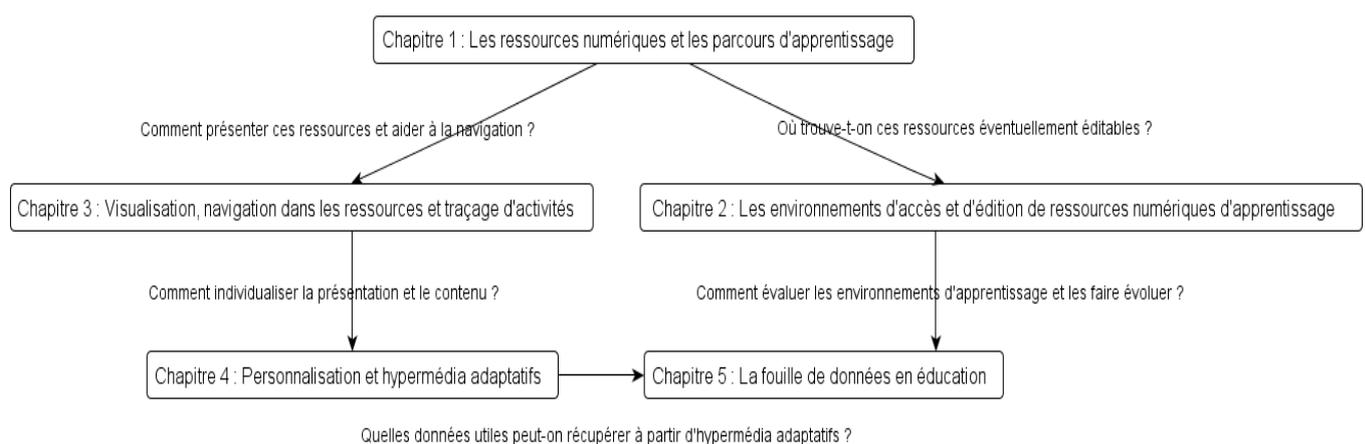


Figure 1 : liens entre les chapitres de la partie I

Chapitre 1 : Les ressources numériques et les parcours d'apprentissage

I.	INTRODUCTION	17
II.	L'OBJET D'APPRENTISSAGE : DIFFERENTES DENOMINATIONS MAIS DES OBJECTIFS CONVERGENTS	17
III.	PROPRIETES DES RESSOURCES D'APPRENTISSAGE ET LEUR INDEXATION	18
1.	<i>Quelques propriétés des ressources d'apprentissage numériques</i>	18
2.	<i>L'indexation de ressources d'apprentissage</i>	19
a.	<i>Quelques normes d'indexation</i>	20
IV.	MODELES DE DESCRIPTION DES RESSOURCES D'APPRENTISSAGE BASES SUR LA HIERARCHIE ET LA GRANULARITE	21
1.	<i>Le modèle CISCO</i>	21
2.	<i>Le modèle de Heiwy et Ducateau</i>	21
3.	<i>Le modèle SCORM</i>	22
4.	<i>Le modèle Learnativity</i>	23
5.	<i>Le modèle de Verbert et Duval</i>	24
VI.	L'EVALUATION DES OBJETS D'APPRENTISSAGE	25
VII.	DES RESSOURCES D'APPRENTISSAGE VERS DES PARCOURS D'APPRENTISSAGE	25
1.	<i>La notion de parcours d'apprentissage</i>	25
2.	<i>La scénarisation pédagogique</i>	26
3.	<i>Le graphe des épistèmes comme ressources et parcours pour le langage Python</i>	26
VIII.	CONCLUSION	27

I. Introduction

Avec la multiplicité des contenus, le développement des moyens d'accès à l'information, l'utilisateur se trouve parfois noyé dans un flot incontrôlé d'informations. Une ressource, quelle que soit sa nature, est une source d'information mais elle n'est pas forcément source effective de connaissances. C'est ainsi que des ressources non nécessairement créées dans un but d'information ou d'apprentissage peuvent parfois servir comme supports à l'apprentissage.

À l'inverse, les ressources créées ou fabriquées intentionnellement dans un but pédagogique ou didactique ont l'avantage d'être ciblées selon ce but. C'est ce que définit (Sosteric & Hesemeier, 2002) comme « *une entité numérique destinée à une utilisation à des fins pédagogiques, qui inclut, de façon interne ou externe, des indications sur le contexte approprié d'utilisation de cet objet* ». C'est sur ce type de ressource contextualisée que nous allons nous focaliser.

L'apparition conceptuelle de la notion d'objet d'apprentissage n'est pas nouvelle. Elle est attribuée indirectement à Gerard R.G (Campbell, 2003a)(Gerard, 1967). L'idée faisait appel au découpage d'un curriculum pour pouvoir réutiliser certaines de ses parties. Cette idée peut être anodine mais elle a permis d'ouvrir d'autres horizons notamment dans le contexte de contenus numériques d'apprentissage.

Plus tard, elle a été vulgarisée par Richard Hodgins en 1992, grâce notamment au groupe de travail « Reusable learning objects » qui introduisait un objet d'apprentissage comme une collection d'objets d'information assemblés utilisant des métadonnées pour s'adapter à la personnalité et aux besoins de l'apprenant (Hodgins, 2000). Quelles sont les typologies de ressources numériques d'apprentissage existantes ?

Quels modèles sont-ils susceptibles d'être utilisables ou adaptés ? Quels sont les éléments à prendre en compte pour une bonne utilisation ou exploitation de ces ressources d'apprentissage ?

Nous expliciterons la définition d'un objet d'apprentissage et ses différentes dénominations et nous présenterons quelques modèles et standards liés à sa description. Nous donnerons quelques pistes sur l'exploitation de ces ressources notamment par leur évaluation. Nous décrirons le passage de ressources vers des parcours pédagogiques que nous illustrerons par le graphe des épistèmes.

II. L'objet d'apprentissage : différentes dénominations mais des objectifs convergents

À l'aide d'une métaphore et à partir d'une taxonomie (Wiley, 2000) donne une définition, pour expliciter les objets d'apprentissage. Selon lui, un objet d'apprentissage est toute ressource numérique qui peut être réutilisée pour faciliter ou soutenir cet apprentissage. Le modèle taxonomique qu'il préconise repose sur la taxonomie de Bloom, à partir de cinq niveaux à savoir :

- les lectures (connaissances) : un contenu narratif lié à un objectif, des évaluations alternatives peuvent être incluses pour aider à acquérir les connaissances nécessaires ;

- les tutoriels (compréhension et application) : des leçons indépendantes qui incluent généralement des objets d'apprentissage réutilisables pour une appropriation pratique et son évaluation ;
- les applications (analyse) : des expériences structurées pour analyser des problèmes ;
- les simulations (synthèse) : des situations favorisant une approche exploratoire et active de collaborations entre étudiants et experts
- étude de cas (évaluation) : des situations complexes présentées d'une manière où les apprenants doivent interpréter et proposer des solutions multiples

Il y a peu de consensus pour définir ce qu'est une ressource numérique d'apprentissage mais il y a différentes dénominations : une ressource numérique d'apprentissage Learning Object (LO) Open Educational resource (OER) (CourseWare⁶) du MIT, définition de ARIADNE. RLO (Reusable Learning Object). Tous ces termes désignent le même concept mais renvoient à des nuances selon l'utilisation et le contexte d'application. Un autre aspect qui mérite d'être cité c'est l'aspect libre ou ouvert des ressources avec wikiversité, wikibooks⁷ ou même creativecommons⁸ pour l'éducation. Par extension, les ressources d'apprentissage numériques ou électroniques peuvent parfois désigner les environnements et les applications éducatives comme les portails, des logiciels propriétaires ou encore des documents ciblés comme les portfolios ou tout document pouvant faire partie d'un scénario pédagogique.

OER Commons⁹ qui est une initiative de l'ISKME (ISKME, 2008) a introduit le type de ressource éducative ouverte (ou libre). Nous verrons que, pour exploiter ce type de ressources, il existe des outils de création et de partage (dans la section outils auteurs pour la conception de ressources). Il est à noter que certaines formes de courseware ont évolué plus récemment vers un nouveau type de modalité matérialisé sous la forme de cours massifs ouverts en ligne (ou Massive Open Online Courses). Nous y reviendrons plus loin dans ce mémoire.

III. Propriétés des ressources d'apprentissage et leur indexation

Nous énonçons ici quelques propriétés qui paraissent nécessaires dans chaque objet d'apprentissage et nous explicitons le concept d'indexation illustré d'un exemple d'utilisation.

1. Quelques propriétés des ressources d'apprentissage numériques

Si certaines caractéristiques les distinguent par rapport à leur type de support de présentation (son, texte, vidéo, multimédia), d'autres caractéristiques plus générales (Agrawal, 2006) doivent être prises en compte comme :

- la granularité (le niveau où intervient l'objet d'apprentissage en question) : est-ce qu'il fait partie d'un module, d'un cours, d'une partie de cours, voire d'une séquence ?

⁶ Il existe notamment un consortium dédié à cette orientation sur <http://www.ocwconsortium.org>

⁷ <http://fr.wikibooks.org/wiki/Accueil>

⁸ <http://creativecommons.org/education>

⁹ <http://www.oercommons.org/>

- l'autonomie : la ressource se doit d'être utilisée d'une façon indépendante ;
- la réutilisabilité ou le réinvestissement : si le type de ressource l'admet, on peut proposer un contenu dans un contexte différent mais assez proche dans la finalité ou l'objectif ;
- l'agrégabilité : il s'agit de types de ressource qui peuvent se présenter avec d'autres pour être réassemblés ou agrégés ;
- et la reconnaissance de la propriété intellectuelle.

2. L'indexation de ressources d'apprentissage

Au sujet de l'indexation (Hernandez, Mothe, Ramamonjisoa, Ralalason, & Stolf 2009) proposent un modèle de représentation multi-facettes des objets d'apprentissage pour améliorer la réutilisabilité en utilisant notamment les métadonnées dérivées de standards, leur usage dans des scénarios et des liens sémantiques entre ces objets. (Broisin, 2005; Catteau, Marquié, Vidal, & Broisin, 2009a) proposent des modèles pour la gestion des ressources d'apprentissage et de suivi de leur cycle de vie (cf. Figure 2). Ce cycle de vie prend généralement en compte l'étape de conception, d'indexation, de validation, d'utilisation ou d'exploitation. (Falquet, Guyot, & Hurni, 2003) proposent des ressources sous forme d'hyperlivre pour l'apprentissage collaboratif. Certaines recherches abordent également la nécessité de la dimension économique dans la gestion des ressources d'apprentissage (Campbell, 2003b). La figure suivante met en évidence l'évolution des ressources (de la création jusqu'à l'utilisation effective) selon différentes approches et dénominations. Comme l'illustre la Figure 2, il n'y a pas consensus sur la dénomination et le contenu des étapes même si certaines sont communes. Ainsi, CanCore aura comme étapes l'initiation, la validation et la terminaison. ARIADNE par exemple, utilisera l'étape de création (*authoring*), la validation, l'indexation et l'exploitation.

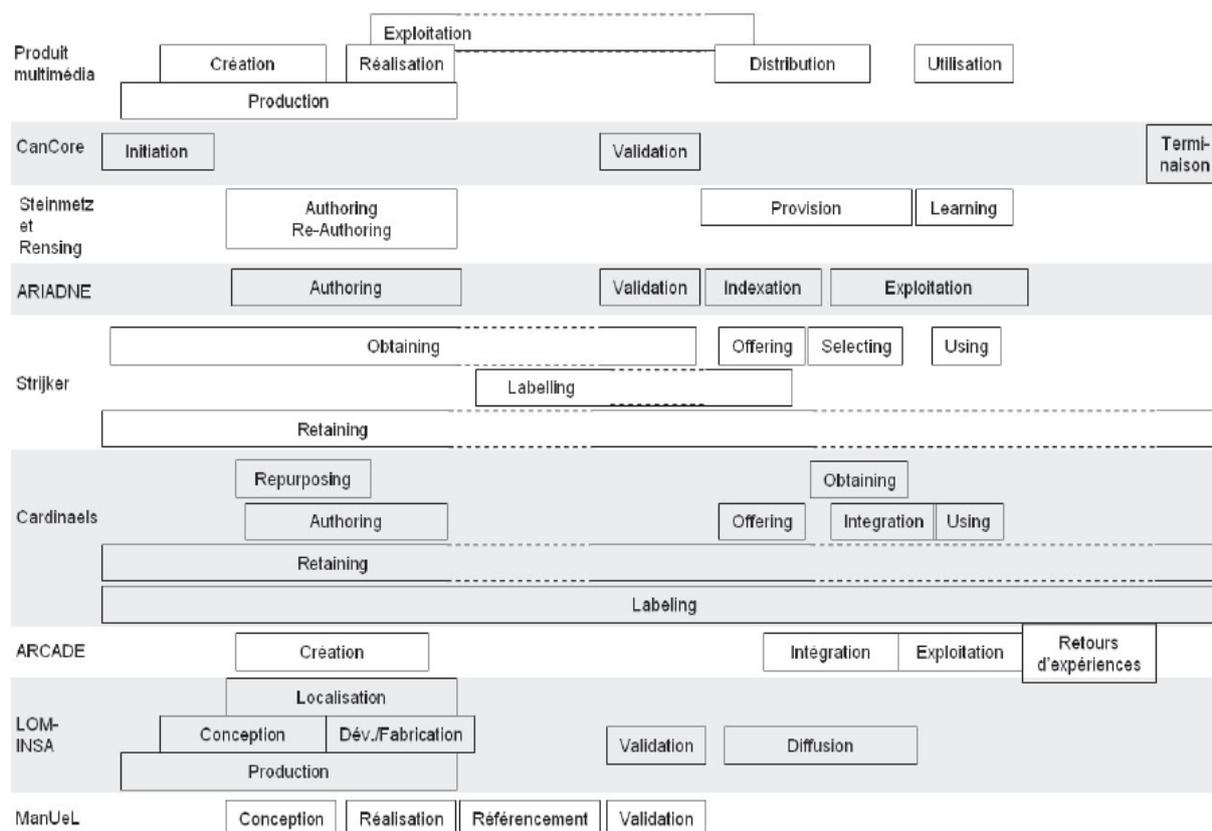


Figure 2 : cycle de vie des ressources selon différentes approches (Catteau, 2008a)

a. Quelques normes d'indexation

Un des objectifs derrière l'indexation est de pouvoir énumérer et détailler les métadonnées d'une ressource d'apprentissage pour faciliter ensuite la recherche, l'exploitation et la réutilisation.

Une des normes les plus répandues est LOM. Il s'agit essentiellement d'une norme d'indexation pour décrire chaque ressource d'apprentissage et pour aider à l'accès par exemple aux grands réservoirs de ressources. Elle contient plusieurs items. Ainsi, chaque objet d'apprentissage peut être décrit à l'aide de métadonnées selon une nomenclature à 9 catégories (De La Passardière & Grandbastien, 2003) : généralités, cycle de vie, méta-métadonnées, informations techniques, informations pédagogiques, droits, relations, commentaires et classification. Il existe d'autres variantes selon des cadres plus spécifiques : LomFr, LRM¹⁰... Il existe d'ailleurs des mécanismes de transformation pour rendre ces normes compatibles entre elles, citons par exemple l'utilisation des transformations IMS LRM vers IEEE LOM 1.0.

D'autres initiatives comme celles avec la norme MPEG-21 (Vaucelle, 2009), une norme de la famille MPEG qui est un cadre de représentation des objets numériques (Digital Item) basé sur un ensemble de modules indépendants, serait une alternative normative pour les ressources d'apprentissage. Cette norme porte non seulement sur la description et l'indexation mais aussi sur l'ensemble des processus de création de ces ressources simples ou composites. Il existe également une partie qui gère les questions de droits.

¹⁰ Learning Resource Meta-data Specification, <http://www.imsglobal.org/metadata/>

IV. Modèles de description des ressources d'apprentissage basés sur la hiérarchie et la granularité

La question de la hiérarchisation et la granularité des ressources est primordiale. Pour décrire ces aspects, nous présentons les modèles CISCO, celui de Heiwy et Ducateau, SCORM, Learnativity et celui de Verbert et Duval.

1. Le modèle CISCO

Le modèle CISCO¹¹ suggéré par (Barritt & Alderman, 2004) introduit la notion d'objet d'apprentissage réutilisable (Reusable Learning Object) et celle d'objet d'information réutilisable (Reusable Information Objects). Un RLO est une collection de 7 ± 2 RIO (Figure 3). Chaque RIO est composé de trois composants : les items de contenu, de pratique et d'évaluation. Les concepteurs ont pour possibilité de créer des objectifs en attachant RLO et RIO. L'un des avantages de cette démarche est de pouvoir utiliser un contenu standardisé qui est susceptible d'être réutilisé par d'autres concepteurs, découpé selon les besoins et être évalué selon des techniques d'évaluation standards.

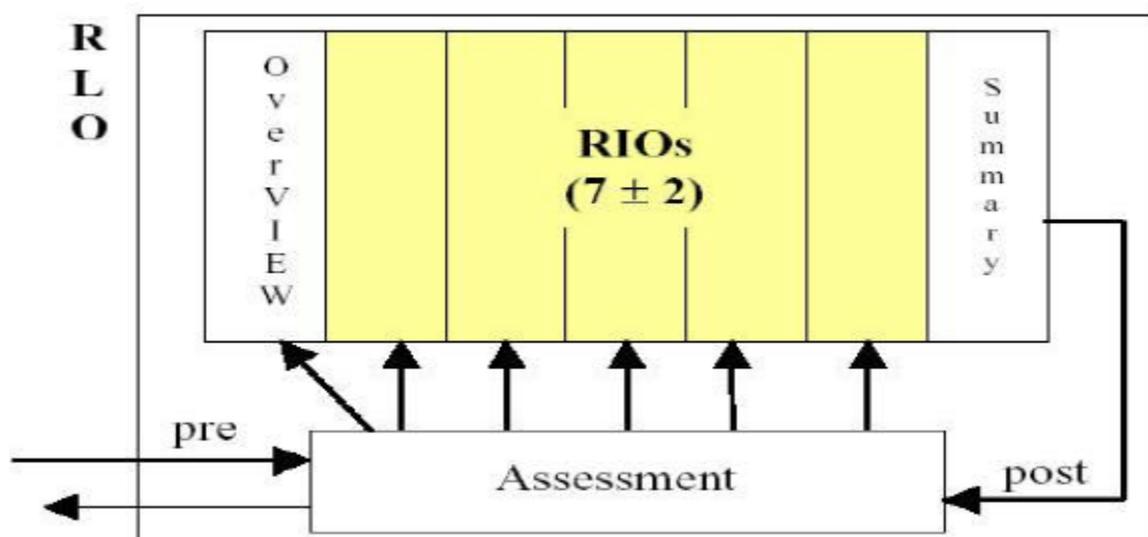


Figure 3 : modèle CISCO

2. Le modèle de Heiwy et Ducateau

Un autre type de modèle de « ressources pédagogiques » (Figure 4), proposé par (Heiwy & Ducateau, 2003), formalisé en UML représente les ressources (d'apprentissage) par des liens de prérequis en ajoutant un aspect temporel pour décrire les parcours et en veillant à séparer le fond de la forme et le contenu de la scénarisation.

¹¹ Tient son nom de la ville de Cisco aux Etats-Unis

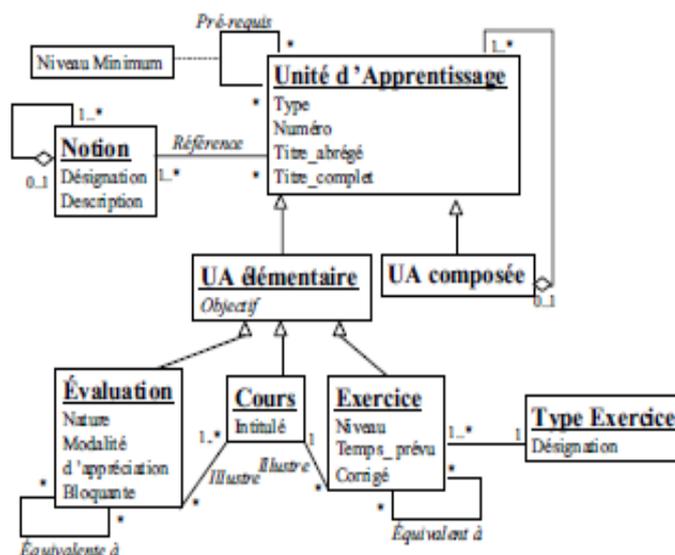


Figure 4 : modèle de « ressources pédagogiques » (Heiwy et Ducateau, 2003)

3. Le modèle SCORM

La norme SCORM (Dodds, 2001) comme illustrée dans la Figure 5 prévoit un mécanisme d'agrégation de contenu et un environnement d'exécution :

- *Content Aggregation Model* (CAM) s'appuie sur des règles de séquençement des activités d'apprentissage (IEEE LTSC, AICC, ADL, IMS, ARIADNE) servant à lire ces contenus;
- *Run Time Environment* (RTE) facilite principalement les échanges entre LMS.

Les ressources d'apprentissage (Learning resources) utilisées peuvent être soit :

- des SCO (Shareable Content Object) : unité d'apprentissage ayant des métadonnées et des tâches d'apprentissage,
- soit des Asset représentant la ressource atomique (fichier html, vidéo...)

L'Agrégation de ces SCOs désigne un paquetage ou encore une unité pédagogique plus élaborée.

Les traces (voir section III du chapitre 3) capturées grâce au RTE (*Run Time Environment*) peuvent renseigner sur le suivi de l'apprenant (tracking model) et sur la progression de l'activité et les interactions avec les ressources notamment en terme d'états de chaque unité pédagogique (complétée, commencée, tentée...).

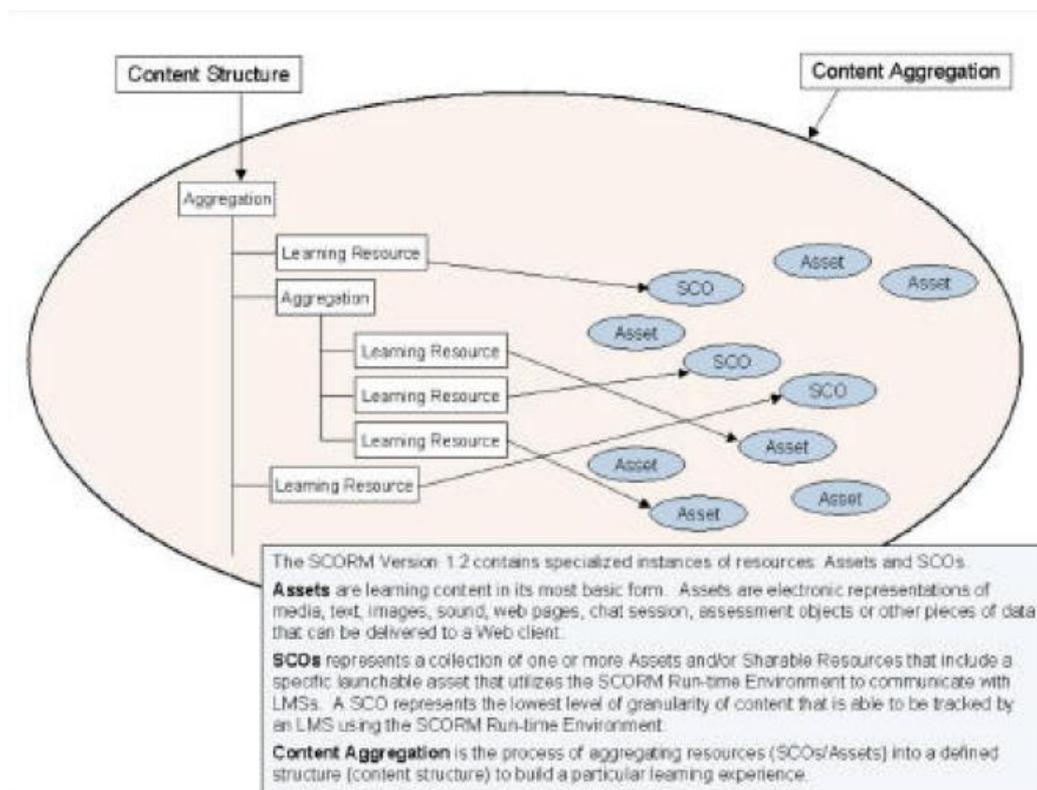


Figure 5 : modèle SCORM

Dans les modèles présentés ci-dessus, la notion de granularité ou de hiérarchie paraît récurrente et importante. Même si chaque modèle a sa propre classification, ces modèles s'accordent à dire que les « grains » ou les éléments de plus petite taille (comme les assets dans le modèle SCORM) sont les plus potentiellement réutilisables et décontextualisables.

4. Le modèle Learnativity

Le modèle Learnativity (Wagner, 2002) explicite cinq niveaux (Figure 6) :

- les données brutes et les éléments médiatiques (le grain le plus fin, animation, illustration..),
- les objets d'information qui est un ensemble d'éléments bruts faisant référence à certains principes, concepts, faits ou procédures,
- les éléments appartenant à un seul objectif semblables aux objets d'apprentissage (learning objects) mais conçus d'une façon plus restrictive,
- les agrégats assemblés sont souvent des leçons ou des chapitres avec des objectifs terminaux plus conséquents,
- et le dernier niveau qui fait référence aux collections, aux cours ou même aux curriculums.

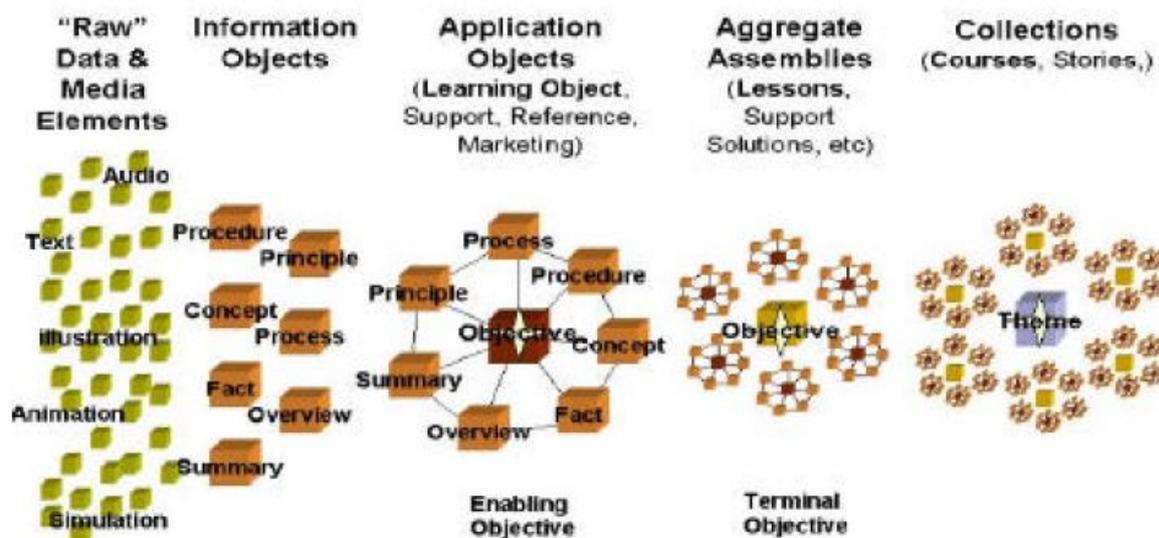


Figure 6 : Learnativity content model (d'après Duval et Hodgins 2003)

5. Le modèle de Verbert et Duval

(Verbert & Duval, 2004) proposent un modèle de ressources d'apprentissage (Figure 7) basé sur la fragmentation des contenus selon des objectifs définis et en séparant les aspects de la navigation des aspects de présentation.

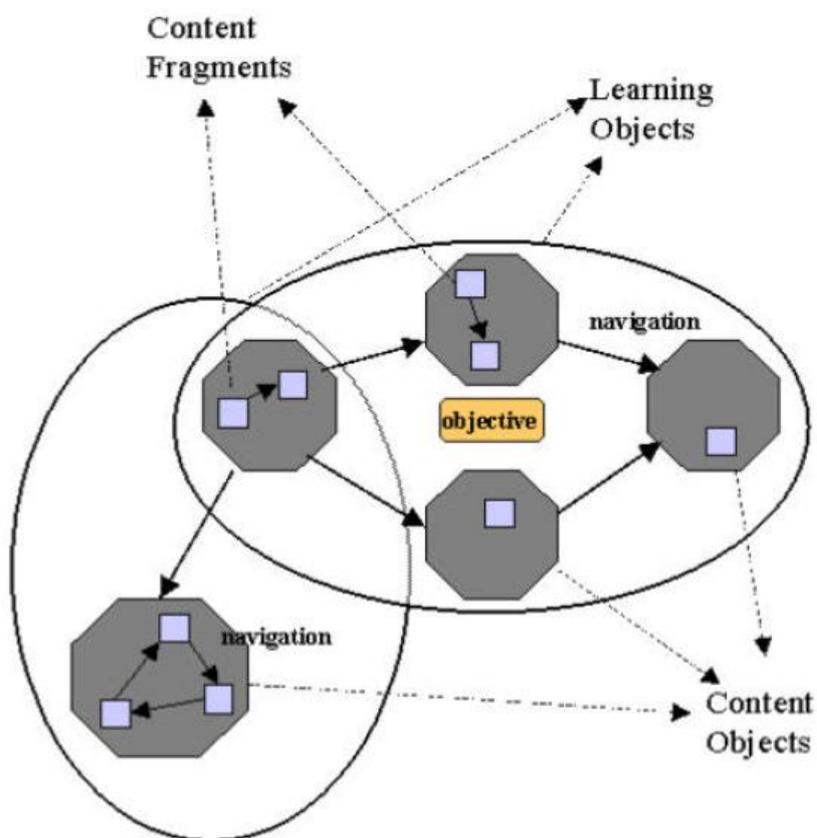


Figure 7 : modèle de Verbert et Duval

Les différents modèles décrits prouvent qu'il existe différentes manières de représenter la granularité des objets d'apprentissage (cours, leçons, exercices, exemples...) et leurs exploitations selon les objectifs et les situations visés (compréhension, application, analyse...).

VI. L'évaluation des objets d'apprentissage

Le fait de pouvoir évaluer les ressources d'apprentissage permet de mieux mettre à jour les contenus, de supprimer ou de modifier certains aspects selon les retours d'expériences des différents utilisateurs. Il existe à cet effet un outil méthodologique d'évaluation des objets d'apprentissage : LORI¹² 1.5 (Nesbit, Belfer, & Leacock, 2003) qui introduit différents critères comme :

- L'exactitude et la qualité du contenu (1),
- l'adéquation avec les objectifs d'apprentissage (2),
- le feedback et l'adaptation (3),
- la motivation (4),
- la présentation et le design (5),
- l'utilisabilité et l'interaction (6),
- l'accessibilité (7),
- la réutilisabilité et le réinvestissement(8)
- et la conformité aux standards (9).

La qualité du contenu, l'adéquation avec les objectifs d'apprentissage et l'accessibilité nous paraissent vitaux afin de pouvoir répondre au mieux aux besoins des tuteurs/concepteurs amenés à réguler leurs contenus et leurs démarches d'intervention.

VII. Des ressources d'apprentissage vers des parcours d'apprentissage

Un parcours peut d'emblée faire référence à une alternative qui peut être choisie parmi un certain nombre de propositions pour atteindre un objectif. Le parcours d'apprentissage peut se décrire sous la forme d'un ensemble d'éléments plus au moins structuré et organisé pour atteindre un objectif intermédiaire ou terminal d'apprentissage. Dans notre cas, les ressources d'apprentissage forment les éléments constitutifs de ce parcours, envisagé comme un processus d'acquisition à exploiter au niveau de l'intervention pédagogique. Nous présentons d'abord la notion de parcours d'apprentissage puis nous donnons un exemple de métalangage de scénarisation pédagogique avec IMS-LD. Nous décrivons ensuite le graphe des épistèmes sur l'apprentissage du langage Python.

1. La notion de parcours d'apprentissage

Essayer de donner une définition claire et précise d'un parcours d'apprentissage n'est pas chose aisée vu les différents points de vue.

¹² Learning Objects Review Instrument

Une définition d'un parcours d'apprentissage serait une séquence d'activités choisie par un apprenant parmi une multitude de séquences d'activités qui permettent de construire les connaissances d'une façon progressive. Il est désigné par « learning path » ou encore par « learning pathway ».

Selon (Jih, 1996) l'intérêt des parcours réside dans le fait qu'ils laissent des traces d'interactions avec les systèmes interactifs d'apprentissage. Ils peuvent construire des transitions personnalisées entre l'information accédée et leur propre base de connaissances.

Le parcours pédagogique comme le rappelle Ernst (2008) fait appel à un trinôme constitué de prérequis, d'un champ cognitif et d'objectifs pédagogiques.

Une autre définition mais appliquée au monde de l'entreprise et du monde professionnel (Rosenbaum & Williams, 2004), souligne que la séquence idéale d'activités d'apprentissage permet aux employés d'atteindre une compétence métier dans un temps le plus court possible, selon le principe de la rentabilité.

Sans pour autant nier l'arrière plan pédagogique correspondant, nous nous attachons à discerner les éléments distinguables dans un scénario pédagogique tel qu'il est défini dans la norme IMS-LD. Plus sommairement, il peut être vu comme étant une séquence d'unités d'apprentissage comme dans l'EIAH ECSAIweb (Sanrach & Grandbastien, 2000) défini pour répondre à des objectifs explicites d'apprentissage.

2. La scénarisation pédagogique

La scénarisation pédagogique peut faire appel à certains outils propriétaires. Il existe également un métalangage de modélisation pédagogique qui s'intitule IMS-LD qui doit son apparition au langage EML (Educational Modeling Language) à l'Open University of the Netherlands en 2003. Plus finement, d'après ses auteurs, il sert à scénariser l'activité pédagogique d'une manière neutre (sans imposition d'une approche pédagogique) selon la métaphore théâtrale et selon trois niveaux (A, B et C)....

- le niveau A contient les éléments cœur du «méta-langage »,
- le niveau B permet l'utilisation de propriétés génériques et de conditions,
- le niveau C ajoute la possibilité de notifier les activités.

En d'autres termes, il est nécessaire de préciser les orientations et choix pédagogiques, les acteurs et rôles, les activités, les processus et les outils mis en œuvre.

3. Le graphe des épistèmes comme ressources et parcours pour le langage Python

Dans le cadre d'un projet connexe avec nos travaux, nous nous appuyerons sur des ressources brutes préconçues en langage Python désigné par le graphe des épistèmes.

Le graphe des épistèmes peut être vu comme une méta-ressource englobant toutes les notions nécessaires et dans la continuité d'un hyperlivre (Falquet et al., 2003). La construction et l'élaboration de ces épistèmes suivent des règles bien définies comme explicité dans (Ortiz, 2012a).

La construction du domaine de l'apprentissage du langage Python s'appuie sur trois concepts (Ortiz, 2012a) : l'unité épistémique, le graphe des matières et le rasoir d'Occam.

- Une unité épistémique ou épistème est un segment de connaissances de très petite dimension qui a été isolé du corpus de connaissances à exposer.
- Le graphe des matières définit les relations de prérequis entre deux épistèmes.
- Le principe de délimitation de chaque unité est le principe de parcimonie dit du « rasoir d'Occam ».

La notion de parcours, dans ce graphe des épistèmes, peut être vue comme un sous-graphe extrait du graphe complet. L'exemple du jeu « plus au moins » pourrait être un objectif à atteindre faisant intervenir un certain nombre de prérequis (et donc d'épistèmes). Être en mesure d'atteindre cet objectif consiste à parcourir la totalité des sommets du graphe en suivant les arrêtes du haut vers le bas. Mais il existe de nombreux parcours selon le modèle de l'apprenant, et c'est là que le principe de la différenciation est mis en évidence.

Les principaux bénéfices sont de :

- pouvoir prendre en compte des connaissances préalables de l'apprenant,
- limiter la charge cognitive,
- mieux contrôler la granularité et délimiter d'une façon précise un contenu et donc d'isoler une difficulté d'apprentissage qui peut être un facteur d'adaptativité,
- permettre l'extensibilité, peu sensible au facteur d'échelle, une extension par simple ajout d'unités à condition de respecter les contraintes de prérequis,
- rendre le contenu de l'unité plus consistant, celui-ci peut être modifié sans pour autant remettre en question la totalité du graphe,
- et autoréguler grâce à la planification et un contrôle.

Cependant, les inconvénients peuvent avoir un lien avec :

- le séquençement qui peut parfois conduire à l'atomisation, alors que l'apprentissage repose aussi sur le principe de l'intégration,
- la duplication de contenus avec le rasoir d'Occam,
- et l'édition et la modification qui peuvent s'avérer difficile à maintenir pour assurer une cohérence globale.

VIII. Conclusion

À travers les différents modèles de descriptions de ressources d'apprentissage et leur réutilisation pour pouvoir proposer des parcours d'apprentissage, nous pouvons formuler les principales remarques qui suivent.

Les ressources que nous visons en priorité sont des ressources de granularité forte ou de haut niveau. Il s'agit en l'occurrence de cours ou de manuels de référence, autrement dit des ressources que nous appelons complexes (Miled, 2011) assez similaires aux courseware. Ce type de ressource peut être vu comme étant lui-même découpé en sous unités. Ces types de ressources bien que nécessitant plus d'adaptation et de contextualisation, méritent d'être étudiés puisque ils requièrent un travail d'édition et de mise à jour collaborative (cf. la section III du chapitre 2 de la partie I).

De plus, le type de ressources dont nous disposons peut facilement se modéliser sous forme d'un graphe. Ainsi, à la différence d'une ressource simple, il s'agit d'une ressource composite mais constituée de micro-ressources homogènes appelées épistèmes ou unités épistémiques. Celles-ci correspondraient à des SCO ou s'en approcheraient dans le modèle SCORM, des Information Objects dans le modèle Learnativity ou encore aux RIO dans le modèle CISCO.

En ce qui concerne la relation entre traces et ressources, nous estimons que le suivi de l'apprenant peut s'appuyer sur les interactions avec les ressources en termes d'état, c'est pourquoi nous prévoyons d'utiliser les états fournis par ADL SCORM et d'ajouter un autre état plus spécifique à l'apprentissage de la programmation. Le modèle de ressources que nous tâcherons d'utiliser s'appuie sur une représentation à base de graphe (voir la section II du chapitre 2 de la partie II) En termes d'évaluation de la ressource d'apprentissage, nous tâcherons de nous limiter à certains critères issus du LORI (revoir la section VI de ce chapitre) et non pas la totalité pour la simple raison qu'il reste difficile à faire une méta-évaluation, en l'occurrence, évaluer le critère lui-même. Exemple : avec le critère de motivation, il paraît difficile de dire si une ressource participe réellement à la motivation, puisque cela reste assez variable et subjectif d'un utilisateur à un autre.

Un sous-ensemble d'unités peut constituer, selon notre approche, un parcours d'apprentissage. Ceci dit, au-delà de cette définition rudimentaire, nous tâchons de différencier deux types de parcours : un parcours recommandé ou prescrit par l'enseignant et/ou le tuteur et le parcours effectif réalisé par l'apprenant.

- Le parcours d'apprentissage recommandé :

Il est généralement une suite d'unités choisies à l'avance pour constituer une séquence permettant de répondre à un objectif particulier. Par exemple : pour pouvoir accomplir « afficher une séquence de nombre en langage Python », l'enseignant estime qu'il aura besoin de l'unité afficher et de l'unité boucle.

- Le parcours d'apprentissage effectif :

Il dépend des activités de l'utilisateur (souvent l'apprenant) et peut servir à alimenter son modèle dynamique d'apprentissage et à mieux cerner son profil, c'est la manière dont il construit le savoir à acquérir. Nous verrons dans la partie traces d'interactions qu'il est possible de lier les actions de l'apprenant par actions unitaires ou atomiques pour catégoriser son type de parcours.

Le parcours d'apprentissage comme le rappelle (Ernst, 2008) est orienté par le trinôme : prérequis, champ cognitif et objectifs pédagogiques. Nous pourrions même transposer ce point de vue pour un épistème conçu de cette manière :

- l'ensemble de prérequis de l'épistème (notions nécessaires ou utiles),
- l'épistème lui-même ou le concept central,
- et un exercice utilisant l'épistème en question qui sert ainsi comme objectif pédagogique.

Il convient également de tenir compte de la relation entre connaissances et tâches d'apprentissage. Certaines méthodes existantes ont été utilisées pour les EIAH comme OOHDM¹³, ou encore plus récemment SHDM¹⁴.

¹³ Object-Oriented Hypermedia Design Method

Nous pouvons nous demander maintenant comment exploiter ces ressources d'apprentissage. Comment les utiliser pour proposer des parcours, si possible différenciés, aux apprenants ? Quels types d'environnements numériques disponibles nous permettent d'accéder soit à des ressources brutes soit à des parcours organisés voire scénarisés ? Comment les réinvestir, les réguler ou les modifier ?

Chapitre 2 : Les environnements d'accès et d'édition de ressources numériques d'apprentissage

I.	INTRODUCTION	31
II.	LES ENVIRONNEMENTS D'ACCES AUX RESSOURCES.....	31
1.	<i>Les entrepôts de ressources d'apprentissage et les environnements généralistes : centraliser les ressources pour faciliter l'accès.....</i>	32
2.	<i>Les LMS</i>	33
3.	<i>Les environnements dédiés à l'apprentissage de la programmation.....</i>	34
III.	LA CONCEPTION ET L'EDITION COLLABORATIVE DE RESSOURCES.....	35
1.	<i>L'édition collaborative de modèles</i>	36
2.	<i>Le Cas MediaWiki.....</i>	36
3.	<i>Quelques outils de travail collaboratif (CSCW) et outils-auteurs pour la création de ressources</i>	36
4.	<i>Services pour l'accès aux ressources et activités collaboratives</i>	36
5.	<i>Plateformes pour l'accès et la création de ressources d'apprentissage</i>	37
IV.	CONCLUSION	40

I. Introduction

L'exploitation de ressources et de parcours d'apprentissage doit se faire d'une manière facilement accessible à l'utilisateur. Cette section a pour objectif d'introduire les environnements d'accès aux ressources. Quelle que soit leur nature, ces environnements, plutôt diversifiés, permettent de lire et d'utiliser les ressources d'apprentissage. Certains d'entre eux permettent même d'introduire des mécanismes de co-édition de ressources d'apprentissage. Nous présentons dans un premier lieu les environnements d'accès aux ressources d'apprentissage par extension, à travers les entrepôts de ressources, les environnements généralistes, spécifiques à certains domaines et les plateformes d'enseignements en ligne ou les Learning Management Systems (LMS).

II. Les environnements d'accès aux ressources

La plupart des environnements liés à l'utilisation des objets d'apprentissage¹⁵ et à leur co-édition¹⁶ permettent de partager les ressources et de suivre leur évolution selon un cycle de vie approprié. À titre d'exemple, la Figure 8 présente le cycle de vie de la fondation ARIADNE. Celui-ci comprend, entre autres, les phases d'enregistrement, d'alerte, de requête, de publication, de validation et de transformation. Ces environnements peuvent prendre la forme de plateformes dédiées à un domaine spécifique comme l'apprentissage du langage Python¹⁷, ou de plateformes d'apprentissage en ligne LMS¹⁸ de type Moodle, Blackboard, Claroline ou même encore de dépôts de ressources tels que ARIADNE (Joris, Klerkx et al., 2010b), MERLOT, LORNET ou encore NIME¹⁹. Ces types d'environnements regroupent bien souvent des ressources hétérogènes, isolées et rarement disposées d'une manière faisant ressortir une cartographie complète de leurs dépendances. Les sites dédiés à une thématique spécifique, s'ils apportent parfois un contenu exhaustif du domaine, ils ne s'inscrivent pas dans des objectifs d'aiguillage de l'apprenant et de retour sur la navigation.

¹⁵ Learning objects

¹⁶ www.Cnx.org

¹⁷ <http://www.py4inf.com>

¹⁸ Learning Management Systems

¹⁹ National Institute of Multimedia Education, <http://www.nime.ac.jp>

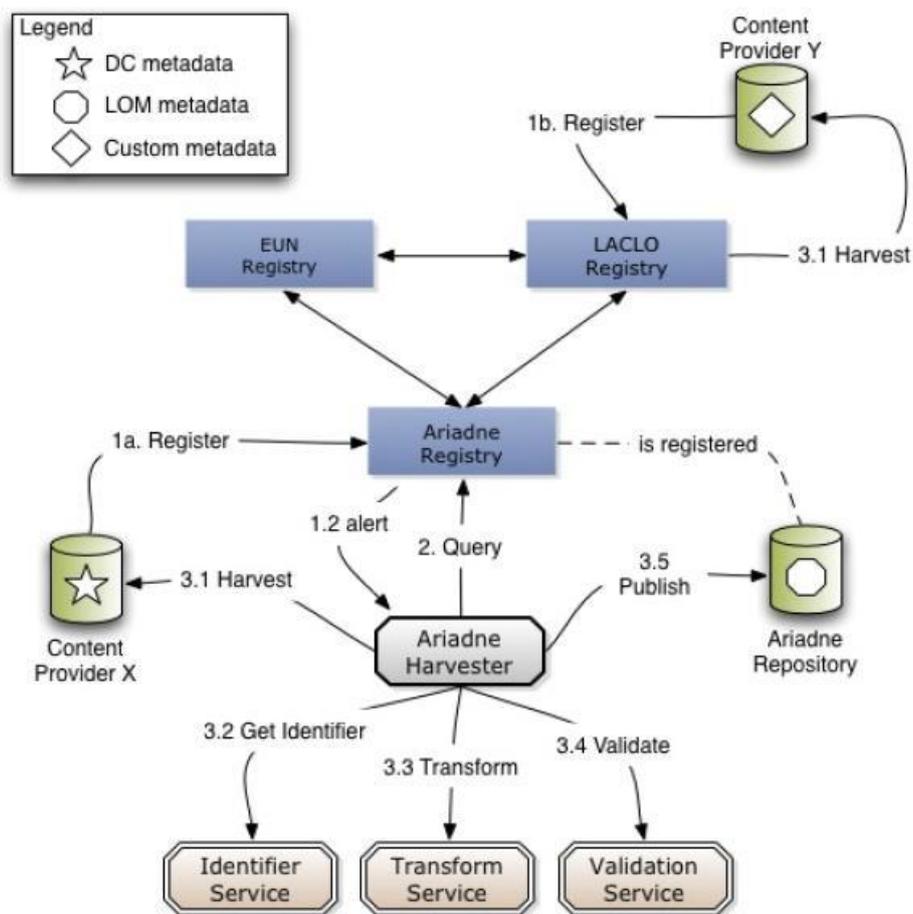


Figure 8 : cycle de vie selon l'approche ARIADNE

Les environnements décrits ici représentent des environnements d'accès aux ressources numériques d'apprentissages, sans forcément s'intéresser au processus de réception. Se déclinant en réservoirs pour ressources, LMS, ou en environnements dédiés, ils ont pour point commun leurs objectifs finaux, même s'il existe des spécificités propres à chacun.

1. Les entrepôts de ressources d'apprentissage et les environnements généralistes : centraliser les ressources pour faciliter l'accès

Les entrepôts de ressources rassemblent généralement une multitude d'objets d'apprentissage de nature hétérogène en terme de contenu disciplinaire (e.g. mathématique, langues, histoire...) et en termes de format (texte, audio, vidéo) et de granularité (cours, exercice, exemple...). Il existe d'ailleurs des recherches qui décrivent des typologies sur les LOR (McGreal, 2008).

L'une des fonctions remplies par les dépôts de ressources est de fédérer un nombre important de ressources d'apprentissage sur une même plateforme et à partir de différentes sources pour permettre l'accès et le partage. Deux approches sont utilisées : la première stipule que seules les métadonnées de chaque document numérique sont stockées et que le document en question est accessible grâce à son URL. C'est le cas de MERLOT (cf. Figure 9). Outre ses fonctionnalités de recherche, de consultation et de dépôt de ressources pluridisciplinaires, le portail américain propose une évaluation des contenus par des comités éditoriaux et par les utilisateurs (sous forme de commentaires et de description des utilisations). Remplissant des

fonctions équivalentes mais avec une deuxième approche consistant à stocker en même temps les métadonnées et le document numérique, l'entrepôt de ressources de la fondation européenne ARIADNE adopte dans la soumission de documents, les étapes d'identification, de transformation, de validation et de publication des objets d'apprentissage (Klerkx et al., 2010a). Certains entrepôts de ressources fournissent des mécanismes de collaboration en ligne dans le processus de création et d'évaluation des ressources, que ce soit par des outils auto-inclus ou d'autres outils évolutifs. Les actions permises sont ainsi de rassembler, désassembler et modifier des parties de contenus pour réassembler de nouvelles ressources.



Figure 9 : page d'accueil de MERLOT

D'autres environnements généralistes comme la plateforme de formation ouverte²⁰ : sous l'égide de l'UNESCO, permettent un accès collaboratif à de nombreuses ressources à licence ouverte (OER) dans différentes disciplines. La navigation peut se faire par type de ressource, par communauté ou par catégorie.

2. Les LMS

Les *Learning Management Systems*, ou plateformes d'apprentissage en ligne sont des plateformes similaires à des CMS²¹ mais qui offrent la possibilité de gérer un module de formation en exploitant des ressources d'apprentissage et des groupes d'utilisateurs. Différentes fonctionnalités (parfois couplées nativement) existent: des moyens de communication asynchrones (forums, wiki) ou synchrones (messagerie instantanée), des outils de suivi des apprenants. Il en existe sous format propriétaire comme Blackboard ou libre: Claroline, Moodle, Dokeos...)

²⁰ <http://otp.unesco-ci.org/>

²¹ Content Management Systems

Ils sont parfois couplés aux ENT (environnements numérique de travail) dans certains établissements. La majorité de ces plateformes se conforme aux normes d'indexations telles que LOM ou les normes d'exécution SCORM pour assurer l'interopérabilité.

Les LMS font encore l'objet de recherches par exemple avec la conférence annuelle Moodle Research conference²².

3. Les environnements dédiés à l'apprentissage de la programmation

Il existe une multitude d'environnements destinés à proposer des ressources et à en partager. Nous choisissons de présenter ceux qui sont relatifs à l'apprentissage de la programmation. On peut s'inspirer cependant d'environnements d'apprentissage des langues tels que Busuu²³. L'apprentissage des langues nous semble plutôt proche de l'apprentissage d'un langage de programmation.

- PythonLearn²⁴, ou encore learnpython²⁵ ont pour objectif de fournir une collection générique de ressources (principalement de l'université du Michigan) pour permettre un apprentissage selon le rythme de l'utilisateur. LearnPython apporte une interactivité supplémentaire en laissant la possibilité d'exécuter du code (pas seulement en Python).
- CodeAcademy²⁶: est un environnement interactif de programmation proposant plusieurs langages tels que Javascript, Python, Ruby ou encore PHP. Il intègre un interprète qui permet de vérifier si le code tapé est correct ou non. Il propose une communauté en incluant des fonctionnalités pour encourager les interactions entre membres. Hormis le mode « apprendre », cette plateforme peut également s'utiliser en mode « enseigner » et permet la création de contenus pour un langage cible (Figure 10).
- « Python in education »²⁷ qui peut être vu comme un hyperlivre mais selon une présentation linéaire,

²² <http://research.moodle.net/>

²³ <http://www.busuu.com/fr>

²⁴ <http://www.py4inf.com>

²⁵ <http://www.learnpython.org/>

²⁶ <http://www.codecademy.com/>

²⁷ <http://pentangle.net/python/>

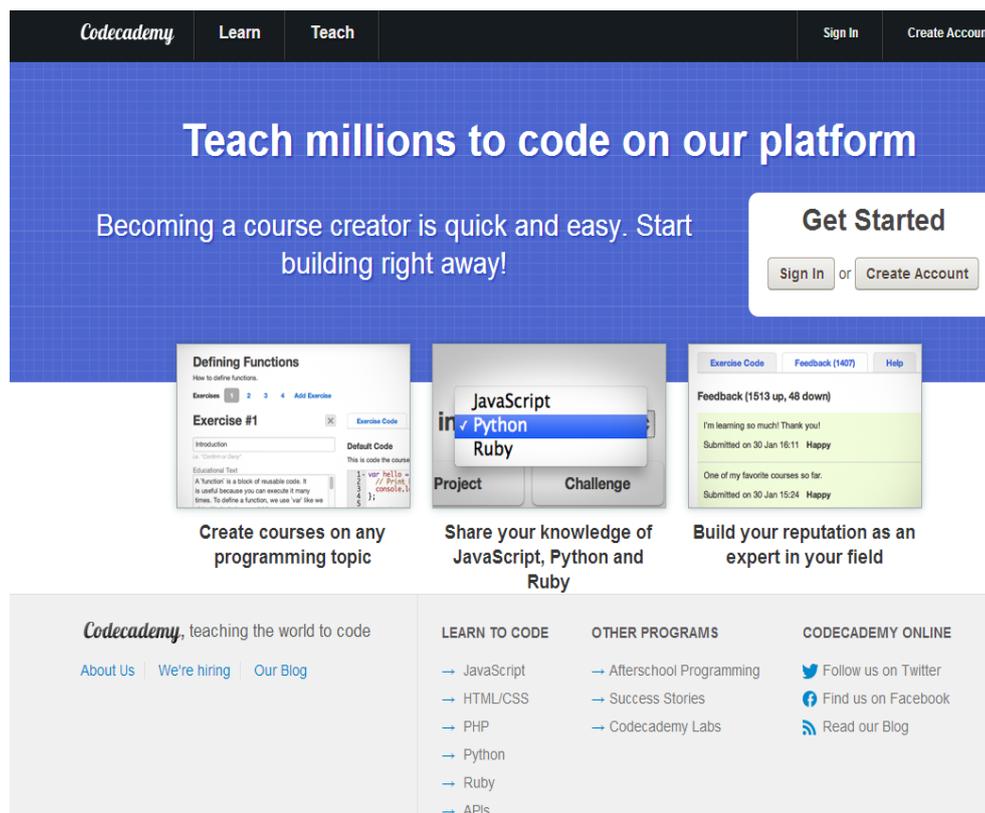


Figure 10 : Code Academy

Après avoir énuméré et analysé, de façon synthétique, des exemples d'environnements d'accès ressources, nous présentons d'autres dispositifs et approches permettant de prendre en compte l'aspect de la conception et de l'édition collaborative des ressources d'apprentissage.

III. La conception et l'édition collaborative de ressources

On attribue parfois le travail collaboratif ou coopératif au monde de l'entreprise. En effet, pour certaines tâches complexes, il est parfois plus judicieux de pouvoir diviser des tâches et de les distribuer aux différents acteurs concernés. (Heutte, 2011) souligne que « la distinction entre coopératif et collaboratif s'opère en distinguant les relations qu'entretient chaque individu avec les membres du groupe, sa responsabilité par rapport aux actions, sa capacité à influencer sur la définition et l'enchaînement des actions permettant d'atteindre l'objectif assigné au groupe. ».

De nombreux travaux de recherche s'intéressent aux CSCW²⁸ et aux CSCL²⁹ (des conférences annuelles sont organisées dans ce sens). Nous présentons le concept de collaboration à travers l'ingénierie dirigée par les modèles, le cas de MediaWiki et ses extensions permettant de soutenir ce processus ainsi que quelques outils de collaboration. Ensuite, nous explorons les

²⁸ Computer Supported Cooperative Work

²⁹ Computer Supported Collaborative Learning

architectures, les plateformes prenant en compte l'édition collaborative de ressources d'apprentissage.

1. L'édition collaborative de modèles

Dans le cas de l'ingénierie collaborative logicielle (collaborative software engineering), (Michaux, Blanc, Shapiro, & Sutra, 2011) utilisent un graphe pour modéliser les entités manipulées et proposent un modèle collaboratif pour gérer les questions d'édition de ce graphe (copie, fusion, suppression) et de détection de conflits. L'outil développé C-Praxis permet selon eux de passer à l'échelle.

L'ingénierie dirigée par les modèles peut être également profitables au domaine des EIAH, notamment sur des activités de scénarisation pédagogiques collaboratives (Vachet, 2006), incluant une phase d'opérationnalisation et d'instanciation ou coopératives (Nodenot, 2005). Ce type de processus peut « systématiser » le cycle de conception ou d'édition de scénarios pédagogiques.

2. Le Cas MediaWiki

Il s'agit d'un environnement collaboratif d'édition. Il se base sur des extensions modulaires. Media Wiki propose des extensions sémantiques avec semantic MediaWiki, Vote pour évaluer les ressources quelque soit leurs types. D'autres extensions permettent également d'évaluer des contenus notamment Ratings, Reader Feedback. L'édition semi-automatique de wiki peut également se faire avec l'extension AutoWikiBrowser. Ces aspects peuvent favoriser l'édition collaborative de ressources.

3. Quelques outils de travail collaboratif (CSCW) et outils-auteurs pour la création de ressources

Avant de traiter des environnements de conception et d'édition collaborative de ressources, il est utile de citer certains outils prioritairement utilisés en milieu d'entreprise tels que. FengOffice³⁰, Alfresco³¹, ou encore les collecticiels (ou groupware) avec PhpGroupware, Egroupware.

Par ailleurs, il existe une multitude d'outils auteurs pour générer du contenu pédagogique. Qu'il s'agisse de cours, d'exercices, et plus spécifiquement de QCM, la plupart doivent gérer certaines normes pour être exécutables indépendamment de la plateforme (LMS) : e.g. la norme IMS-QTI pour les questions à choix multiples.

De plus, les plateformes d'apprentissage en ligne (LMS) proposent un environnement WYSIWYG³² pour créer, éditer et lire des contenus.

4. Services pour l'accès aux ressources et activités collaboratives

De nombreux travaux de recherche décrivent des architectures logicielles dédiées à la gestion de ressources d'apprentissage. Généralement, les fonctionnalités habituelles attendues sont :

³⁰ www.fengoffice.com

³¹ <http://www.alfresco.com/>

³² What You See is What You Get : désigne généralement une interface qui permet de composer visuellement le résultat attendu

la recherche, l'accès, l'indexation et l'annotation des contenus. Certaines utilisent ces fonctionnalités d'une manière indépendante sous forme de services (Catteau, Marquié, Vidal, & Broisin, 2009b) pour faire le lien entre plateformes d'apprentissage et viviers de connaissances. En marge de ces services, il semble important de tenir compte de deux aspects :

- d'une part de la dimension collaborative dans le processus d'évolution des ressources. Cela implique, ce qui est valable dans chaque activité collaborative, le traçage et la gestion des interactions (Lonchamp, 2008),
- et de l'autre, la conscience des modifications (change awareness) (Tam & Greenberg, 2006a) permettant de mieux cerner les changements qui se sont réalisés au cours du cycle de vie de la ressource.

Plus généralement, comment prendre en considération l'aspect validation d'une version donnée d'une ressource ? La validation peut se faire d'une manière informelle en regardant par exemple les commentaires des différents participants ou autres acteurs. Il est peut-être utile de proposer une approche basée sur des règles de validations individuelles ou globales ; c'est-à-dire par des mécanismes de validation reposant en partie sur la communauté et une autre sur une validation automatique.

Certaines approches comme l'utilisation de questions pour produire des ressources, ou de FAQ dynamique (Ng'ambi & Brown, 2009), ou encore les systèmes de votes sont parfois utiles pour aider à la décision dans un processus collaboratif.

5. Plateformes pour l'accès et la création de ressources d'apprentissage

Nous présentons ici un aperçu d'exemples de plateformes en essayant de mettre en évidence certaines de leurs spécificités.

- COMES³³ se décrit comme « un atelier collaboratif de conception mutualisée de ressources d'apprentissage personnalisables ». Cette plateforme permet entre autres de publier, de concevoir selon des « grains³⁴ » (cours, activités, étude de cas...) de personnaliser, de mutualiser et d'adapter du contenu selon des parcours spécifiques.
- Connexions³⁵ (Figure 11): ici, les acteurs peuvent créer des contenus et collaborer entre eux, les tuteurs peuvent construire et partager des collections personnalisées et les apprenants accèdent par la recherche et l'exploration du contenu visé selon la licence CC-BY 3.0³⁶. Les types de ressources présentés sont les modules, les collections (ensemble de modules) et les lentilles (objets de focus par des organismes ou certaines communautés). Ces dernières peuvent être homologuées, affiliées ou favorites ou sur des listes de membres. L'ensemble des fonctionnalités disponibles permet de créer un nouveau module (ou nouvelle collection), chercher pour éditer un contenu publié, ajouter aux favoris (en interne ou en externe dans des gestionnaires de favoris de type del.icio.us), être alerté des changements par RSS voire de tracer l'avancement ou revenir à une version où la dernière lecture s'est arrêtée.

³³ <http://comes.c2imes.org/>

³⁴ Voir la notion de niveaux de granularité d'une ressource

³⁵ www.Cnx.org

³⁶ Creative commons. Voir <http://creativecommons.org/> et <http://creativecommons.org/licenses/by/3.0/fr/>

Si les différents rôles ou profils d'utilisation sont les auteurs, les mainteneurs, les titulaires du droit, l'éditeur ou les traducteurs, il est utile de constater dans ce genre d'environnement la manière de gérer les problèmes de droits d'auteurs. Un contributeur qui souhaiterait dériver une copie d'une ressource et apporter des modifications doit le notifier à l'auteur original ; dans tous les cas, l'auteur ainsi que le(s) contributeur(s) deviennent automatiquement co-auteurs. Un contributeur peut demander à avoir des privilèges en demandant une requête à l'auteur et chercher de nouveaux collaborateurs pour créer un workgroup. L'espace personnel pour un utilisateur enregistré est désigné par MyCNXHome, mais il est possible de lire, télécharger et partager en mode invité.

Figure 11 : page d'accueil de Connexions

- P2P university³⁷ : même si la particularité de cette plateforme réside dans l'apprentissage par les pairs, on distingue trois types d'acteurs : les contributeurs, les mentors (plutôt tuteurs) et les apprenants eux-mêmes.
- Khanacademy³⁸ : offre une panoplie de ressources (fortement orientées vers les vidéos) dans un grand nombre de disciplines (voir Figure 12). En regardant ceux qui

³⁷ <http://p2pu.org/>

³⁸ www.khanacademy.org

s'intéressent à la programmation, l'apprenant peut non seulement parcourir des rappels de cours, mais aussi programmer et regarder les programmes produits par les autres apprenants. Dans notre travail, nous nous sommes intéressés à la question du partage et de la co-production des ressources dans la communauté des enseignants, mais nous n'avons pas envisagé le partage de productions des apprenants.

The screenshot shows the Khan Academy interface for the 'Intro to Variables' lesson. On the left, there is a navigation menu with categories like 'COMPUTER PROGRAMMING' and 'DRAWING AND ANIMATION'. The main content area displays the lesson title and a code editor with the following JavaScript code:

```
noStroke();
// face
fill(255, 255, 0);
ellipse(202, 208, 300, 300);
// eyes
fill(0, 0, 0);
ellipse(157, 151, 40, 40);
ellipse(304, 142, 40, 40);
// mouth
fill(255, 0, 0);
ellipse(257, 240, 120, 136);
```

To the right of the code editor is a visual representation of a yellow smiley face with two black eyes and a red mouth. Below the code editor, there are buttons for 'Loading audio...', 'Save as a spin-off', and 'Restart'. At the bottom, there is a 'Questions' section with a search bar and a question: 'Can I get Javascript for myself and create a game?'.

Figure 12 : un exemple d'activité (sur les variables) dans Khanacademy

- I2G Intergeo³⁹ : c'est une plateforme créée par le projet Intergeo, co-financé par le programme eContent Plus de la commission Européenne et par d'autres partenaires, autour de la géométrie « interopérable ». Elle permet entre autres le partage du matériel pédagogique libre et interactif, donne un historique des ressources et un classement des résultats de recherche par ceux des évaluations ou leurs poids.
- Le réseau EdReNe⁴⁰ : ce réseau thématique a pour but principal d'impliquer les concepteurs de matériel pédagogique dans les réservoirs de ressources basés sur le web pour permettre de partager, de développer et d'apporter des pratiques, des expériences et de nouvelles façons d'organiser ces entrepôts de ressources.
- CodeWitz⁴¹ : c'est un environnement qui procure des outils d'aide à la compréhension et d'aide à l'enseignement de la programmation par des mécanismes de visualisation (Ala-Mutka, 2012) et sous forme d'objets d'apprentissage.

³⁹ <http://i2geo.net>

⁴⁰ <http://edrene.org>

⁴¹ <http://www.codewitz.net/>

- Le projet SFoDEM⁴² (Suivi de Formation à Distance pour les Enseignants de Mathématiques), qui s'est déroulée de septembre 2000 à juin 2006 dans l'académie de Montpellier, s'intéresse à la conception collaborative de ressources.

D'autres projets ou initiatives méritent également d'être cités, Sésamath⁴³, Talisapire⁴⁴ qui procure des solutions propriétaires pour la transformation de ressources et d'adaptation de format de présentation et gestion de licences. Wikieducator⁴⁵ est une communauté qui permet de développer du contenu libre (OER) utilisé avec eXe⁴⁶ ou sur tout autre LMS.

IV. Conclusion

À travers ce chapitre, nous avons pu identifier une typologie des différents environnements d'accès aux ressources et aux parcours. Nous avons passé en revue quelques environnements (environnements généralistes, LMS, environnements pour l'apprentissage d'un domaine spécifique). D'autres environnements basés sur le web intègrent des possibilités d'édition collaborative. Cela peut représenter des pistes relatives aux démarches qui nous semblent potentiellement exploitables et transposables pour une approche d'édition collaborative de ressources complexes (sur le langage Python et dans d'autres disciplines de façon plus générale).

Les questions d'édition collaborative peuvent également intégrer les questions liées :

- au support de multiples facettes de ressources comme dans (Hernandez et al., 2009) ;
- à la prise en compte du domaine d'application ou plus globalement la question de la réutilisation dans un autre domaine contexte (Klerkx et al., 2010a) (Littlejohn, 2003) ;
- au regard porté sur les opérations de base : scission, clonage (ou copie), déclinaisons, unité épistémique contient, contraintes de prérequis entre les unités et leur édition;
- aux parcours obligatoires et fortement recommandés , faire co-exister ou faire converger une version unique, ceci amène la question de ressources vivantes (Joab, Guin, & Trouche, 2003) ;
- à l'adoption de bonnes pratiques de co-écriture avec l'utilisation de formes génériques ou de modèles ouverts qui sont appelés à être développés, ce qui est le cas des cours comme « graines » (DePaula et al., 2001) ;
- et au processus de validation de l'écriture (avec les acteurs impliqués), ainsi que les mécanismes pour assurer une robustesse.

Notre proposition visera à soutenir l'enseignement et l'apprentissage d'un thème complexe faisant appel à de nombreuses ressources interdépendantes. Nous avons prévu une première implémentation dans le cas d'un environnement dédié à l'apprentissage de la programmation en Python.

Au-delà d'une utilisation conventionnelle, quels moyens de représentation préconiser pour les ressources « complexes » ? Quels outils peuvent être exploités pour matérialiser l'avancement de l'apprenant et son propre parcours ? Comment visualiser les ressources d'apprentissage et

⁴² http://sfodem.um2.fr/IntroSfodem/IntroSfodem_HTML/Pages/intro_ressources_peda_1.html

⁴³ <http://www.sesamath.net/>

⁴⁴ <http://talisaspire.com>

⁴⁵ <http://fr.wikieducator.org/Accueil>

⁴⁶ <http://www.wikieducator.org/EXe>

PARTIE I : ANCRAGES THÉORIQUES, REVUES DE LA LITTÉRATURE ET POSITIONNEMENT DES TRAVAUX

Chapitre 2 : Les environnements d'accès et d'édition de ressources numériques d'apprentissage

aider à utiliser un parcours donné ? Comment donner une forme interactive au graphe des épistèmes ?

Nous verrons plus en détail dans la partie 2, quelques orientations relatives à ces questions.

Chapitre 3 : Visualisation, navigation dans les ressources et traçage d'activités

I.	INTRODUCTION	43
II.	LE CHAMP DE LA VISUALISATION ET LA NAVIGATION	43
1.	<i>La visualisation de l'information et la navigation dans les graphes</i>	43
2.	<i>La visualisation de l'information et l'aide à la navigation dans les EIAH</i>	44
3.	<i>Quelques outils et interfaces de visualisation de graphes</i>	45
III.	LES TRACES D'INTERACTIONS.....	49
1.	<i>La notion de trace en EIAH</i>	49
2.	<i>Traces et indicateurs en EIAH</i>	49
3.	<i>Objectifs et finalités</i>	49
4.	<i>Traces primitives et traces modélisées</i>	50
5.	<i>Indicateurs sur les traces</i>	51
6.	<i>L'expérience tracée et l'acquisition dynamique de connaissances</i>	51
7.	<i>Les traces déclaratives</i>	51
8.	<i>Liens entre traces d'activité et traces déclaratives</i>	51
IV.	CONCLUSION	52

I. Introduction

L'accès aux ressources et aux parcours d'apprentissage quel que soit l'environnement d'accès (LMS, plateforme spécifique, voire dépôt de ressources) ne peut pas omettre les questions de visualisation et de navigation. En effet, la manière de présenter et de mettre en forme les contenus peut avoir une incidence sur la manière dont les différents utilisateurs perçoivent et opèrent l'accès à la connaissance. Parallèlement, la navigation et la collecte des actions des utilisateurs dans ces ressources doivent être également prises en compte dans la conception, la régulation ou la réingénierie de systèmes d'apprentissage en ligne.

Nous présentons le champ de la visualisation et la navigation d'une façon générale, à l'aide de graphes dynamiques et ses applications dans les EIAH. Nous corrélons ses aspects avec la nécessité du traçage des activités en nous focalisant sur les paradigmes existants, pour définir les traces, leurs indicateurs et leurs adaptations. Nous ouvrirons comme perspective l'utilisation de traces déclaratives (auprès des utilisateurs) en complément des traces informatiques recueillies.

II. Le champ de la visualisation et la navigation

Nous introduisons ici la notion de visualisation de l'information et son utilité à travers les graphes. Avant de donner des exemples d'outils de visualisation des graphes, nous montrons l'intérêt de ces techniques de visualisation pour l'aide à la navigation et plus généralement pour les EIAH.

1. La visualisation de l'information et la navigation dans les graphes

La question de la recherche ou de la sélection de l'information devient d'autant plus importante que la quantité d'information croît. La visualisation interactive des graphes peut résoudre certains problèmes d'accès et de présentation surtout si le graphe devient de plus en plus volumineux. Elle permet également de mieux identifier les dépendances entre les nœuds. Différentes techniques de visualisation et mécanismes de navigation et d'interaction sous-jacents existent (Herman, Melançon, & Marshall, 2000). L'exploration de réseaux complexes peut se faire à l'aide de mécanisme de visualisation dynamique comme le suggère Visual Complexity⁴⁷.

Si la navigation dans les graphes permet de visualiser l'ensemble du graphe structurel, elle doit permettre en particulier de faire des opérations d'agrandissement/réduction, de déplacement panoramique et de focalisation sur une zone de prédilection consistant à mettre l'accent sur une région ou un sous-ensemble de nœuds. Toutes ces opérations doivent être garanties avec une certaine facilité de navigation ou navigabilité pour l'utilisateur. Bien qu'il existe de nombreux outils de visualisation de graphes, nous ne nous intéressons qu'à ceux qui sont interactifs (autorisant les opérations citées plus haut pour s'adapter au profil de l'utilisateur et à sa quête), navigables (servant de support pour la navigation dans notre ensemble de ressources).

Pour ce qui est des différentes mises en formes (graph layout), on peut retenir celles qui se basent sur :

⁴⁷ <http://www.visualcomplexity.com/vc/>

- les *lois des forces* (Force Directed) : tous les nœuds sont en interaction selon la loi physique des forces. La visualisation peut être statique ou animée, le mouvement permettant de mieux appréhender les dépendances complexes,
- les *arbres hyperboliques* (HyperTree),
- les *graphes radiaux* (Rgraph) come illustré dans la Figure 13,
- ainsi que les *arbres à tiroirs* (Treemap visible sur la Figure 14)

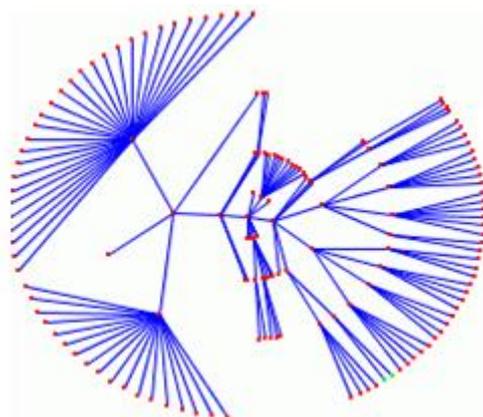


Figure 13 : graphes radiaux

On peut citer comme exemple le site <http://newsmap.jp/> qui utilise cette disposition tout en focalisant les thématiques prioritaires par ordre de grandeur.



Figure 14 : Treemaps

2. La visualisation de l'information et l'aide à la navigation dans les EIAH

Une question peut se poser dès lors : dans quelle mesure peut-on appliquer ces techniques de visualisation dans le champ des ressources d'apprentissage ?

Le champ de la visualisation de l'information a déjà été introduit dans les EIAH, notamment sur des questions de visualisation des relations entre les objets d'apprentissage (Catteau et al., 2009b; Catteau et al., 2007), sur les thématiques de visualisation de profil d'apprenant (Ginon & Jean Daubias, 2012) ; (Jean-Daubias, Ginon, & Lefevre, 2011), voire sur des questions d'aide à l'accès et à la recherche de ressources d'apprentissage déposées dans des réservoirs

(Klerkx et al., 2004). Certains travaux soulignent les limites de l'accès et de la recherche des ressources par des formulaires (Klerkx et al., 2005) et proposent des stratégies basées sur une visualisation et une navigation interactive.

Certaines recherches ont également utilisé les graphes dynamiques pour l'apprentissage et le développement du matériel pédagogique (Hadzilacos, Kalles, Karaiskakis, & Pouliopoulou, 2009).

En ce qui concerne la présente recherche, nous comptons utiliser les types de visualisation ForceDirected, HyperTree et RGraph. D'autres études ont montré qu'il y avait un apport de la visualisation des graphes dans la représentation et la fouille de données (Auber et al., 2007). Plus spécifiquement, la visualisation interactive aide l'utilisateur à interagir sur la carte et à comprendre la structure du graphe à travers sa navigation (Seifi, Guillaume, Latapy, & Le Grand, 2012). Nous estimons alors que ce genre de visualisation interactive, par rapport à une visualisation figée ou statique, peut aider l'utilisateur à choisir la ressource et à mieux naviguer dans un graphe de ressources, sans forcément avoir besoin de connaître leur intitulé pour les exploiter. Cette proposition constitue une alternative à une présentation linéaire (type tables des matières) qui ne peut faire état de la richesse des relations entre les différentes ressource ou à la recherche par mots-clés qui, dans le cas d'un débutant en programmation par exemple, nécessite de connaître le vocabulaire alors qu'il est justement en cours d'apprentissage. Par ailleurs, le nombre des concepts, mots-clés et donc ressources à référencer est tel, qu'une présentation exhaustive s'accommode mal des espaces restreints qu'offrent les écrans numériques.

3. Quelques outils et interfaces de visualisation de graphes

Nous avons étudié quelques bibliothèques et outils pour la visualisation de graphes dynamiques. Nous dégageons les principales caractéristiques dans le Tableau 1.

- **Graphviz**, framework PEAR

La classe Image_Graphviz permet la création et la visualisation (côté serveur PHP) de graphes orientés et non orientés s'appuyant sur les outils Graphviz d'AT&T. Toutefois, les possibilités d'interaction dynamique (glisser, déposer) ne sont pas directement gérées côté serveur.

- [Arbor.js](http://arborjs.org/)

Il s'agit d'une visualisation avancée de graphes (Figure 15) gérant simultanément l'animation et l'interactivité. <http://arborjs.org/> Elle utilise la librairie JQuery et les « web workers » permettant l'exécution de scripts JavaScript concurrents (un peu comme les multi-threads)

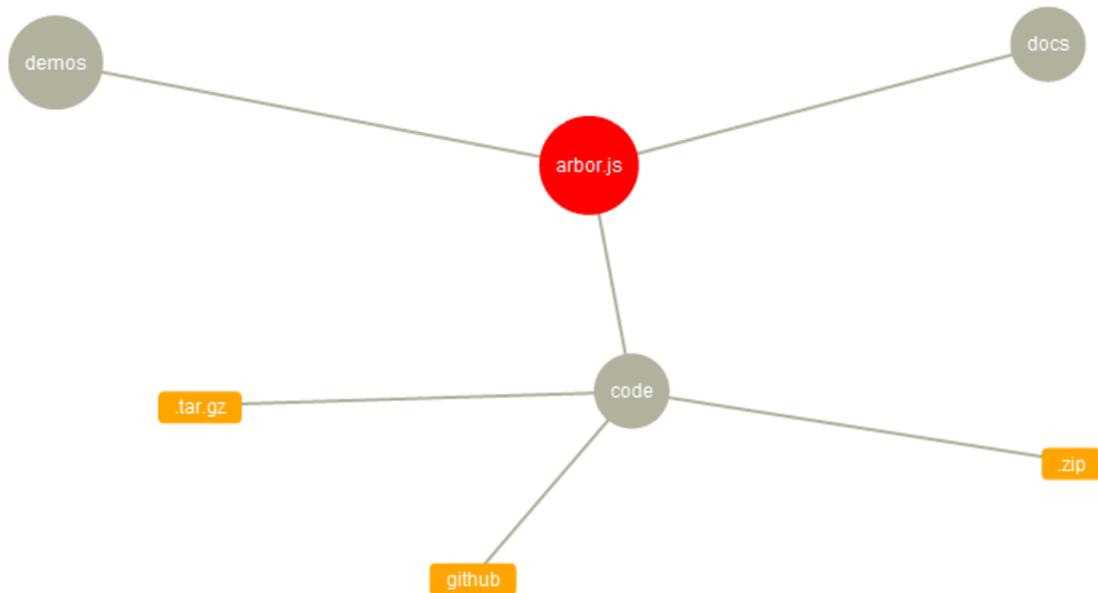


Figure 15 : Arbor.js

- [Canviz JS](http://code.google.com/p/canviz/) renderer for Graphviz graphs <http://code.google.com/p/canviz/> :

Canviz (Figure 16) est une librairie JavaScript (rendu xdot) servant à représenter les graphes de type Graphviz dans un navigateur web. Permettant la génération et l'envoi d'images Bitmap, le serveur a besoin uniquement des données xdot, ce qui est plus rapide que de générer des images bitmaps. L'utilisateur peut ajuster la taille sans faire appel au serveur. Néanmoins certaines fonctionnalités non encore gérées pourraient être utiles :

- les interactions avec le mouvement de la souris (mouse-over and mouse-out) permettant le changement d'attributs (couleur, taille, etc...) sur des portions du graphe
- les animations décrivant le changement d'états dans le graphe
- une interface déplaçable (à la manière de GoogleMaps) pour l'exploration de grands graphes.

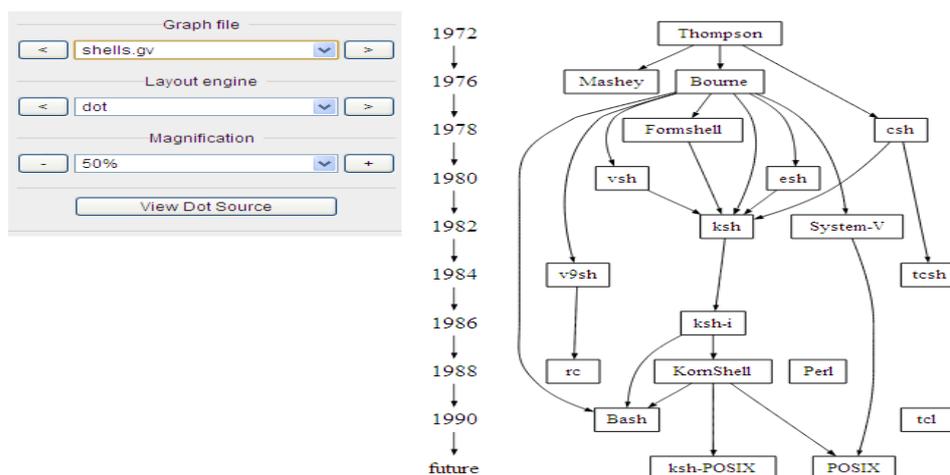


Figure 16 : Canviz

- **[Flare](#)**
Flare est une librairie ActionScript (pour le lecteur Flash) permettant une visualisation et une navigation interactive.
- **[JavaScript Canvas Hyperbolic Tree](#)**
Conçu pour les arbres hyperboliques, il sert à afficher un flot important de données interreliées. Ce projet a été intégré avec JavaScript InfoVis Toolkit.
- **[JavaScript InfoVis Toolkit](#)** est un outil pour l'agencement de graphes gérant plusieurs cas d'utilisations.
- **[JS Graph It](#)** l'intérêt de cette librairie en JavaScript, est l'utilisation rapide de la représentation (Figure 17) sans l'ajout de code JavaScript supplémentaire.
- **[Dhoston Springy](#)** Spring.js est un graphe basé sur l'algorithme des forces <https://github.com/markusgattol/springy>

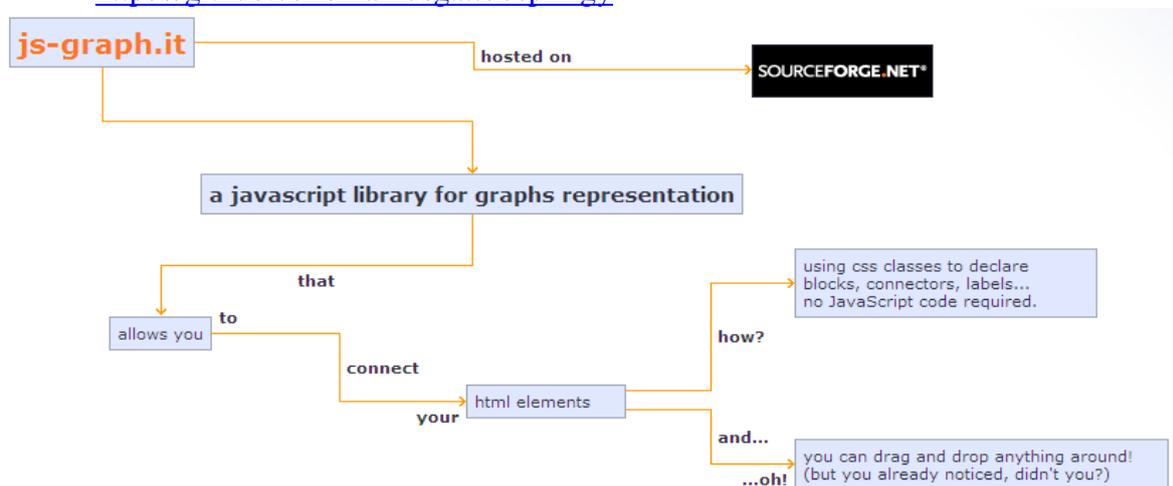


Figure 17 : Js-graph.it

Parmi les autres projets similaires existants, nous pouvons citer :

- [jsPlumb](#) : plug-in en JQuery pour la création de graphes interconnectés ;
- [NodeBox](#) Python Graph Visualization ;
- Jsviz <http://www.jsviz.org/blog/> <http://caitlingannon.com/2007/09/21/jsvis-javascript-visualization-software/>;
- OpenJacob draw2D <http://www.draw2d.org/draw2d/> (en JavaScript);
- <http://prefuse.org/> (en swing Java) ;
- <http://www.graphdracula.net/> (en JavaScript) ;
- <http://www.anchor modeling.com/modeler/>

En ce qui concerne le site Pearltrees⁴⁸, il ne s'agit pas véritablement d'une librairie de visualisation mais plutôt d'un site qui permet de partager des signets liés à des contenus web. La particularité de ce site réside dans la navigation qui se fait selon des *arbres à perles*

⁴⁸ www.pearltrees.com

(*pearltrees*). Nous avons estimé judicieux de citer cette plateforme pour cet aspect de navigation dynamique.



Figure 18 : Pearltrees

Le survol d'une « perle » affiche la page principale du site cible (cf. Figure 18). Le clic sur l'un des nœuds (perle) fait apparaître les sous-nœuds. L'utilisateur peut créer soit une perle (signet) soit un arbre à perle regroupant un ensemble de perles. La sélection de l'un des arbres à perles fait disparaître les autres perles et ne garde que les perles qui y sont contenues. La possibilité de se déplacer et de zoomer dans cette arborescence (par drag and drop) facilite la visualisation de l'ensemble mais permet également d'avoir une vision locale parfois nécessaire.

Voici à présent un comparatif de quelques outils étudiés (Tableau 1) identifiés les plus adéquats à notre cas de recherche :

Outil/solution/librairie	Technologie utilisée	Installation et intégration (1 à 5)	Interactivité (1 à 5)	Possibilité de chargement d'un fichier externe pour le graphe (oui/non)	Richesse de la modalité des visualisations (1 à 5)
JS Graph It	JavaScript	1	3	Non	1
Flare	ActionScript/Flash	1	4	Non	4
PEAR/Image_Graphviz	PHP	2	1	Oui	1
Canviz	JavaScript	1	1	Oui	1
NodeBox	Python (support MacOS X)	2	3	Non	2
Prefuse	Java(Swing)	2	3	Non	3
Arbor	JavaScript/ web workers	1	4	Non	3
JIT	JavaScript	2	4	Oui	4

Tableau 1 : comparatif de quelques bibliothèques de visualisation dynamique

La valeur à 1 désigne la note la plus basse, tandis que la valeur 5 désigne la note la plus haute. Le graphe devant être recalculable dynamiquement, le fait de pouvoir charger sa structure dans un fichier externe rend la tâche plus simple. Parmi les trois candidats restants (PEAR/Image_Graphviz, Canviz et JIT), JIT est celui qui offre les meilleures performances en termes d'interactivité et de richesse de mode de visualisation. C'est donc l'outil JIT que

nous retiendrons comme élément de base à notre système de visualisation et de navigation. Le point que nous avons estimé déterminant hormis les plutôt bonnes performances, les richesses des bibliothèques correspondantes, est que la documentation est massive et il existe même une communauté (sous forme de Google groups) qui permette d'interagir. Ceci est primordial pour pouvoir l'exploiter au maximum (voir dans la partie 2, la section II du chapitre 4).

III. Les traces d'interactions

Nous définissons en premier lieu le concept de trace, nous explicitons la notion d'indicateurs ensuite, les objectifs et les enjeux correspondants. Nous illustrerons nos propos à l'aide de travaux plus récents comme ceux effectués sur les traces modélisés et l'expérience tracée.

1. La notion de trace en EIAH

Il n'y a pas une unique façon de définir la trace mais la plupart des définitions s'accordent à faire intervenir la notion temporelle des activités de l'apprenant. Ainsi selon (Djouad, Mille, Reffay, & Benmohammed, 2010), « une trace est une collection d'observés temporellement situés. » Pour (Cram, Jouvin, & Mille, 2007) la trace d'interactions serait « tout objet informatique dans lequel s'accumulent des données à propos des interactions entre un système informatique et son utilisateur ».

La trace peut être catégorisée de plusieurs manières dépendant du cycle où elle intervient. Il est utile ici de mentionner que le cycle de l'évolution de la trace peut être arbitraire et modulable selon les besoins des acteurs impliqués (concepteur, tuteur ou apprenant). Ce cycle peut prendre racine dans l'étape de collecte, où l'on désigne généralement ces traces comme *brutes* ou premières et qui sont non directement exploitables).

2. Traces et indicateurs en EIAH

Ces traces sont éventuellement transformées selon certains modèles (par exemple M-Trace) ou encore selon des indicateurs prédéfinis (orientés tâche ou non), selon l'approche FORMID (Guéraud, Adam, Pernin, Calvary, & David, 2004), pour être analysées et exploitées par la suite, ou encore à partir d'indicateurs portant sur la qualité générale de la production de l'apprenant, par exemple sur les codes sources soumis, s'il s'agit de programmation (HOP3X).

Outre les questions relatives à leurs utilisations et exploitations, il convient de mentionner des défis à relever par rapport à l'exploitation des traces dans des environnements technologiques plus récents tels que le *Cloud* (Chaabouni, Miled, & Laroussi, 2013).

3. Objectifs et finalités

Dans la littérature des EIAH, le recours aux mécanismes de traces peut avoir comme finalité la régulation ou l'auto-régulation de l'apprentissage (Heraud, Marty, France, & Carron, 2005), la modélisation et le profilage de l'apprenant, la personnalisation et l'adaptation de l'apprentissage (Brusilovsky, 1999)(Brusilovsky, 2001) ou encore la réingénierie des EIAH (Choquet & Iksal, 2007a). Certaines Plateformes comme PSCL DataShop⁴⁹, MULCE⁵⁰ ou

⁴⁹ <https://pslclatashop.web.cmu.edu/>

⁵⁰ <http://mulce-doc.univ-bpclermont.fr/>

encore plus récemment UnderTracks⁵¹ ont pour but de partager et de proposer un accès aux corpus des traces aux chercheurs en EIAH. Une des approches d'analyse des traces et de sa mise en œuvre c'est l'utilisation de l'approche par indicateurs.

4. Traces primitives et traces modélisées

Généralement, il est communément admis que le cycle de fonctionnement d'un EIAH, passe par la conception (1), le déploiement ou l'installation (2), l'exécution(3) et l'analyse (4). C'est dans cette dernière étape qu'il peut parfois nécessiter de transformer les traces et les éléments contextuels, on parle alors de traces modélisée à partir d'une trace brute ou primitive. Le modèle de traitement de traces *CSE* (Pernin, 2005) se base sur (1) la Collecte des traces primaires issues d'une situation d'apprentissage, (2) la Structuration des traces et le calcul d'indicateurs pédagogiques, (3) l'Exploitation des indicateurs et (4) la Régulation (ou adaptation) en fonction du *feedback* retourné. Quant aux indicateurs pédagogiques, ils visent à assister le tuteur dans la régulation et la réingénierie de son scénario pédagogique. D'autres systèmes, tel que le système à base de Traces (SBT) exploitent les traces modélisées (Djouad, Settouti, Prié, Reffay, & Mille, 2010) et opèrent des transformations de fusion, de sélection ou encore de réécriture comme dans la Figure 19. Une trace modélisée dans un SBT, comme présentée dans (Djouad, Mille, Reffay, & Benmohamed, 2009) est décrite par un modèle de trace et d'instance de ce modèle, où chaque instance est située dans le temps. Selon (Clauzel, Sehaba, & Prié, 2009) une trace modélisée est associée à un modèle de trace. Celui-ci peut être considéré comme une ontologie formelle décrivant le vocabulaire de la trace. Une des exploitations de la trace modélisée est visible sur l'approche MUsETTE (Modéliser les Utilisations et les Tâches pour Tracer l'Expérience) (Prié, 2006).

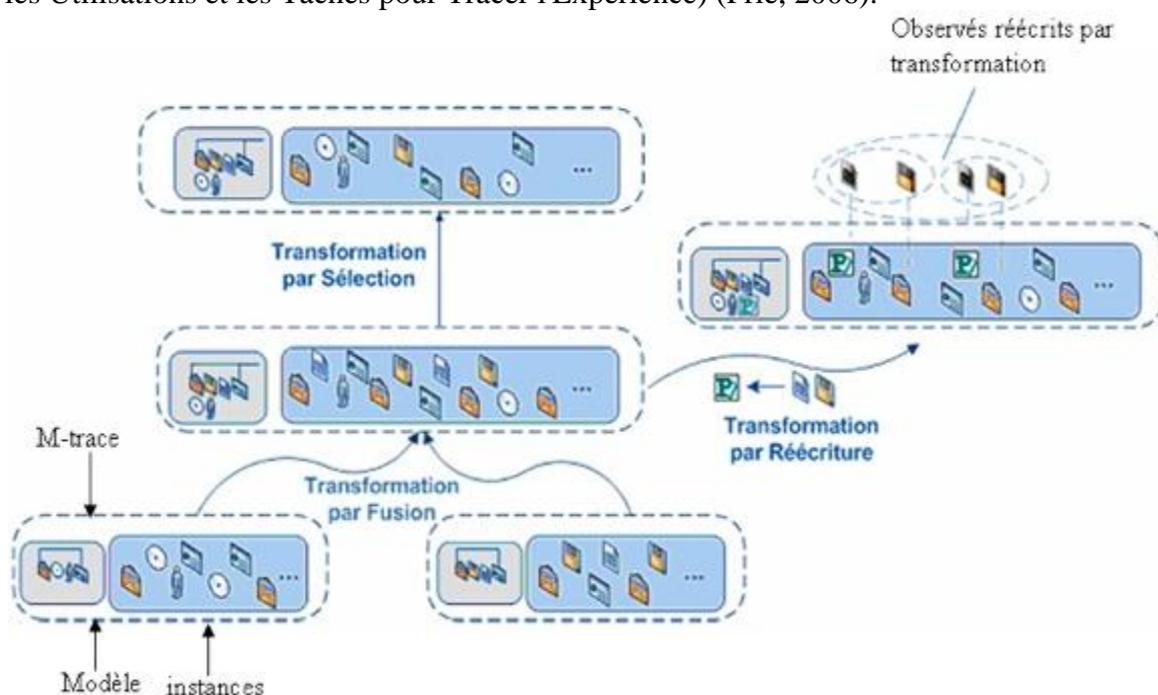


Figure 19 : transformation de M-traces (Djouad et al., 2009)

⁵¹ <https://undertracks.imag.fr/>

5. Indicateurs sur les traces

Les indicateurs d'analyse des interactions peuvent être définies selon (Dimitracopoulou, 2004) comme des « variables interprétées prenant en compte l'activité de l'apprenant, le profil des apprenants (si l'activité le permet) et le contexte d'interaction ».

L'ingénierie des indicateurs vise à classer ceux-ci selon leurs types : cognitifs, sociaux et affectif, ou selon leur niveau de calcul (haut ou bas niveau) dépendant du degré de leur interprétabilité. Ainsi, un clic peut être peu significatif en termes de connaissances utiles. À l'opposé, une trace renseignant sur l'ouverture de tel item peut être plus parlante. D'autres classifications, peuvent concerner, la fenêtre d'observation par exemple selon que l'on cherche à tirer des informations selon une session complète, ou bien selon une sonde temporelle précise ou encore par contenu (ou par exercice).

L'aspect de réutilisation et de capitalisation des indicateurs a également été traité notamment avec les patrons d'indicateurs utilisant par exemple le langage UTL (Choquet & Iksal, 2007b).

6. L'expérience tracée et l'acquisition dynamique de connaissances

L'analyse de traces vise, dans sa globalité, à donner du sens aux modalités utilisées par l'apprenant dans l'environnement d'apprentissage, et si le contexte le nécessite, à son interaction avec les autres apprenants ou le(s) tuteur(s). Ceci dit, certaines approches existantes d'acquisition des connaissances (comme le raisonnement à partir de cas) fixent les éléments à étudier d'une manière parfois statique ; et il pourrait s'avérer que ce genre d'orientation connaît des limites en termes d'évolutivité (par exemple pour certains indicateurs rentrant en jeu). Considérés comme une extension à cela, d'autres approches suggèrent la prise en compte simultanée de plusieurs points de vue conjugués sur une situation d'activité dynamique s'appuyant sur un raisonnement à partir de l'expérience tracée (Mille, 2006).

7. Les traces déclaratives

Les traces ne sont généralement pas directement exploitables, c'est pour cela qu'on parle de trace brute ou première; si bien que, souvent, on adapte et reformule cette trace à l'aide d'un certain nombre d'étapes intermédiaires pour pouvoir l'utiliser. Dès lors, nous aurons une trace modélisée qui définit quels concepts sont représentés selon des indicateurs bien déterminés.

Par ailleurs, au-delà des traces d'interactions informatiques (ou d'activités), qui sont générées, d'une façon volontaire ou involontaire, par l'interaction de l'utilisateur avec le système, nous pouvons introduire les traces déclaratives. Ces types de traces sont engendrés directement par les utilisateurs et sont la conséquence de ce qu'ils laissent volontairement (Ji, Michel, Lavoué, & George, 2013). Un exemple de traces est les réponses à certains questionnaires, entretiens, participations aux forum ou blogs, etc...

8. Liens entre traces d'activité et traces déclaratives

Il est parfois utile d'essayer de comprendre pourquoi telle action a été menée ou choisie. Seulement, si l'on ne dispose que de traces d'interactions, certes ayant l'avantage d'être plus

objectives et plus fiables sur la nature des actions entreprises, il est plus difficile de connaître la raison qui pousse un utilisateur à agir d'une manière spécifique. Si l'on cherche à faire la correspondance entre l'ensemble de traces que peut laisser l'utilisateur y compris les traces volontaires, il est alors peut être judicieux de parler d'identité numérique. Selon (Merzeau, 2009, 2010), le « *digital user* » serait un récepteur de « message », un utilisateur d'outils et un consommateur de services et de produits ainsi qu'un producteur de ressources et donc cette identité serait une collection de traces.

En ce qui concerne notre orientation, nous combinons ces deux types de traces dans l'analyse de certains comportements des utilisateurs.

IV. Conclusion

L'ensemble de ce chapitre nous a permis d'aborder deux points qui nous semblent essentiels pour pouvoir exploiter des ressources ou des parcours d'apprentissage: la visualisation et la navigation ainsi que le traçage d'activités d'apprentissage.

Nous avons pu voir qu'il existait différentes techniques pour visualiser et aider à la navigation, notamment à l'aide de techniques de graphes et d'arbres dynamiques (Hypertree, Treempas, Rgraph, ou Forcirected). Nous avons retenu la bibliothèque JavaScript Infovis Toolkit.

Nous avons également mis en évidence que les traces brutes issues de la collecte de systèmes d'apprentissage ne suffisent pas, compte tenu du fait que ces traces peuvent être de trop bas niveau et qui ne seraient pas suffisamment interprétables. Néanmoins, il est possible de transformer ces traces brutes et d'utiliser des modèles de traces et des opérations de transformations pour synthétiser les indicateurs recherchés.

Les traces d'utilisations sont susceptibles de servir à l'identification des tendances liées à certains profils. Toutefois, nous visons en priorité à:

- exploiter des traces pour modéliser l'utilisateur, dans une première étape ;
- à apporter de l'aide à la navigation ,
- à faire le lien entre la partie édition et la partie utilisation dans une étape ultérieure.

L'objectif étant d'avoir un environnement mis à jour (selon les usages).

Le modèle de navigation que nous proposons doit se baser sur des technologies dynamiques mais qui supporte par extension un mécanisme de traçage spécifique aux apprentissages.

Il ne faut pas omettre la notion de navigabilité ou facilité de navigation.

Nous nous proposons d'exploiter les traces issues des activités de navigation et d'utilisation pour des besoins d'édition afin d'assurer une *meilleure aide à la navigation* dans un ensemble de ressources qui se formalise sous la forme d'un graphe à épistèmes.

Les types de visualisation couplés au mécanisme de traçage que nous tâcherons d'exploiter ont pour intérêt de permettre de bien s'adapter au comportement de l'utilisateur (voir la section IV du chapitre 4 de la partie I), par exemple, en permettant de se focaliser sur certaines zones, de zoomer, de se déplacer, etc...

Quelles autres techniques d'adaptation pouvons-nous réutiliser ? Quel type de navigation permet-il une navigation individualisée ? Ces questions auront des éléments de réponse dans le chapitre suivant.

Chapitre 4 : Personnalisation et hypermédia adaptatifs

I.	INTRODUCTION	54
II.	GENESE : DE LA VERSATILITE, L'ADAPTABILITE A LA PERSONNALISATION	54
III.	PROFIL, MODELE D'UTILISATEUR ET MODELE D'APPRENANT	54
IV.	TECHNIQUES D'ADAPTATION DES HYPERMEDIAS.....	55
1.	<i>Les techniques de présentation adaptative</i>	55
a.	Présentation multimédia adaptative	56
b.	Présentation de texte adaptatif	56
2.	<i>Les techniques de navigation et de support adaptatifs</i>	56
a.	Direct guidance	56
b.	Adaptive link sorting.....	56
c.	Adaptive link hiding	56
d.	Adaptive link annotation.....	57
e.	Map adaptation	57
V.	SYSTEMES HYPERMEDIA ADAPTATIFS.....	57
1.	<i>Tendances de l'adaptation : le web sémantique et la recommandation</i>	58
2.	<i>Suivi et personnalisation des objets d'apprentissage et des parcours</i>	59
VI.	CONCLUSION	59

I. Introduction

Concevoir un hypermédia à finalité d'apprentissage reposant sur un mécanisme de visualisation de navigation dynamique et de traçage d'activités semble nécessaire pour la régulation de l'apprentissage. Il est cependant insuffisant s'il l'on souhaite faire de l'apprentissage différencié ou du moins permettre de répondre à certaines attentes de chaque utilisateur d'une manière plus individualisée. Le mécanisme de traçage, comme nous l'avons vu, peut servir entre autres à faire de l'adaptation ou de la personnalisation. Nous avons également fait l'hypothèse qu'une navigation dynamique et interactive peut aider l'utilisateur à naviguer et à mieux se représenter la structure d'une ressource. Quelles techniques d'adaptation pouvons-nous alors réexploiter pour faire de l'aide à la navigation individualisée ?

Nous discuterons ainsi dans un premier temps de l'apparition des hypermédiats adaptatifs, des différents modèles à prendre en compte notamment de l'apprenant et du domaine et des différentes techniques d'adaptation existantes.

Nous donnerons quelques pistes sur l'évolution des tendances de la personnalisation.

II. Genèse : de la versatilité, l'adaptabilité à la personnalisation

Les débuts de l'hypertexte, souvent attribués au MEMEX⁵² de Vannevar Bush et au projet Xanadu⁵³ de Ted Nelson ont permis à l'utilisateur de naviguer entre des rubriques d'information et de faire de liens entre elles. Plus tard, les technologies hypertextes ont profité de l'émergence des réseaux et du web. Même si les premières formes des hypermédiats destinés au monde de l'éducation étaient relativement figées, le développement de versions prenant en considération les différents types d'apprentissage ou les différents types d'apprenants a laissé apparaître une nouvelle étape dans la différenciation ou l'adaptation. La 1^{ère} période se basait sur la versatilité, c'est-à-dire le changement, mais qui n'était pas toujours le plus adapté pour l'utilisateur. L'adaptation permet de mieux répondre à certaines caractéristiques de l'apprenant. La personnalisation se veut comme une manière unique d'interaction avec la personne utilisatrice. Celle-ci peut définir en plus des traits ou autres caractéristiques qui seront exploités dans la manière de présenter l'apprentissage.

On parle également de différenciation, individualisation voire de personnalisation de parcours d'apprentissage. Mais que peut-on personnaliser ? Il peut essentiellement s'agir :

- des activités proposées,
- des séquences d'activités (nombre et ordre)
- des fonctionnalités,
- des rétroactions,
- ou encore de l'interface.

III. Profil, modèle d'utilisateur et modèle d'apprenant

⁵² Memory Extender ou Memory Index

⁵³ <http://www.xanadu.com/>

Pour prendre en compte l'usage, il est parfois utile d'élaborer un profil regroupant certaines caractéristiques de similarités ou de ressemblances. Ceci peut se décrire sous forme de modèle d'apprenant qui est au départ une structure de données vide. Ce modèle peut être soit statique (si le diagnostic a lieu en une seule fois au début d'une activité) soit dynamique (tout au long des activités) pour ainsi rendre compte d'une façon évolutive des :

- connaissances,
- *misconceptions* (ou erreurs),
- compétences,
- et des comportements de l'apprenant.

Le profil de l'apprenant peut alors être assimilé à un modèle d'apprenant instancié à une période donnée pour rendre compte du diagnostic.

Le diagnostic qui s'appuie sur le recueil des observables peut être soit comportemental (Balacheff, 1994) soit cognitif. La construction du modèle de l'apprenant peut se décliner selon la source d'information, d'une façon :

- explicite (à l'aide de questions posées),
- implicite (en utilisant les interactions avec le système par les traces),
- ou structurelle (par exemple en exploitant le parcours dans un graphe).

Même s'il existe des profils dits « stéréotypés » (ou profils types), le modèle de l'apprenant peut également comporter ses préférences,

- ses stratégies (ou heuristiques),
- ses buts ou ses intentions : essai/erreur, utilisation de l'aide,
- ses comportements (taux d'interactivité),
- et des informations métacognitives.

Qui dit adaptativité dit souvent prise en compte du profil de l'utilisateur. Le modèle de l'utilisateur (ou de l'apprenant) peut se faire par recouvrement (*overlay model*) ou par système à base de *misconceptions* en diagnostiquant les actions erronées ou prématurées, tardives, inutiles ou manquantes, ou prise en compte du contexte dans les interactions (George, 2010) et analyse d'activités.

IV. Techniques d'adaptation des hypermédias

Brusilovsky (1997) classe les techniques d'adaptation en deux principales catégories : celles liées à la présentation et à la navigation (illustrées par la Figure 20)

1. Les techniques de présentation adaptative

L'idée est d'adapter le contenu d'une page accédée par un utilisateur, à ses connaissances, à ses objectifs et à d'autres caractéristiques qui lui sont spécifiques. À titre d'exemple, un utilisateur expérimenté pourra avoir plus de détails et d'informations sur le sujet, dépassant parfois les objectifs initialement visés, alors qu'un utilisateur novice aura des explications additionnelles. On distingue deux types de présentation : la présentation multimédia et la présentation textuelle.

a. Présentation multimédia adaptative

Le contenu d'une page peut être conçu non seulement sous forme de texte mais aussi sous forme d'un ensemble multimédia d'items.

b. Présentation de texte adaptatif

Elle peut inclure l'adaptation de la langue naturelle, par exemple dans un contexte multilingue, ou selon le degré de maîtrise de la langue. Ceci a été largement influencé par la recherche sur les explications adaptatives et la présentation adaptative dans les systèmes tuteurs intelligents.

2. Les techniques de navigation et de support adaptatifs

L'idée principale est d'apporter des mécanismes permettant de mieux orienter les utilisateurs (selon leur profil) dans l'hyperespace, en adaptant la manière de présenter les liens sur les objectifs, les connaissances et les autres caractéristiques d'un utilisateur individuel.

a. Direct guidance

Une des technologies les plus simples est celle qui peut être appliquée dans n'importe quel système qui décide quel est le meilleur nœud par rapport au modèle de l'utilisateur. Pour ce faire, le système peut proposer un plan des liens des meilleurs nœuds à visiter. Ceci peut également servir à recommander ceux qui ne sont pas directement accessibles. Le principal inconvénient est qu'il n'y a pas d'assistance fournie et donc cela constituerait plutôt une forme sommaire d'aide à la navigation. Bien qu'utile, cette technique mérite d'être couplée à une technologie apportant plus de support.

b. Adaptive link sorting

Cette technique repose sur le classement des liens selon un ordre satisfaisant à un ou plusieurs critères issus du modèle de l'apprenant. Même si dans certains cas, cela peut réduire le temps de navigation en n'accédant qu'aux liens triés par ordre de priorité, il peut arriver que cette instabilité de présentation soit épisodiquement perturbante, car le modèle de l'utilisateur doit avoir une certaine stabilité.

c. Adaptive link hiding

Un des points clés de l'Adaptive Link Hiding est de ne montrer que le strict essentiel, en cachant certains liens qui seraient jugés non appropriés par rapport à la situation courante de l'utilisateur (liens qui seraient éloignés de ses objectifs, ou encore sur des notions non encore accessibles). Cette méthode est facile à intégrer, et permet de réduire la charge cognitive de l'apprenant, en évitant qu'il ne se perde dans la complexité de l'hyperespace. Il peut également y avoir des variantes, comme la désactivation de liens (avec des liens qui ne seraient pas directement accessibles) ou encore la suppression de liens qui va au-delà de la simple dissimulation de liens.

d. Adaptive link annotation

Ce procédé adaptatif permet d'ajouter des informations supplémentaires aux liens, par exemple en donnant des commentaires, des indices textuels ou d'autres formes visuelles. Une forme basique est le fait de distinguer les liens visités de ceux non encore visités. Cette forme d'annotation peut être statique ou dynamique dépendant du modèle évolutif de l'apprenant.

e. Map adaptation

Il s'agit d'une adaptation basée sur la disposition, la forme ou le contenu de la carte de navigation d'une façon globale ou locale, telle qu'elle est présentée à l'utilisateur. D'autres techniques que nous avons déjà explicitées peuvent être employées pour adapter la carte de navigation, mais elles restent insuffisantes pour agir sur la structure générale de la carte.

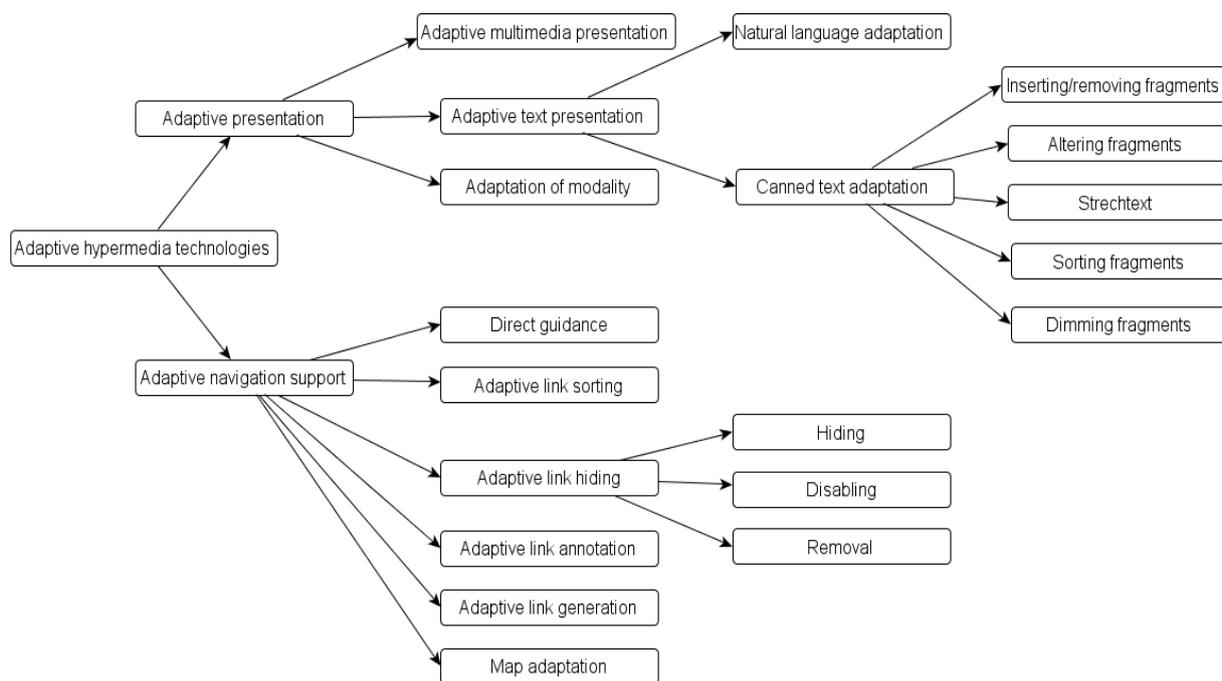


Figure 20 : techniques d'adaptation (d'après Brusilovsky)

V. Systèmes hypermédia adaptatifs

Selon la littérature des hypermédia adaptatifs, différentes méthodes (Brusilovsky, 2001), *frameworks*, techniques et services dans ce domaine ont été déjà étudiés (De Bra et al., 2003; Kobsa, Müller, & Nill, 1998).

La plupart des systèmes utilise la boucle standard de modélisation de l'utilisateur et d'adaptation (cf. Figure 21). En effet le système récupère les données sur l'utilisateur, il les traite et les transforme ensuite pour permettre de générer une présentation différente et d'adapter la manière dont il va naviguer.

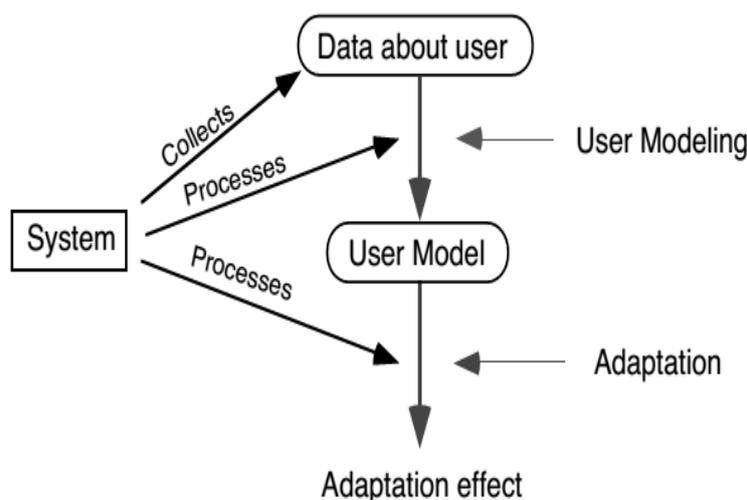


Figure 21 : cycle d'adaptation (Brusilovsky, 1996)

Concernant la programmation, plusieurs idées et techniques d'adaptation ont été implémentées avec succès comme par exemple : QuizPACK (Quizzes for Parametrized Assessment of C Knowledge) (Brusilovsky & Sosnovsky, 2005) et QuizJET (Java Evaluation Tool (Hsiao, Brusilovsky, & Sosnovsky, 2008) ou encore HOP3X⁵⁴ (Després & Jacoboni, 2010; Ouraiba, Choquet, Cottier, Despres, & Jacoboni, 2010; Després, S, & Jacoboni, 2010) . D'autres systèmes comme BlueJ (plutôt de type constructiviste) ou Scratch (Kölling & Rosenberg, 1996) (Maloney et al., 2004) peuvent être considérés comme des IDE préconisés par les enseignants pour les débutants, où il est possible de paramétrer certaines actions.

Nous pouvons également citer les deux environnements suivants qui aident à la modélisation de l'utilisateur :

- Cumulate⁵⁵ se base sur un serveur de modélisation de l'utilisateur et de l'apprenant pour construire des hypermédias adaptatifs. À base d'inférence interne et externe, il capture des données sur l'activité et les événements générés par l'apprenant pour proposer une adaptation adéquate.
- KnowledgeTree⁵⁶ est un portail dans la sphère des systèmes adaptatifs et de la modélisation de l'utilisateur. Il fournit un nombre d'outils pour l'apprentissage des langages C, Java ou SQL.

1. Tendances de l'adaptation : le web sémantique et la recommandation

L'utilisation de réseaux sémantiques ou de cartes conceptuelles peut être appliquée à la définition du modèle de l'apprenant et au modèle du domaine. Certains systèmes s'intéressent à l'adaptation de l'évaluation parfois en adaptant la norme IMS-QTI⁵⁷ (Cheniti-Belcadhi, Braham, Henze, & Nejdil, 2004; Cheniti-Belcadhi, Henze, & Braham, 2008) en utilisant des modèles sémantiques ou des ontologies. Celles-ci pourraient être décrites sommairement comme un ensemble structuré de concepts d'un champ spécifique, où ces derniers seraient

⁵⁴ <http://eiah.univ-lemans.fr/HOP3X/HOP3X.xml>

⁵⁵ <http://adapt2.sis.pitt.edu/wiki/CUMULATE>

⁵⁶ <http://adapt2.sis.pitt.edu/kt/content/Show>

⁵⁷ <http://www.imsglobal.org/question/>

reliés par des relations de type héritage, composition, similarité, etc... Dans le contexte du web sémantique, cela peut être utile de réutiliser ces concepts communs pour assurer une certaine standardisation d'une modélisation d'un domaine donné.

D'autres systèmes adaptatifs construits autour des ontologies méritent d'être mentionnés. Citons par exemple le système OPAL (Cheung, Wan, & Cheng, 2010), ou encore ALS-CPL (El Mezouary 2011) qui est un hypermédia adaptatif sur l'apprentissage du langage C et dont le modèle de domaine est basé sur un concept de granularité fine des objets d'apprentissage. L'intégration de la recommandation est également un élément assez courant dans le processus de l'adaptation. Klačnja-Milićević, Vesin, Ivanović, & Budimac (2011) l'appliquent à un tuteur intelligent sur le langage Java, en proposant des tâches plus au moins élaborées selon le profil de l'apprenant.

2. Suivi et personnalisation des objets d'apprentissage et des parcours

Il peut y avoir trois types de suivi qui peuvent être mis en œuvre pour coupler la personnalisation ou l'individualisation des objets d'apprentissage, voire des parcours (Soller, Martínez, Jermann, & Muehlenbrock, 2005) :

- le « mirroring » : il s'agit d'une simple réflexivité d'un certain nombre d'actions de l'apprenant généralement sur des éléments capturés sans traitement ou analyse poussée ;
- le « monitoring » : par rapport au « mirroring », le monitoring assure un suivi sur plus d'actions de l'utilisateur et transmet des indicateurs (calculés selon des degrés variables) à l'apprenant ou à d'autres acteurs (tuteur, enseignant). Ces données « utiles » ne sont générées généralement qu'après des traitements spécifiques (étape liée aux traces transformées ou modélisées) ;
- et le « guiding » : la finalité du « guiding » ou guidage est de proposer à l'apprenant une progression pas à pas sous forme d'assistance personnalisée ou de rétroactions adaptatives. L'utilisateur ayant un degré de liberté réduit, se voit ainsi contraint de suivre les prescriptions du système s'il veut atteindre les objectifs d'apprentissage. Ceci pourrait développer, dans certaines situations, un caractère parfois intrusif chez l'utilisateur.

VI. Conclusion

Le système que nous proposerons est considéré comme un hypermédia adaptatif basé sur le web et orienté vers la pratique d'exercices où l'adaptation se fait principalement sur une carte de navigation comme le système RATH (Hockemeyer, Held, & Albert, 1998). Ce système hypermédia adaptatif se base sur une modélisation de connaissances liées par des dépendances de type prérequis. Les connaissances de l'apprenant est un sous-ensemble des connaissances expertes. L'intérêt de cette orientation réside dans la multitude des parcours possibles de navigation, et par conséquent chaque trajectoire sera individuelle puisque le

choix est laissé à l'apprenant sans nécessairement le contraindre. Ceci constitue également une forme de personnalisation ou d'individualisation.

Par rapport aux techniques citées ci-dessus, l'hypermédia épistémique adaptatif (dont le prototype HiPPY est présenté dans le chapitre 4 de la partie 2) que nous construisons, pourra s'appuyer sur :

- une navigation adaptative basée sur la présentation d'un graphe plutôt qu'une présentation adaptative de texte, il permettra plus exactement :
 - de disposer d'un suivi (ou « *monitoring* ») sommaire, puisqu'il proposera des parcours à choisir selon les notions maîtrisées concernant le type de suivi (cf. section V.2);
 - d'assurer un aspect de dissimulation de liens (ou « *link hiding* » ou « *disabling* ») et de désactivation de liens qui ne seraient pas conseillés directement à l'apprenant (technique de présentation adaptative décrite en section IV);
 - de rendre compte de l'évaluation épistémique de l'utilisateur en exploitant les aspects annotatifs (techniques de navigation et de supports adaptatifs, section IV.2);
 - et d'avoir une adaptation de la carte navigationnelle qui permet de se focaliser sur certaines régions pour avoir à la fois cette idée de vue localisée ou de vue globale.

Une fois que le système à caractère adaptatif sera mis en place, il reste toutefois des questions sur l'exploitation d'une quantité importante de traces, surtout s'il ya énormément de données à collecter. Y a-t-il des techniques ou des heuristiques qui permettent de classifier les traces ou de repérer des informations qui ne sont pas directement présentées ? Au-delà des techniques statistiques, y-a-t-il des procédés ou des processus d'analyse orientée en apprentissage artificiel ? Existe-il des techniques et des algorithmes de fouille qui fonctionnent mieux dans le contexte de l'éducation et d'autres qui seraient à rejeter ? C'est ce que nous allons tenter de cerner dans le chapitre 5.

Chapitre 5 : La fouille de données en éducation

I.	INTRODUCTION	62
II.	TECHNIQUES ET ALGORITHMES COMMUNS EN FOUILLE DE DONNEES EN EDUCATION	62
1.	<i>Apprentissage supervisé</i>	62
a.	Les arbres de décision	63
b.	Les réseaux de neurones	64
2.	<i>Apprentissage non supervisé</i>	64
a.	Clustering.....	64
b.	Les règles associatives	65
c.	Sequence mining.....	65
III.	QUELLES ANALYSES SONT NECESSAIRES ET QUE RECHERCHER ?.....	66
IV.	OUTILS ET PLATEFORMES SPECIFIQUES POUR LA FOUILLE DE DONNEES EN EDUCATION.....	68
V.	TENDANCES ET PERSPECTIVES SUR LA FOUILLE DE DONNEES EDUCATIVES.....	69
VI.	CONCLUSION	70

I. Introduction

Pouvoir collecter les traces d'interactions liées à l'utilisation de systèmes d'apprentissage est une étape cruciale. Cependant, que faut-il faire pour exploiter ces traces ? Ces traces, pour qu'elles soient formatées selon des modèles précis, nécessitent une interprétation plus poussée pour identifier des informations qui peuvent être cachées, et afin de faire croiser des informations utiles.

Ce chapitre a pour objet d'étudier un certain nombre de techniques de fouille de données qui sont les plus utiles à l'exploitation de traces d'interaction dans les EIAH. Nous rappelons dans un premier temps les techniques et algorithmes en fouilles de données les plus connues dans un contexte d'éducation. Dans un second temps, nous présentons les données utiles à regarder en faisant le lien avec les techniques citées. Nous illustrons cela par des outils fréquemment utilisés en « Data-Mining » et plus spécifiquement même en « Educational Data Mining ». Nous ouvrirons des perspectives d'évolutions sur les tendances à venir avec l'émergence de nouvelles modalités d'apprentissage ou encore avec le développement de techniques plus avancées.

II. Techniques et algorithmes communs en fouille de données en éducation

Selon (Baker & Yacef, 2009) (Romero & Ventura, 2010a), il existe différents types d'algorithmes pour la fouille de données dans un contexte d'éducation. Ces types varient selon les objectifs de :

- prédiction :
 - o cela concerne par exemple les techniques de classification, régression, ou l'estimation latente de connaissances « Latent Knowledge Estimation ».
- description par la découverte de structures :
 - o avec le clustering, l'analyse en facteurs, la découverte de structure de domaine ou de modèles, ou encore avec l'analyse de réseaux.
- fouille de relations ou de dépendances (qui peut être exploitée dans la description ou dans la prédiction) :
 - o en utilisant entre autres la fouille de règles d'association, la fouille de corrélations, la fouille de motifs dans les séquences « Sequential Pattern Mining et la fouille de données causales (causal data mining).

Ces différentes techniques ont pour point commun la distillation de données utiles ou interprétables (selon des degrés variables) pour un jugement humain.

1. Apprentissage supervisé

L'apprentissage supervisé se réfère à la capacité d'apprendre à réaliser des tâches à partir de données déjà étiquetées. D'une certaine manière, le but est de refaire la même tâche à partir

de données déjà existantes. Cet apprentissage peut concerner, selon ce qu'on cherche à déterminer :

- soit les méthodes de classification (partant de l'entrée on souhaite déterminer la sortie)
- soit les régressions (partant de la sortie, l'entrée est à déterminer).

Nous décrivons comme exemples les arbres de décision et les réseaux de neurones.

a. Les arbres de décision

Les arbres de décision sont des outils de support matérialisés souvent sous la forme d'une structure arborescente (constituée de nœuds et de branches) pour modéliser des décisions et des conséquences, selon une répartition d'une certaine population avec des attributs fixés et connus. Les nœuds feuilles donnent souvent une prédiction sur les données à classifier.

Le principal intérêt des arbres de décision réside dans le fait que cela est facilement interprétable par les utilisateurs.

Parmi les applications des arbres de décision en éducation (Romero & Ventura, 2010b) nous pouvons citer celle où l'analyse permet de :

- recommander une séquence optimale d'apprentissage pour les apprenants (et donc de faire de l'adaptation),
- prédire les performances ou la satisfaction d'un apprenant,
- ou encore détecter le modèle mental d'un apprenant dans un ITS⁵⁸.

Un des algorithmes utilisés pour obtenir des arbres de décision est le C4.5, illustré par un exemple en Figure 22. Sur cette figure, nous prenons l'exemple de joueurs de tennis qui décideraient de jouer selon l'aspect du ciel. Si le ciel est couvert (overcast), ce sera positif (yes). Si le ciel est ensoleillé (sunny) avec une humidité (humidity) inférieure ou égale à 75% ou bien s'il pleut sans qu'il y ait de vent alors la décision sera également positive. En revanche, la réponse sera négative dans les autres cas : ensoleillé avec une humidité supérieure à 75% et pluvieux avec vent. D'autres algorithmes existent comme J48, ID3 ou encore CART qui est une forme évoluée de ID3. Cette technique nous sera utile dans le cadre de nos recherches notamment dans une optique d'analyse et de modélisation de résultats.

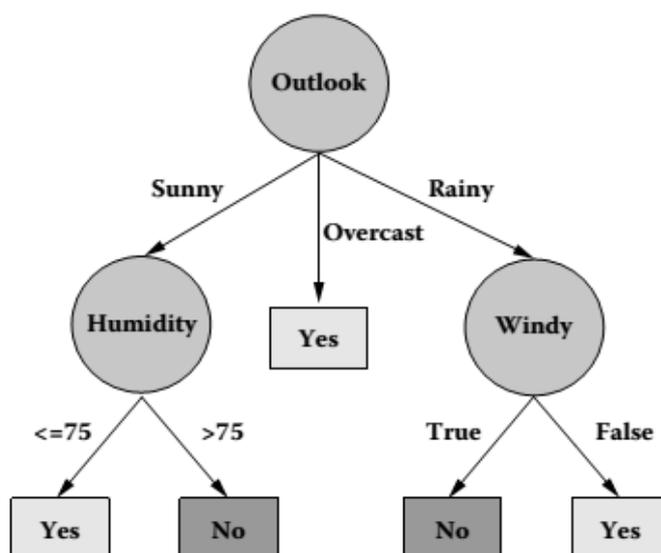


Figure 22 : arbre de décision induit par C4.5 (d'après Wu & Kumar, 2010)

⁵⁸ Intelligent Tutoring System.

b. Les réseaux de neurones

Selon (Livieris, Drakopoulou, & Pintelas, 2012), pour prédire les performances des apprenants, les réseaux de neurones artificiels sont des modèles de calculs parallèles comprenant des processus interconnectés et caractérisés par un apprentissage à partir de l'expérience et de la découverte de nouvelles connaissances. Par analogie au fonctionnement des neurones biologiques, chaque neurone contient un certain nombre d'entrées (synapses). Le réseau est entraîné sur un jeu de données pour déterminer la correspondance entrée-sortie et attribuer des pondérations (coefficients synaptiques). Cette technique, même si nous avons estimé judicieux de la présenter, nous ne la retiendrons pas dans nos recherches. Elle pourrait engendrer rapidement des modèles complexes et moins facilement interprétables par l'humain. De plus, nous nous inscrivons plutôt dans une démarche descriptive quant à l'exploitation des résultats alors que ce type de technique serait plus orienté sur la prédiction.

2. Apprentissage non supervisé

À l'inverse de l'apprentissage supervisé, dans l'apprentissage non-supervisé, il n'existe pas d'attribut particulier cible, on dit alors que les données ne sont pas étiquetées. Nous illustrons cette approche d'apprentissage machine par le clustering, les règles associatives et le « sequence mining ».

a. Clustering

Le clustering, à ne pas confondre avec la classification en mode supervisé, a pour objectif de segmenter, en l'occurrence de diviser un certain nombre d'objets donnés en entrée en groupes ou « *clusters* ». Le fait de décider si un objet appartient à tel groupe est régi par la notion de distance (qui peut être euclidienne par exemple). Par conséquent, la finalité est de minimiser la distance intra-groupe pour avoir des éléments homogènes, mais en outre de maximiser cette distance pour avoir des sous-ensembles distincts. K-means est un exemple d'algorithme fréquemment utilisé dans ce but. Ici, la Figure 23 illustre le résultat de l'exécution de K-means qui donne en sortie 3 clusters.

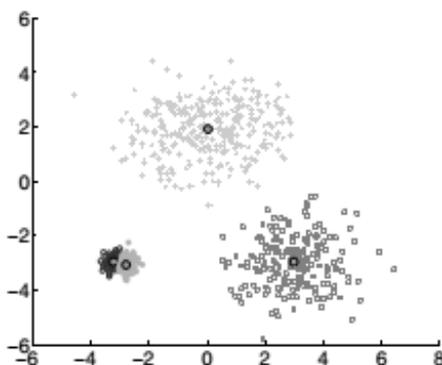


Figure 23 : résultat de la convergence du clustering (Wu & Kumar, 2010)

b. Les règles associatives

Les règles d'associations sont, à partir d'une base de transactions (ou groupement d'événements), des règles de type « Si → Alors », qui tendent à établir des corrélations causales entre des éléments spécifiques. Chaque transaction peut être décrite par un identifiant et une liste d'items. Parmi les algorithmes utilisés, nous pouvons citer FSC « Frequent Set Counting »

Trois métriques sont importantes pour calculer les fréquences d'apparition : le **support** (1), la **confiance** (2), l'intérêt de la corrélation appelé « *lift* » (3). Le cosinus, quant à lui se déduit du « *lift* ». Ces métriques sont également utilisées dans des algorithmes de « *sequence mining* » tels que GSP (*Generalized Sequence Patterns*).

Pour une règle $A \rightarrow B$, c'est-à-dire « si A alors B », ces métriques se calculent de la façon suivante :

- le support (ou la couverture) =
$$\frac{\text{nombre d'items satisfaisant la règle}}{\text{nombre d'items total}}$$
- la confiance =
$$\frac{\text{nombre d'items satisfaisant la règle}}{\text{nombre d'items liés à la condition Si}}$$
- le lift (ou encore l'intérêt de la corrélation) =
$$\frac{P(A \wedge B)}{P(A)P(B)}$$
- et cosinus (proportionnel au lift) =
$$\frac{P(A \wedge B)}{\sqrt{P(A)P(B)}}$$

Selon (Merceron & Yacef, 2008) la métrique du cosinus (un indicateur proportionnel au lift) permet d'interpréter facilement les résultats de corrélation en éducation.

De plus, un cosinus > un seuil de 0.65 peut signifier qu'il s'agit d'une règle d'association qui mérite d'être retenue.

c. Sequence mining

Le *sequence mining* ou fouille de séquences examine la récurrence de certaines occurrences ou motifs, selon un ordre lié à la séquence. Des techniques de fouille de texte (*Text-mining*) peuvent également être intégrées pour attribuer une syntaxe particulière.

L'intérêt en e-learning réside dans le fait de pouvoir recommander certaines ressources ou parcours en ne connaissant qu'une partie de la séquence. Cela sous-entend bien évidemment le fait d'avoir identifié les séquences les plus fréquentes.

Un des algorithmes les plus utilisés est le GSP, la variable de temps peut optionnellement être incluse. Free-span, Prefix-Span sont d'autres algorithmes plus rapides mais se basant sur la même idée que GSP.

Nous pouvons, à présent, noter quelques remarques sur l'utilisation des différents algorithmes dans l'exploitation de données issues des systèmes d'enseignement en ligne :

- K-means est couramment utilisé et permet par exemple la détection des émotions à partir de fichiers logs uniquement (Baker et al., 2012);
- les règles de décision ont l'avantage d'être facilement interprétables, par exemple JRip et PART se basent sur la création d'arbres de décision et donnent généralement de meilleures règles.
- SVM⁵⁹, une autre panoplie de techniques en apprentissage supervisé, créerait selon (Baker & Yacef, 2009b) des modèles plutôt sophistiqués ; il peut être employé pour le text-mining mais, il n'est pas adapté aux données issues de situations éducatives (logs, notes, interactions avec les systèmes). De même, les algorithmes génétiques et les réseaux de neurones sont bien adaptés à certains problèmes mais pas à un contexte éducatif, car historiquement, ils n'ont pas produit les meilleurs résultats.

III. Quelles analyses sont nécessaires et que rechercher ?

Une des finalités de l'analyse des données issues des systèmes d'apprentissage est d'améliorer les dispositifs d'enseignement en ligne et de pouvoir apporter une meilleure adaptation des contenus (voir le chapitre sur l'analyse des traces). Comment alors exploiter ces techniques de fouilles de données dans un contexte d'éducation ?

Divers environnements permettent d'évaluer les apprenants et de faire la corrélation entre usages et performances en suivant parfois les démarches utilisées (Delozanne, Le Calvez, Merceron, & Labat, 2007). Le système de Kumar en C++ (Kumar, 2005) et le système Logic-ITA (Merceron & Yacef, 2004) sont des cas d'illustration. D'autres situations d'études (Labat et al., 2007) cherchent à déterminer quels types de connaissances sont mises en œuvre pour la résolution dans un problème donné.

L'analyse des usages et des activités est donc capitale si l'on souhaite pouvoir évaluer l'impact de l'utilisation d'un système sur la manière de procéder pour résoudre une tâche donnée, ou même plus globalement sur l'apprentissage. C'est le cas par exemple avec l'analyse de l'activité de recherche d'apprenants de langue à l'aide d'un dictionnaire en ligne (Hamel, 2013). Dans ce type d'activité, les questions de recherche s'intéressent à la mesure de l'efficacité et de l'efficacité⁶⁰ d'une certaine trajectoire. Peuvent intervenir alors les paramètres de temps, d'utilisation et d'accès, et la relation entre efforts et score obtenu. Le résultat de cette recherche a permis d'identifier des patrons de parcours « recherche » en fonction du profil de départ des utilisateurs.

La détection du comportement fait encore l'objet de recherches assez poussées, l'idée est qu'à partir de traces, on puisse déterminer le comportement d'un étudiant. Nous pouvons énumérer :

- les comportements de désengagement ou d'engagement :

⁵⁹ Support Vector Machines

⁶⁰ L'efficacité, au-delà de l'efficacité, est la manière d'optimiser un certain nombre de paramètres pour arriver au résultat attendu.

Gaming the System (Baker, Corbett, Koedinger, & Wagner, 2004; Baker et al., 2008; Cheng & Vassileva, 2005), souvent avec des comportements dits hors-tâche "Off-Task Behavior". Le concept de « gaming the system » c'est-à-dire, détourner l'usage du système d'apprentissage en utilisant ses propriétés intrinsèques pour faire croire un pseudo-apprentissage. On peut citer comme exemple, le fait de deviner d'une façon systématique, de répondre au hasard, etc...

- les comportements métacognitifs : comme l'évitement de l'assistance « Help Avoidance » (Alevan, McLaren, Roll, & Koedinger, 2004)
- l'exploration des comportements pour modéliser l'apprenant à l'aide de techniques d'apprentissage machine supervisé et non- supervisé (Amershi & Conati, 2009)

Quels niveaux de grain d'analyse choisir alors ? Ce qui paraît important à prendre en compte c'est la fenêtre d'observation, est-ce une session complète, un jour, par observation, par chapitre, par exercice ? Cela permet d'assurer un équilibre entre un nombre élevé d'observables qui seraient parfois inutiles ou sources de bruits et un nombre très réduit qui nous empêcheraient de construire une quelconque conclusion. Il est possible de définir un modèle lié au parcours d'apprentissage effectif de l'apprenant selon des niveaux de parcours. C'est ce que nous allons modéliser dans la partie 2 (chapitre 2).

D'autres questions plus générales peuvent notamment concerner les constats suivants:

- les raisons d'abandon des étudiants,
- la prédiction des étudiants qui vont abandonner,
- les préindicateurs de réussite des étudiants,
- le fait de savoir si les compétences sont correctement développées chez les étudiants (généralement en utilisant le modèle prédictif Bayes Knowledge Tracing),
- le fait de savoir si les ressources (vidéos, forum, wiki, rappels de cours...) sont utiles à la résolution d'une question ou d'un problème donné,
- quelles catégories peuvent intervenir pour classer les apprenants (experts, butineurs ou "lurkers", néophytes) ?
- comment les enseignants gardent-ils les étudiants motivés ?
- y a-t-il des sous-communautés à travers les interactions dans les forums ?
- quelles corrélations entre données objectives démographiques (âge, genre, langue, pays) et résolutions ?
- les ressources à améliorer ?
- quelles compétences sont nécessaires pour résoudre un problème ?

Pour notre part, il s'agira d'analyser les tendances dégagées en termes de navigation dans les épistèmes et leurs liens avec la validation des exercices, prioritairement dans une perspective descriptive.

Ceci s'appuie sur une aide dans la phase des premiers pas (correction de code) incluant les questions de syntaxe et de constructions typiques. L'hypermédia que nous proposerons est orienté épistèmes dans sa définition conceptuelle et dans sa construction. En d'autres termes, il s'appuiera sur un domaine de connaissances matérialisé par le graphe des épistèmes. Il offrira également certaines propriétés métacognitives (par exemple en rendant compte de la manière dont l'apprenant fit appel aux épistèmes : avant la résolution de l'exercice, comme support d'aide à l'apprentissage, ou après la résolution, comme support à la validation) mais d'une façon moins élaborée que celle utilisée dans les rétroactions épistémiques (Luengo, Vadcard, Tonetti, & Dubois, 2011).

IV. Outils et plateformes spécifiques pour la fouille de données en éducation

À notre connaissance, il n'existe pas un outil (ou une plateforme) de référence pour traiter les données issues d'environnements d'enseignement en ligne aussi variés soient-ils (ITS, LMS, simulateurs, hypermédia adaptatifs sur l'apprentissage, MOOCs..). Une des raisons derrière cela, est la variété des systèmes et surtout la variabilité des données et de leur modèle de collecte. Nous proposons de présenter brièvement les outils Tada-ed, EDM workbench, et les plateformes PSLC DataShop et UnderTracks.

- Tada-ed⁶¹, est un outil de fouille, de visualisation et de transformation de données issues d'exercices en ligne (Merceron & Yacef, 2005) . Destiné aux enseignants, il permet d'identifier les motifs pertinents (par exemple les erreurs commises par les apprenants) selon des indicateurs pédagogiques distinctifs et en appliquant des algorithmes précis de data-mining. Citons par exemple des algorithmes sur les règles associatives pour permettre de relever des cas du genre : « Si un apprenant a commis l'erreur A et l'erreur B, il aurait des chances de commettre l'erreur C ».
- EDM workbench ⁶²: selon ses auteurs (Rodrigo, Baker, McLaren, Jayme, & Dy, 2012) il permet aux chercheurs de définir et de modifier des catégories de comportements d'intérêt (hors-tâche, non réactivité, bachotage..), d'étiqueter les données et de collaborer avec d'autres utilisateurs. Il fournit également un support de compatibilité avec les outils Weka et Rapidminer.
- PSLC⁶³ DataShop désignant Pittsburgh Science of Learning Center est un entrepôt ouvert de données (Koedinger et al., 2010) principalement issues de ITS⁶⁴ mais aussi de laboratoires virtuels, de simulations et d'autres environnements d'apprentissage collaboratifs. Destiné en priorité aux analystes de données issues d'environnements d'enseignement en ligne, cette plateforme se veut comme un espace de partage d'expériences à partir de données capitalisées.
- UnderTracks⁶⁵ est préconisé par (Bouhineau, Luengo, Mandran, Ortega, & Wajeman, 2013) : cet entrepôt de traces se veut une plateforme intégrée de collecte, de conservation, d'exécution et de partage de traces d'interactions et de processus d'analyse ou de traitement de données. Un des objectifs cités par ses auteurs est de pouvoir s'ouvrir sur un grand nombre de types d'EIAH.

Même s'il existe des outils et des plateformes spécifiques pour la fouille de données en éducation (Educational Data Mining), les outils de data Mining plus généraux, vu leur communauté et leur caractère libre (dans la majorité des cas), restent néanmoins fréquemment utilisés en contexte d'éducation. Voici quelques uns : Weka, RapidMiner (Rapid-I), Knime. Ces outils s'appuient également sur des processus d'analyse et de traitement. Dans une

⁶¹ <http://chai.it.usyd.edu.au/Projects/TadaEd>

⁶² <http://penoy.admu.edu.ph/~alls/archives/593>

⁶³ Pittsburgh Science of Learning Center, voir <http://pslcdatashop.org>

⁶⁴ Intelligent Tutoring Systems

⁶⁵ <https://undertracks.imag.fr/>

moindre mesure, nous pouvons noter Google Refine et SPMF⁶⁶ qui est une bibliothèque spécifique pour l'analyse de motifs séquentiels.

V. Tendances et perspectives sur la fouille de données éducatives

Avec la massification et l'émergence des environnements ciblant un nombre d'apprenants plus important, il existe des moyens de traitements distribués et parallélisables avec des algorithmes plus performants, par exemple avec le framework Hadoop⁶⁷ qui implémente MapReduce pour gérer d'une façon échelonnée des données massives avec les opérations de base Map() et Reduce().

Néanmoins, les données collectées permettent toujours de s'intéresser à des points cruciaux comme :

- les données d'interactions avec le système : il s'agit le plus souvent de traces de bas-niveau (clic, ouverture, navigation) ;
- les données d'interaction avec les pairs (forum, réseaux sociaux, messages, chat...) ;
- et les données sur les profils (indices démographiques).

Avec la rapide multiplication des MOOC à orientation certificative (e.g. Coursera⁶⁸, Udacity⁶⁹) et « l'explosion des données » issues d'environnements d'apprentissage comme le souligne (Siemens & Long, 2011), a pour effet de transformer certains LMS en MOOC comme Canvas Network⁷⁰ ou CoursesSites⁷¹ de Blackboard., une des tendances qui pourraient voir le jour, c'est l'apparition des MOOC sur le Cloud Kop & Carroll (2011) proposent un MOOC basé sur le Cloud et évaluent certaines traces des apprenants. Ces deux aspects ont l'air conciliables même si cela pose épisodiquement de nouvelles difficultés en termes de collecte de traces.

Quelques travaux se sont penchés sur l'analyse de données provenant de MOOC comme (Veeramachaneni, Dernoncourt, Taylor, Pardos, & O'Reilly, 2013) en proposant des modèles standardisés et des modalités de visualisation appropriées (Dernoncourt et al., 2013.) ou encore des frameworks d'analyse (Han, Veeramachaneni, & O'Reilly, 2013.).

Nous pensons que les méthodes et processus de fouille et d'analyse de données que nous introduirons par rapport au contexte France-IOI (nombre d'utilisateurs approchant les 865 en Python, peut-être un millier s'il l'on compte les autres langages) sont transposables à de telles modalités d'apprentissage (MOOC) ou d'évolutions technologiques telles que le *Cloud computing*. Bien que la plupart des MOOC actuels proposent des séquences parfaitement linéaires et identiques pour chaque apprenant, nous pensons que les techniques de

⁶⁶ a sequential pattern mining framework, <http://www.philippe-fournier-viger.com/spmf/index.php?link=algorithms.php>

⁶⁷ <http://hadoop.apache.org/>

⁶⁸ www.coursera.org

⁶⁹ www.udacity.com

⁷⁰ www.canvas.net

⁷¹ www.coursesites.com

personnalisation aidant, ils pourront évoluer vers des systèmes proches de HiPPY où le nombre de ressources accessibles dans le MOOC est bien plus important que celles qui sont utiles à un seul apprenant, mais constituent plutôt une couverture complète des notions à enseigner pour l'ensemble des apprenants potentiellement concernés. Dès lors que le contrôle (dans la navigation à travers ces ressources), sera laissé à l'apprenant, l'analyse des parcours d'apprentissage pourra être utilisée à des fins de support à l'apprentissage, au tutorat ou encore à la réingénierie du MOOC.

VI. Conclusion

Nous avons explicité un certain nombre de techniques et d'algorithmes de fouille de données selon des approches d'apprentissage non-supervisé (clustering, règles associatives ou sequence pattern mining) et selon des approches d'apprentissage supervisé (réseaux de neurones, arbres de décision). Nous avons pu remarquer que certains algorithmes n'étaient pas assez adaptés pour un contexte d'enseignement en ligne (e.g. SVM, les réseaux de neurones, algorithmes génétiques). Même s'il existe peu d'outils spécifiques pour la fouille de données en éducation, les autres outils que nous avons relevés comme Weka, RapidMiner, Knime peuvent être exploités d'une façon complémentaire. En ce qui concerne les données utiles à identifier à partir des résultats d'exécutions des différents algorithmes impliqués, un certain nombre d'indicateurs peut intervenir :

- mesurer l'engagement d'un apprenant pour savoir s'il ne répond pas en bachotant ou par hasard (*Gaming the system*) ou mesurer des motifs récurrents qui permettraient de dire s'il s'agit d'un succès ou d'un échec avec Pattern mining ou des règles d'association, et ainsi pouvoir classifier les apprenants ;
- cerner le niveau de la fenêtre d'analyse (sur une journée, une session, par problème) ;
- constater, comme nous le verrons, qu'il est possible d'introduire la notion de trajectoire multi-niveaux qui est liée à la bonne résolution d'un exercice (voir en partie 2 la section III du chapitre 3).

Au-delà de ces aspects, il ne faut pas sous-estimer certaines évolutions de modalités comme les MOOCs (Cours Massifs en Ligne) qui apportent une massification des données mieux connu sous le nom Big Data, ou autres émergences technologiques comme le cloud computing. Ces deux éléments auront vraisemblablement un effet sur les développements des processus communs aux domaines de *l'Education Data mining* ou du *Learning Analytics*.

Conclusion de la partie I

Cette première partie nous a permis de voir, à travers ses cinq chapitres, quels étaient les éléments nécessaires et les prérequis pour construire un hypermédia épistémique adaptatif et traçable basé sur une navigation dynamique. En guise de synthèse de cette première partie, nous pouvons annoncer que l'hypermédia épistémique que nous présenterons en partie 2 (HiPPY) se composera principalement de cinq couches (et d'une sixième couche optionnelle), où chaque couche transmet ses données à la couche inférieure (cf. Figure 24) :

- La couche 1, couche ressources (*resources layer*) est celle qui contient d'une façon brute les objets d'apprentissage indépendamment de la granularité, pour nous, il s'agit principalement des unités épistémiques comme entités.
- La couche 2, couche parcours (*pathway layer*) utilise une agrégation de ressources pour produire des parcours d'apprentissage. Cette couche désigne par extension le parcours prescrit et le parcours effectif de l'apprenant.
- La couche 3, la couche navigation (*navigation layer*) exploite les données de parcours et sert à représenter le parcours sous forme de navigation statique ou dynamique, elle est importante tant sur la présentation que sur le contenu.
- La couche 4, la couche de traçage (*tracing layer*) utilise les données renvoyées par la couche d'en dessous, elle permet de collecter tous les éléments issues de la navigation en termes de contenus, d'événements enregistrés, il s'agira bien entendu d'étudier les indicateurs de plus haut niveau pour pouvoir apporter des éléments d'individualisation.
- La couche 5, couche de fouille et d'individualisation (*mining and individualization layer*), filtre et différencie, selon les données du profil de l'utilisateur, certains éléments comme le diagnostic, la progression et la manière de naviguer. Elle peut s'appuyer sur des traitements d'analyse simples et sur des algorithmes de fouilles de données plus avancés pour générer des statistiques, établir des corrélations.
- La couche 6 (optionnelle) d'édition ou de réingénierie (*reengineering layer*) va exploiter les données de la couche fouille pour proposer des idées de changements ou d'édition portant sur l'interface générale ou sur le contenu structurel. Cette couche est liée à la couche ressources.

Le schéma suivant illustre les liens entre ces couches :

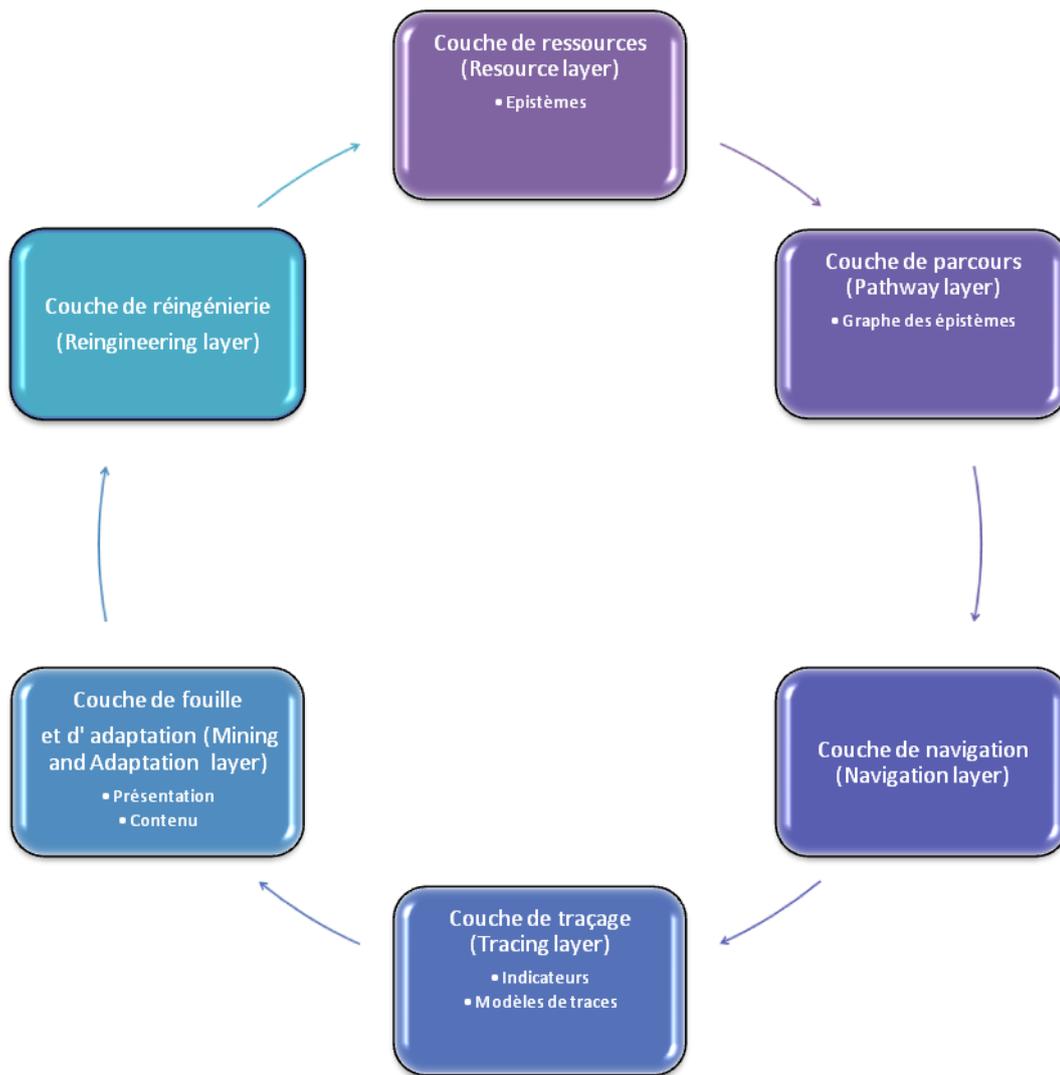


Figure 24 : architecture en couches de l'hypermédia épistémique

PARTIE II : D'UN GRAPHE STATIQUE DES ÉPISTÈMES VERS UN HYPERMÉDIA ÉPISTÉMIQUE ADAPTATIF

Introduction de la partie II :

Suite à la partie I, où nous avons traité d'un certain nombre de concepts et de modèles et positionné nos ancrages théoriques par rapport aux travaux de la littérature, cette partie II a pour objectif de montrer nos principales contributions en termes de choix de modélisations, de conceptions, d'implémentation et d'application pour retenir différents outils les mettant en œuvre. Cette partie se veut dans la continuité de la partie I, en ayant la plupart du temps le même cheminement exposé. C'est pourquoi elle aura pour objet de décrire nos contributions sur les concepts déjà traités de ressources et de parcours d'apprentissage, de visualisation et de navigation dynamique, sur les dispositifs de traçage et sur les processus d'analyse et de fouille de données. Bien évidemment nous tenterons de susciter le questionnement et de proposer des mécanismes d'aide à la navigation individualisée et des principes d'aide à l'édition. Le 1^{er} chapitre expose l'approche globale introduite qui corrèle les liens entre édition et navigation. Elle fait suite à l'architecture en couches introduite en fin de partie I. Le second chapitre s'intéresse à la modélisation et à la conceptualisation en ressources et parcours à partir du graphe des épistèmes. Le troisième chapitre énonce l'approche envisagée en termes de visualisation et de navigation dynamique individualisée. Il propose des choix techniques à prendre en compte pour l'implémentation des prototypes décrits par la suite. D'autres éléments de contribution notamment le mécanisme de modélisation des traces et leur exploitation sont donnés dans le chapitre 4 et illustrés par la réalisation de quelques prototypes. Ces prototypes seront une occasion de mettre en œuvre les concepts traités. Le chapitre 5 décrit les outils et techniques algorithmiques envisagés et utilisés pour faire ressortir les données utiles. Les différents liens entre les chapitres de cette partie sont illustrés dans la Figure 25.

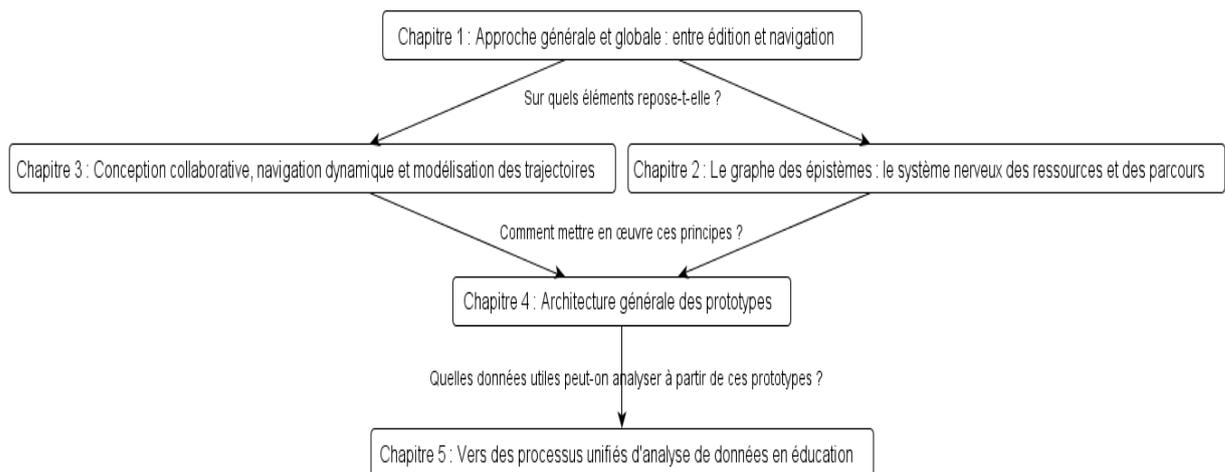


Figure 25 : liens entre les chapitres de la partie II

Chapitre 1 : Approche générale et globale : entre édition et navigation

I.	INTRODUCTION	75
II.	UN CYCLE DE VIE DES RESSOURCES BASE SUR L'ARTICULATION ENTRE LES ACTIVITES D'EDITION ET DE NAVIGATION : UNE RELATION NON TRIVIALE	75
	1. <i>Points de cadrage et contraintes</i>	75
	2. <i>Pourquoi lier édition et navigation ?</i>	76
	3. <i>Comment lier ces activités ?</i>	76
	4. <i>Modélisation générale</i>	77
III.	LES APPORTS VISEES DE CETTE APPROCHE.....	78
IV.	CONCLUSION	80

I. Introduction

La conception et l'édition de ressources d'apprentissage et l'utilisation et/ou la navigation dans ces ressources sont parfois assimilées à des activités antagonistes. En effet, elles font intervenir des acteurs différents et des finalités divergentes. Indépendamment de son utilité et son utilisation, si l'on considère qu'une ressource doit être vivante, il faut alors prévoir des mécanismes d'évolution pour répondre aux attentes de différents utilisateurs : apprenants, concepteurs, tuteurs, mainteneurs, etc.

Sur le modèle en couches que nous avons proposé (sur la Figure 25), le lien entre édition et navigation peut se traduire par la dépendance entre la « couche de réingénierie » et la « couche de ressources ». Même si nous n'avons pu totalement expérimenter cette approche, l'une des contributions de cette thèse est l'introduction de ce type de dépendance que nous illustrons par quelques possibilités de corrélation entre navigation et révision au profit de la présentation et du contenu.

Dans la première section de ce chapitre, nous décrivons le cycle de vie de la ressource, basé sur la dépendance entre les activités de conception/d'édition avec les activités d'utilisation et de navigation. Nous donnerons quelques schémas synthétiques illustrant cette approche, centrée sur les activités et sur la ressource (ou macro-ressource). Nous donnons en deuxième section les principaux apports de cette approche tout en montrant l'évolution des différents niveaux de la ressource et les acteurs impliqués.

II. Un cycle de vie des ressources basé sur l'articulation entre les activités d'édition et de navigation : une relation non triviale

De nombreux travaux ont traité des objets d'apprentissage (Pernin 2003) et de leur conception et leur édition plus spécifiquement, dans une optique d'ingénierie pédagogique (Paquette 2002) (Burgos et al. 2005) (Burgos, Hummel, Tattersall, Brouns, & Koper, 2007) (Wiley & Gurrell, 2009). D'autres travaux se sont penchés sur l'accès et la navigation dans les ressources (Nash, 2005) (Wolpers, Memmel, & Giretti, 2009) (Wolpers et al., 2008). Cependant, peu d'entre eux s'intéressent à la relation entre ces activités d'édition et de navigation qui ont des objectifs différents, faisant intervenir concepteurs et apprenants.

Il s'agit en fait de repenser un cycle de vie mettant en exergue les rôles des apprenants et les concepteurs/tuteurs. Par rapport aux cycles de vie des ressources que nous avons eu à traiter ou à citer (ARIADNE par exemple), si le processus collaboratif apparaît, nous n'avons pas trouvé dans les modèles proposés, une relation explicite entre les activités de navigation et les activités d'édition ou réciproquement.

1. Points de cadrage et contraintes

Dans le cadre d'un projet connexe visant l'élaboration d'un document numérique pour l'apprentissage du langage Python en autonomie (Ortiz 2012), nous disposons d'un grand

nombre (plusieurs centaines *a priori*) d'unités d'apprentissage (non encore stabilisées en terme de contenu). L'ensemble de ces unités peut se formaliser sous la forme d'un graphe, où chaque nœud est une unité et chaque arc est une dépendance, notamment de type prérequis. Ce graphe acyclique orienté (DAG) est le système nerveux de la ressource. Il éclaire les concepteurs sur la cohérence et la couverture de la ressource. Il sert aussi de carte aux apprenants pour les guider à travers la ressource vers leurs objectifs. Bien qu'il existe d'autres approches pour modéliser les relations entre les unités (approche par compétences, connaissances sous forme d'ontologies), nous nous limitons à la relation de type prérequis dans la représentation destinée à l'apprenant (parfois débutant) pour ne pas surcharger la visualisation et lui en simplifier la lecture (cf. la section II.3 du chapitre 2 de la partie II). Toutefois, ce type de graphe réexploité est de nature statique et ne permet pas une visualisation et une navigation interactive. Il est nécessaire donc de le transformer et de le rendre également sensible au traçage. Comment peut-on améliorer la visualisation générale et aider à mieux naviguer dans ces ressources ? Nos contraintes de départ font que nous ne pouvons agir ni sur la structure du graphe, ni sur les types de dépendances entre ces unités. En effet, ces aspects sont traités dans un projet de recherche indépendant et ce graphe des épistèmes nous est donné comme un élément non modifiable. Les représentations informatiques en UML ou avec des réseaux sémantiques que nous avons introduites (voir la section III du chapitre 3 de cette partie II) se basent sur ce DAG pour modéliser les différentes trajectoires des apprenants. Elles ont l'avantage, en plus d'être complémentaires de ce modèle du domaine, d'être mieux sujettes à l'adaptation et à l'évolution qu'une représentation figée d'un graphe. Nous souhaitons cependant offrir un système qui permet l'évolution d'un ensemble vivant de ressources prenant en compte l'expérience de l'apprenant.

2. Pourquoi lier édition et navigation ?

Il peut paraître peu conventionnel de vouloir créer une relation entre ces activités *a priori* séparées qui font appel à des acteurs poursuivant des buts différents. Dans la construction du graphe, le concepteur offre sa vision pédagogique d'expert. Par leurs parcours réels, leurs succès et leurs échecs, leurs retours en arrière, leurs nouveaux choix, etc., les apprenants nous livrent leur point de vue non expert et nous renseignent sur l'origine ou la genèse de leurs productions. Nous pensons que les choix de l'un et les traces des autres devraient dialoguer pour permettre une *évolution* (contrôlée) de la structure du graphe des ressources. Ce dernier devra malgré tout assurer une certaine stabilité pour ne pas perturber l'apprenant si les changements s'opéraient trop fréquemment et à son insu. Bien évidemment, à terme, cette approche doit être transposable pour n'importe quel type de graphe de ressources et par extension à tout type de ressources décrites dans la partie I.

3. Comment lier ces activités ?

Nous suggérons de tracer les actions des apprenants qui concernent notamment les consultations et validations des unités pour repérer les séquences d'unités constituant des parcours. Ainsi, les tendances dégagées permettraient d'informer statistiquement les tuteurs ou les concepteurs sur les différents parcours. On pourrait alors utiliser cette information dans l'édition/mise à jour d'une unité ou d'un parcours. Nous envisageons aussi d'offrir à l'apprenant, par le biais d'annotations, le moyen de commenter ou évaluer des unités ou des parcours proposés par les concepteurs. Ces informations (traces de navigation et annotations) peuvent servir à l'amélioration du graphe général afin d'avoir une mise à jour semi-automatique (soit par le système soit par la communauté). Nous décrirons ces idées plus en détail dans la section modélisation générale de notre proposition.

4. Modélisation générale

Notre orientation générale (Figure 26) montre une mise en relation entre les activités de conception (et d'édition) des ressources qui sont généralement gérées par les enseignants et les activités d'utilisation et de navigation qui sont accomplies par les apprenants. Les activités de conception et d'édition, même si c'est rarement le cas, doivent être fortement collaboratives car nous considérons dans notre hypothèse que les ressources de haut niveau (cours, manuels de références) que nous appelons également ressources complexes (Miled, 2011) nécessitent plus d'un rédacteur pour en assurer l'adaptation et la contextualisation. Ils sont éventuellement encouragés par des moyens incitant au partage et à la modification (par exemple à l'aide de mécanismes de points ou de récompenses pour ceux qui contribuent et interagissent le plus). La ressource éditée peut guider les apprenants dans leur utilisation. Les tendances dégagées par rapport aux objectifs de validation et de respect de parcours, seront remontées aux concepteurs, créant ainsi une dynamique itérative permettant une évolution prenant en compte des données objectives (traces de navigation) et des données liées aux interactions entre les différents acteurs (commentaires, annotations...).

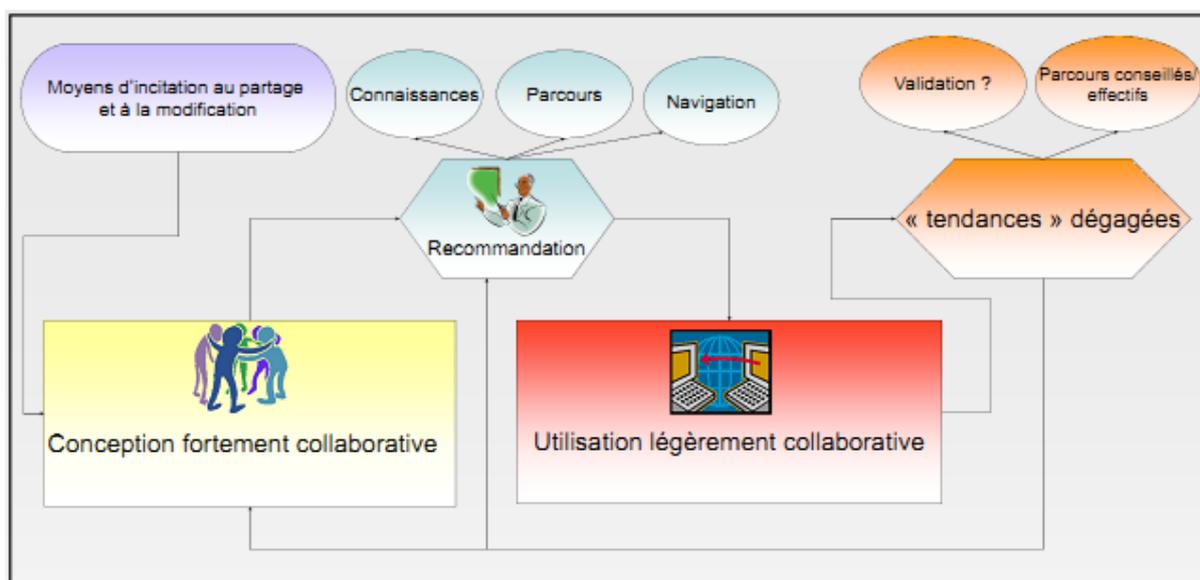


Figure 26 : dépendances entre conception et utilisation de ressources

Ce modèle schématique centré sur les acteurs peut être affiné et mieux centré sur la ressource (comme étant un ensemble potentiellement divisible) comme sur la Figure 27, où l'on discerne que l'évolution de la méta-ressource est régie par les révisions/éditions qui sont elles-mêmes déduites de l'analyse des parcours effectifs et des profils.

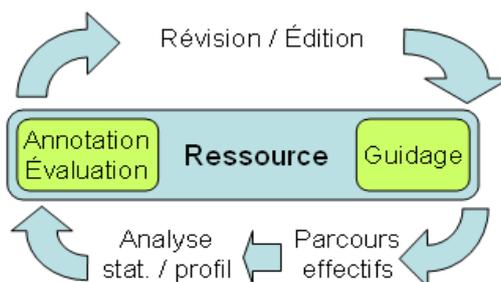


Figure 27 : relation entre activité d'édition et de navigation

III. Les apports visés de cette approche

Avec l'avènement du web 2.0, le partage d'informations et des ressources d'apprentissage paraît désormais crucial dans le développement des compétences scientifiques ou professionnelles. Même si de nombreux travaux de recherche ont abordé les questions de gestion et de partage de ressources, des freins ralentissant leur partage et leur édition subsistent encore. Ces difficultés s'accroissent avec des ressources de plus gros grain nécessitant souvent plus d'adaptation et d'évolution collaborative. Nous suggérons des indications pour améliorer la collaboration visant à mieux mettre à jour les ressources de haut niveau et ce pour mieux tenir compte de leurs évolutions. La question du partage de ressources a été abondamment traitée dans les travaux de recherche (Paquette, 2000) (J.Broisin & Vidal, 2005) (Nguyen & Bruillard 2011). Elle est renouvelée dans le contexte du Web2.0. Nous nous proposons d'explorer les pistes favorisant l'édition collective en vue de l'évolution des ressources.

Certaines ressources de haut niveau (cours, manuels de référence) requièrent plus d'efforts d'adaptation, de transformation et d'agrégation (SDTICE 2007). Face à ces constats, il est légitime de se poser les questions suivantes : Quelles méthodes permettent-elles d'aider à l'édition collaborative de ressources complexes (de gros grain) ? Comment optimiser le partage et la visibilité des évolutions ? Comment tirer parti de l'interaction entre apprenants, enseignants et contributeurs dans le processus de l'évolution de ce genre de ressources ? Il semble important de tenir compte de deux éléments : d'une part de la dimension collaborative dans le processus d'évolution des ressources, et d'autre part, de la conscience des modifications (*change awareness*) (Tam & Greenberg, 2006b). Catteau (2008b) propose un cycle de vie générique accompagné de visualisations multidimensionnelles pour la progression des objets d'apprentissage de forte granularité (modules).

Notre réflexion porte principalement sur les méthodes aidant les auteurs dans l'édition collaborative des ressources et les apprenants dans la navigation dans ce réseau de ressources. Nous structurons notre réflexion en tenant compte de trois aspects :

- la ressource d'apprentissage,
- le parcours d'apprentissage,
- et les interactions entre les apprenants, enseignants et contributeurs.

Nous pensons que l'une des difficultés liée au partage réside dans la granularité même de la ressource. Le fait qu'un contenu soit décontextualisé le rend plus facilement partageable et réutilisable. À l'inverse, un contenu dépendant d'un scénario sera plus difficile à réutiliser. Pour cette raison, nous souhaitons proposer l'accès à des ressources complexes ayant un degré suffisant d'indépendance pour pouvoir faire partie de plusieurs parcours différents. Nous définissons un parcours comme une séquence de ressources complexes (cf. Tableau 2) répondant à certains objectifs d'apprentissage. Sur ce même tableau, nous distinguons trois profils d'utilisateurs : l'apprenant, l'enseignant et le contributeur/intégrateur. Il est à noter que le contributeur/intégrateur peut avoir simultanément le rôle d'enseignant. En revanche, il est exclu pour l'instant qu'il s'agisse d'un apprenant. Sur la base de leur granularité (micro, complexe ou macro), les types de ressources sont présentés en deuxième colonne en face de chaque rôle pouvant les manipuler tandis que les actions et permissions accordées à ces rôles sur ces ressources, sont décrites dans la troisième colonne. Deux types de visualisations sont explicités : une visualisation dite statique sur des éléments inhérents à la structure générale de la ressource (macro-ressource, parcours utilisés) et une visualisation de l'évolution qui a pour

objectif de montrer les changements ou les révisions apportées ou encore les contributeurs concernés.

Profil d'utilisateur	Type de Ressources	Actions	Visualisation statique	Visualisation de l'évolution	
				ressource	Parcours
Apprenant	-Macro-ressource (tous les réseaux de ressources complexes) -Sous-ensemble de ressources complexes (séquence ou parcours) -Ressource complexe	-Accès -Navigation (traces pouvant constituer un parcours)	-Liens entre les ressources -Parcours recommandés -Parcours réalisés	Utilisateurs cibles : Auteurs, contributeurs, utilisateurs Information représentée : - Localisation relative (une micro-ressource fait-elle partie d'une nouvelle séquence ou parcours ?)	
Enseignant	complexe (agrégat de micro-ressources, cours...) -Micro-ressource (exemple, exercice, illustration...)	- Accès - Navigation - Annotation et évaluation - Proposition de parcours (séquences)	- Liens entre les ressources - Parcours généraux - Parcours selon des objectifs - Parcours préférentiels selon profils	- Auteurs, contributeurs, utilisateurs - Respect des parcours donnés pour les apprenants - Évolution des parcours suivis	
Contributeur/ Intégrateur	Micro-ressource	- Accès - Création	- Ressources (selon le niveau macro, complexe ou micro)	- Auteurs, contributeurs, utilisateurs - Nature de l'évolution (modification, agrégation, suppression) - Niveau concerné (macro, complexe, micro)	
	-Ressource complexe - Liens entre les ressources complexes	- Modification - Agrégation - Suppression			

Tableau 2 : acteurs et interventions sur les ressources d'apprentissage

Tout en permettant l'accès à chaque ressource, la navigation dans ce sous-ensemble de ressources doit être influencée par des parcours types (expériences antérieures réussies par d'autres utilisateurs) ou des parcours recommandés par les enseignants. L'édition participative et l'évaluation de la ressource et des parcours devraient constituer les clés pour une meilleure évolution et adaptation des ressources et des parcours pour mieux répondre aux besoins variés des utilisateurs. Cela peut se résumer en trois points fondamentaux :

- une aide à la collaboration,
- une meilleure mise à jour des ressources,
- et une évolutivité contrôlée de l'environnement pour les apprenants.

IV. Conclusion

Bien qu'il existe différentes approches de cycle de vie des ressources, peu s'intéressent aux stratégies à déployer et au suivi à assurer pour les ressources de haut niveau. C'est pourquoi nous avons proposé des pistes ayant comme ambition de favoriser un meilleur partage de ressources complexes afin de donner plus d'importance au caractère collaboratif des activités d'édition des ressources et mieux utiliser les traces de parcours réussis par d'autres utilisateurs des contenus. Même si ces propositions restent à affiner, à développer et à tester, nos travaux s'orientent vers une modélisation d'architecture qui offre d'une part aux enseignants le pouvoir d'influer sur les contenus et les parcours et d'autre part aux apprenants d'adapter leur parcours en fonction de leurs besoins. Cette approche peut être transposée sur n'importe quel domaine d'apprentissage, parfois nécessitant des mises à jour plus fréquentes en termes de présentation et de contenu. Nous verrons dans le prochain chapitre un cas d'application au graphe des épistèmes pour l'apprentissage du langage Python.

Chapitre 2 : Le graphe des épistèmes : le système nerveux des ressources et des parcours

I.	INTRODUCTION	82
II.	DESCRIPTION DU GRAPHE DES EPISTEMES.....	82
1.	<i>Objectifs et motivations</i>	82
2.	<i>La plateforme France-IOI</i>	83
3.	<i>Graphe des concepts et des prérequis</i>	83
III.	ADAPTATION DU GRAPHE DES EPISTEMES POUR LE CAS FRANCE-IOI	86
IV.	CONCLUSION	88

I. Introduction

Dans le chapitre précédent, nous avons présenté l'approche globale ainsi que ses principales caractéristiques. Dans le présent chapitre, nous introduisons le graphe des épistèmes pour le langage Python comme cas d'application à cette approche et comme illustration du domaine d'apprentissage lié à une introduction à la programmation et au langage Python. Nous décrirons l'environnement de ressources et de parcours et donnerons des pistes sur la mise en œuvre pour transformer ce graphe en un hypermédia à navigation dynamique individualisée. Pour décrire et modéliser le domaine, il est parfois utile d'avoir une cartographie complète du domaine, cela peut se faire à l'aide de graphes sémantiques, d'ontologies supportant des relations typées. Le choix qui s'est fait pour le graphe des épistèmes est guidé par des dépendances de type « prérequis ». La première section concernera le graphe des concepts comme ensemble de concepts inter-reliés par des relations de prérequis. La deuxième s'intéressera au cas pratique où l'on souhaiterait l'adapter à un contexte particulier et plus restreint en termes d'exercices et de concepts qui en dépendent. Nous montrerons un cas d'adaptation au contexte France-IOI d'une façon plus détaillée dans les chapitres suivants.

II. Description du graphe des épistèmes

Nous présentons dans cette section le cœur de la ressource que nous prévoyons d'exploiter et le formalisme utilisé pour la mettre en œuvre. Nous définissons le graphe des épistèmes qui a été déjà introduit dans la partie I et nous donnons quelques liens avec la théorie du cadre épistémique. Dans un second temps, nous introduisons une adaptation de ce graphe pour obtenir un graphe adapté au contexte de France-IOI.

1. Objectifs et motivations

Malgré le rôle important de la technologie et en particulier des ordinateurs dans la société, l'enseignement de l'informatique reste encore peu présent dans les écoles primaires et secondaires françaises (Baron & Bruillard, 2011). Avec l'introduction du nouveau cours facultatif intitulé Informatique et Sciences Numériques dans l'enseignement secondaire, de nouveaux besoins sont apparus en termes de préparation des enseignants et en termes d'évaluation des élèves. Dans ce contexte, la plateforme France-IOI pourrait constituer un environnement utile pour répondre à ce renouveau de l'enseignement de l'informatique et de la programmation. Parallèlement, la popularité du langage Python est notoire à travers l'émergence de cours⁷² ouverts massifs en ligne⁷³, mais aussi tout à fait visible sur les statistiques de la plateforme de France-IOI. Cette popularité le rend attrayant pour l'initiation à la programmation. À notre connaissance, peu de recherches se sont intéressées à l'étude des stratégies de résolution des exercices de programmation dans le cadre de l'option ISN⁷⁴. Pour ces raisons, nous avons intégré un hypermédia à épistèmes (Ortiz, 2012b) sur l'apprentissage du langage Python dans la plateforme France-IOI (voir le prototype HiPPY en section V du chapitre 4 de cette même partie). L'expérimentation menée nous a permis d'avoir des retours

⁷² <http://mechanicalmooc.org/>. A gentle Introduction to Python – Mechanical MOOC.

⁷³ <https://www.coursera.org/course/interactivepython>. An introduction to Interactive Programming in Python

⁷⁴ Informatique et sciences du numérique : nouvel enseignement de programmation introduit en septembre 2012 en terminale S.

sur l'évaluation des élèves par rapport à la résolution d'exercices de programmation dans ce langage.

2. La plateforme France-IOI

L'un des principaux objectifs de l'association France-IOI est de préparer les élèves français aux olympiades internationales d'informatique. La préparation se focalise sur les aspects de réflexions et les techniques d'optimisation. La plateforme supporte l'apprentissage de l'algorithmique dans sept langages de programmation (Hiron & Février, 2012). La plateforme compile et (s'il n'y a pas d'erreur à la compilation) exécute le code et compare le résultat (sortie générée) à celui attendu pour évaluer automatiquement le programme soumis. Comme pour les olympiades, le cours ISN se concentre plus sur des questions d'algorithmique que sur la programmation. Cependant, le niveau d'expertise requis est bien inférieur pour les élèves d'ISN. Pour cette raison, une séquence d'exercices (actuellement 104) incluant la présentation de concepts a été adaptée pour ces élèves. La séquence est déjà disponible sur France-IOI⁷⁵. L'ensemble des exercices est organisé en douze thèmes dont voici quelques exemples : affichage de texte, suite d'instructions, variables et calculs, lecture d'entrées, structures répétitives, chaînes et fonctions...

3. Graphe des concepts et des prérequis

Selon la théorie du cadre épistémique « Epistemic frame theory », l'utilisation de ce genre de modèle est un bon moyen pour représenter d'une manière détaillée la connexion entre les concepts supposés être maîtrisés. Cette approche est souvent utilisée en corrélation avec un courant de recherche assez récent qui est l'analyse des réseaux épistémiques « Epistemic Network Analysis » (Shaffer et al., 2009).

Cette partie peut être considérée comme « l'hyper document » ou *space document* défini dans (Henze & Nejdil, 2003). Comme introduit en partie I (chapitre 1), le graphe des épistèmes (une portion est illustrée en Figure 28), qui est un graphe orienté acyclique ou DAG (*Directed Acyclic Graph*) est le résultat d'un processus systématique guidé par quatre principes forts (Ortiz, 2012a) :

- *La saillance* : seul le concept ciblé peut être visible dans un épistème ;
- *La séparation* : chaque concept doit être présenté indépendamment de tout autre ;
- *Le rasoir d'Occam* : ne présenter que ce qui est nécessaire, supprimer tout le reste ;
- *"In code we trust"* : plus un schéma de code sera fréquent parmi tous les programmes disponibles (en ligne), plus il sera « typique ». Exemple : pour sommer des éléments, un code typique serait d'initialiser un accumulateur avant de lui ajouter les éléments à sommer dans une boucle qui parcourt la liste des éléments.

Un épistème peut occuper plusieurs pages. Il présente et commente un code typique utilisant le concept. Le cas échéant, il fera remarquer les erreurs fréquentes. Pour des concepts plus complexes, d'autres médias (vidéo, diaporama) peuvent être utilisés afin de réduire l'explication textuelle. Les épistèmes sont consultables sur la plateforme France-IOI.

⁷⁵ www.france-ioi.com

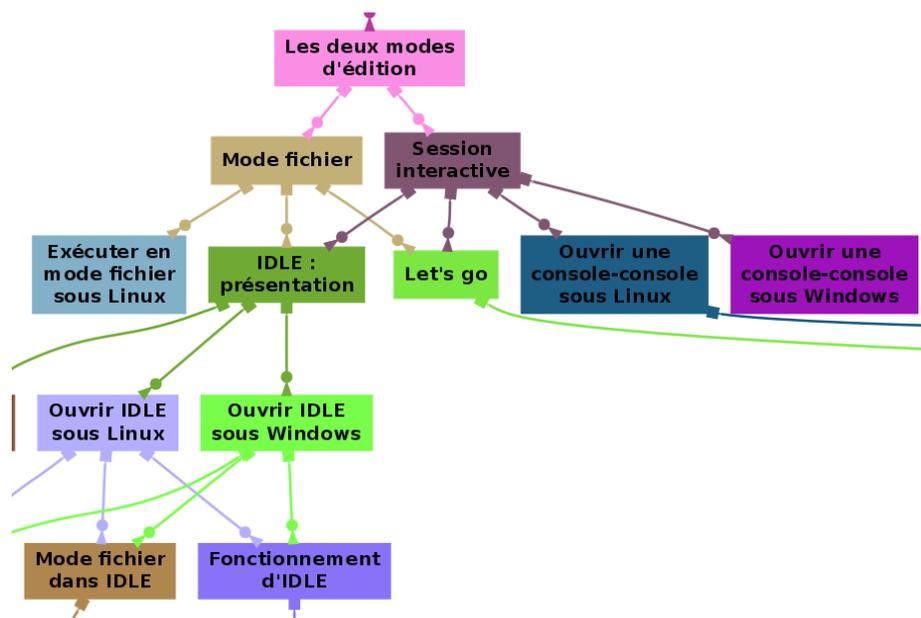


Figure 28 : illustration d'une portion du graphe des épistèmes (D'après P. Ortiz)

Chaque nœud (épistème) se concentre uniquement sur un concept très limité et le présente dans un code typique. L'orientation des arcs du graphe $A \rightarrow B$ signifie généralement : « le concept A est un prérequis pour le concept B ». Parfois, cette règle est assouplie et pourrait être interprétée comme « A est utile pour construire un exemple plus typique et complet présentant B ». À titre d'exemple, la Figure 29 présente le sous-graphe des notions utiles à la résolution de l'exercice « Labyrinthe » présenté sur la Figure 30, et dont l'objectif est de réaliser une séquence d'actions définies dans un module externe. Seul l'épistème « Importation facile et complète » est indispensable pour résoudre l'exercice. Mais l'introduction de ce dernier nécessite tous les autres. Cette même figure présente le sujet de l'exercice « Labyrinthe » contenu dans le chapitre intitulé : « affichage de texte, suite d'instructions ». Bien que le scénario propose une séquence strictement linéaire, les élèves peuvent accéder aux exercices et les résoudre dans un ordre quelconque. Chaque exercice est présenté dans une page web fournissant initialement cinq parties (ou onglets) modulaires : Sujet, Résoudre, Conseils, Activité, Correction. Pour chaque apprenant, la solution n'est délivrée (dans l'onglet « Correction » que si un code valide a été soumis pour résoudre le problème correspondant. Si un apprenant valide seulement certaines parties du chapitre, en laissant des exercices non résolus dans la séquence, il lui sera difficile d'identifier les concepts qu'il a manqués. C'est sur ce point que le graphe épistémique pourra l'aider.

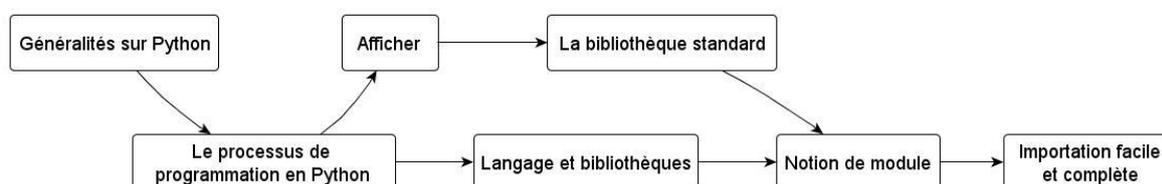


Figure 29 : explicitation du sous-graphe des notions utiles pour l'exercice « Labyrinthe »

Pour une large couverture du langage, et afin de servir comme un continuum intégré de cours d'introduction jusqu'à un manuel de référence pour Python, ce DAG devrait compter plusieurs centaines d'épistèmes. Pour cette raison, nous avons besoin d'outils pour gérer le DAG afin de construire un hypermédia adaptatif. Ces outils peuvent prendre en compte : l'édition du

graphe et des épistèmes, la navigation dans le graphe, la recommandation, la recherche d'information (par un novice qui utilise parfois des intitulés inappropriés pour les concepts), etc. Une première étude visant à sélectionner l'outil graphique de visualisation optimal pour une tâche donnée est présentée dans (Miled, 2012). Il s'agit d'un graphe regroupant des unités épistémiques (un peu moins de deux cent) basées sur l'expérience ou le « vécu » probablement en tant qu'enseignant du langage Python. L'ensemble de ces unités doit pouvoir constituer un document d'apprentissage et un manuel de référence de la découverte à l'utilisation avancée. Ces unités de savoirs (déclaratifs, procéduraux) dépendent également de la difficulté à les enseigner. Même s'il existe des liens entre ces unités, le couplage est volontairement faible pour garantir une évolutivité et une décontextualisation parfois nécessaire pour avoir plusieurs parcours possibles. L'évolution d'une unité ne doit pas avoir de conséquences sur d'autres unités. La navigation dans cet ensemble d'unités doit permettre la recherche, la visualisation d'un ensemble de nœuds (vision locale) ou un nœud donné. Les dépendances sont fortement corrélées à l'usage et la simplicité. La méthode utilisée repose sur un problème résolu et sur l'identification des unités utilisées. Il est peut être intéressant de prévoir un cheminement s'appuyant sur une modélisation cognitive, pédagogique (en termes d'unités et scénarii pédagogiques) et navigationnelle (Ernst, 2008).

>> France-IOI >> Cours et problèmes >> Affichage de texte, suite d'instructions >> Labyrinthe

← Python
Labyrinthe
Validation →

Sujet	Résoudre	Conseils	Notions associées	Activité	Correction																																			
<p>Vous arrivez devant un grand mur d'enceinte qui vous bloque dans votre escalade de la montagne. Les villageois vous avaient prévenu, et ils vous avaient indiqué que la clé de cette porte se trouve au fond du labyrinthe végétal, dont vous apercevez l'entrée sur votre droite.</p> <p>Les villageois vous ont donné la carte du labyrinthe et vous savez que de nombreux pièges y sont disséminés. Vous n'avez cependant pas assez confiance en la carte pour vous aventurer vous-même dans le labyrinthe, et décidez donc d'envoyer votre robot chercher la clé.</p> <p>Ce que doit faire votre programme :</p> <p>Votre programme doit diriger votre robot au sein du labyrinthe suivant :</p> <table border="1" style="margin: 10px auto; border-collapse: collapse; text-align: center;"> <tr><td>#</td><td>#</td><td>#</td><td>#</td><td>#</td><td>#</td><td>#</td></tr> <tr><td>#</td><td></td><td></td><td></td><td>X</td><td></td><td>#</td></tr> <tr><td>#</td><td></td><td>#</td><td></td><td>#</td><td>X</td><td>#</td></tr> <tr><td>#</td><td></td><td>X</td><td></td><td>C</td><td></td><td>#</td></tr> <tr><td>#</td><td>E</td><td>#</td><td>#</td><td>#</td><td>#</td><td>#</td></tr> </table> <p>Votre robot se trouve initialement à l'entrée du labyrinthe (la case marquée E) et il doit atteindre la case où se trouve la clé (marquée C), sans passer par les cases où se trouvent des buissons infranchissables (marquées #) ni celles qui contiennent un piège (marquées X).</p>						#	#	#	#	#	#	#	#				X		#	#		#		#	X	#	#		X		C		#	#	E	#	#	#	#	#
#	#	#	#	#	#	#																																		
#				X		#																																		
#		#		#	X	#																																		
#		X		C		#																																		
#	E	#	#	#	#	#																																		

Figure 30 : le sujet de l'exercice "Labyrinthe"

Ceci constitue un sous-graphe de ressources homogènes (en termes de structuration) susceptible d'être un parcours. Voici, à titre illustratif, l'épistème « Afficher » tel qu'il est utilisable sur France-IOI (Figure 31).

Afficher

Prérequis : [Le processus de programmation en Python](#)

Résumé : *L'obtention d'un affichage, depuis un code Python, nécessite impérativement une instruction d'affichage.*

Soit le fichier Python dont le contenu est exactement le suivant :

```
somme_sans_print.py
```

```
1 42 + 10
```

Lorsqu'on demande à Python d'exécuter le fichier `somme_sans_print.py`, Python calculera $42 + 10 = 52$ et le programme s'arrêtera là ; en particulier, Python ne permettra PAS de voir à l'écran le résultat et l'utilisateur aura l'impression que le programme `somme_sans_print.py` ne fait strictement rien. En fait, il ne fait rien de visible !

Pour visualiser un résultat ou un message, en programmation on utilise le terme d'*affichage* et on parle d'*afficher*. L'affichage se fait en général dans un terminal.

Le langage Python met à disposition du programmeur une fonction d'affichage. Cette fonction est appelée `print`.

Pour obtenir un affichage, il faut

- le demander explicitement à Python dans le code,
- utiliser une fonction que fournit la langage Python, la fonction `print`.

Ainsi, le programme

```
1 print(42 + 10)
```

affichera

```
52
```

- Ligne 1 : pour afficher un objet, par exemple le résultat d'un calcul, on applique la fonction `print` à cet objet en plaçant l'objet entre parenthèses de l'appel de la fonction `print`.

Figure 31 : l'épistème "Afficher"

III. Adaptation du graphe des épistèmes pour le cas France-IOI

Nous décrivons maintenant le processus qui nous a permis d'avoir un graphe réduit pour le contexte des exercices proposés pour France-IOI (Figure 32). Ceci sera détaillé dans la partie 3 pour les résultats de l'expérimentation.

Un premier graphe (G1) composé de deux cent vingt-et-un nœuds (et donc épistèmes) représentant le domaine de connaissances en programmation Python avait été construit (Ortiz, 2012a) en vue d'une programmation « en local ». Un sous-graphe (G2) de soixante-neuf épistèmes en a été extrait, puis a été adapté au contexte de résolution des 104 exercices (ISN) proposés par la plateforme France-IOI. Les principales différences entre G1 et G2 sont dues au contexte d'exécution qui est simplifié (en G2) par le fait que les élèves soumettent leur code à la plate-forme qui se charge de la compilation. Ceci exclut par exemple certaines notions liées à l'installation de l'environnement de compilation et d'exécution.

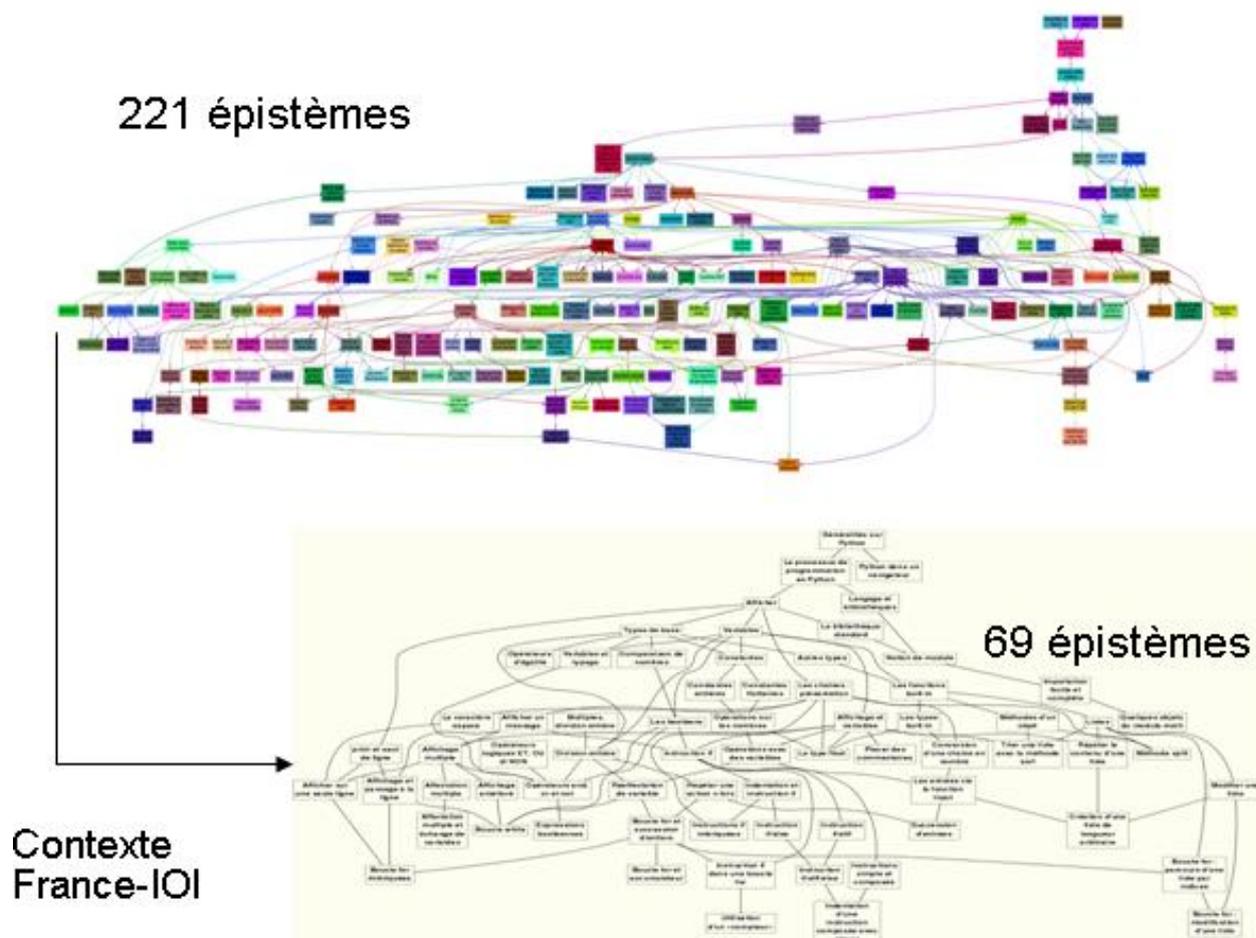


Figure 32 : construction du graphe résultant France-IOI

Une autre adaptation marginale a été introduite pour prendre en compte le changement de version de Python (2.7 à 3.1). En dehors de ces deux différences principales, G1 a été largement réutilisé pour construire G2. Donnons quelques détails sur ce processus systématique : pour chaque exercice, il suffit de fournir le code Python de (au moins) une solution acceptable. À partir de chaque code, on tâche de définir la liste des concepts / épistèmes que ce code exige. Si le concept est déjà présenté par un épistème dans le graphe, il suffit de lier l'exercice à celui-ci. Sinon, il convient de définir et d'intégrer l'épistème manquant dans le graphe selon les 4 principes mentionnés dans la section précédente. Pour construire le graphe final (G2') à partir de (G2), seulement trois nouveaux épistèmes ont dû être définis et trente légèrement modifiés (essentiellement pour prendre en compte le contexte d'utilisation).

IV. Conclusion

Pour modéliser et décrire le domaine d'apprentissage et donc identifier les compétences à partir de ces mêmes notions, il est utile de recourir à une cartographie assez exhaustive de toutes les notions potentiellement exploitables. À l'aide du graphe des épistèmes conçu pour l'apprentissage du langage Python, l'objectif était de donner une vue globale des concepts utiles ayant des relations de prérequis entre eux. Cette manière de typer les liens entre les concepts peut sembler simple. En effet, en employant d'autres types de modélisations comme les réseaux sémantiques ou les ontologies, il eût été possible d'ajouter d'autres types de relations similaires à la modélisation UML pour par exemple montrer des relations d'inclusion (ou de composition), d'équivalence, d'héritage ou de tout autre relation personnalisée. Notre choix est guidé par une meilleure lecture de ce graphe (surtout pour l'apprenant débutant) et par l'évitement de toute surcharge cognitive y compris des liens. Au niveau d'une présentation dynamique, on ne devrait pas trop surcharger avec des dépendances parfois inutiles (du moins pour l'apprenant). Cet ensemble de concepts ou d'unités (tel qu'il est présenté dans le chapitre suivant) pourra être exploité pour donner certaines propriétés selon l'utilisation de l'apprenant, sa navigation et son degré de réussite aux exercices. D'autres pistes concernant les concepteurs ou éditeurs seront données dans le chapitre qui suit.

Chapitre 3 : Conception collaborative, navigation dynamique et modélisation des trajectoires

I.	INTRODUCTION	90
II.	MODELISATION DU PROCESSUS COLLABORATIF.....	90
1.	<i>Du modèle de conception à l'édition de micro-ressources</i>	90
2.	<i>Modèle de l'apprenant, de la tâche et du domaine</i>	92
3.	<i>Modélisation des états d'une unité.....</i>	93
4.	<i>Modèle de rafraîchissement du graphe des unités selon l'influence des parcours antérieurs</i>	94
5.	<i>Mécanismes automatiques de mise à jour.....</i>	97
III.	MODELISATION DES TRAJECTOIRES	97
1.	<i>Définitions.....</i>	98
a.	La micro-trajectoire	98
b.	La macro-trajectoire.....	98
c.	La trajectoire épistémique.....	99
2.	<i>Comment mesurer et évaluer les trajectoires ?.....</i>	99
3.	<i>Modélisation des trajectoires avec des réseaux sémantiques</i>	101
IV.	INSTANCIATION DE LA DEPENDANCE ENTRE EDITION ET NAVIGATION	104
V.	CONCLUSION	106

I. Introduction

Nous avons décrit dans le chapitre précédent le graphe des épistèmes comme support de ressources et de parcours pour l'apprentissage du langage python. Nous avons montré qu'il constituait un socle prépondérant pour la construction de l'hypermédia adaptatif que nous voulons construire. Cependant, pour pouvoir mieux être mis à l'épreuve et utilisé par les apprenants, il doit avoir des propriétés intrinsèques de navigation dynamique et individualisée. Nous préciserons la nature des trajectoires que nous utiliserons à cet effet. D'autres mécanismes pour faciliter la mise à jour ou la révision doivent être également prévus notamment pour les concepteurs ou les tuteurs ou tous ceux qui auront à réutiliser une partie de cet hypermédia. Nous introduirons à l'aide de modèles informels quelques modalités visant à systématiser le processus de conception et d'édition. Nous donnerons quelques propositions sur la modélisation du domaine et des unités épistémiques ou (épistèmes) dans un contexte dynamique de navigation. Nous préciserons de surcroît des mécanismes de mise à jour et les parties concernées.

II. Modélisation du processus collaboratif

Ces mécanismes sont éventuellement encouragés par des moyens incitant au partage et à la modification. La ressource éditée peut guider les apprenants dans leur utilisation. Les tendances dégagées par rapport aux objectifs de validation et de respect de parcours, seront remontées aux concepteurs, créant ainsi une dynamique itérative permettant une évolution prenant en compte des données objectives (traces de navigation) et des données liées aux interactions entre les différents acteurs (commentaires, annotations...).

1. Du modèle de conception à l'édition de micro-ressources

Dans notre vision du cycle, de la conception et de l'édition d'une unité jusqu'à son intégration dans un parcours, exposé sur le cas d'utilisation (Figure 33), nous pouvons noter que cela implique quasiment tous les acteurs. Le concepteur écrit un patron d'unité qui sera suivi pour une majorité d'unités. Selon le besoin, une unité sera composée de micro-ressources (exercice d'application, présentation d'un concept, tests). Une fois produite, l'unité pré-utilisable sera indexée, validée et publiée pour être accessible par les apprenants, que ce soit dans un accès indépendant (unité seule) ou bien une unité intégrée dans un parcours correspondant à un objectif d'apprentissage. Si un contributeur (co-concepteur) veut apporter des modifications sur une unité, il peut générer une copie de la structure et ainsi l'adapter selon ses besoins et contextes. La gestion des droits se fait selon le modèle proposé par Connexions⁷⁶ qui stipule que le propriétaire original devient automatiquement co-auteur. On notera également, qu'il est possible d'avoir des responsables d'unités et de parcours pour mieux aider à mettre à jour et suivre les évolutions.

⁷⁶ www.cnx.org - rebaptisé openstax

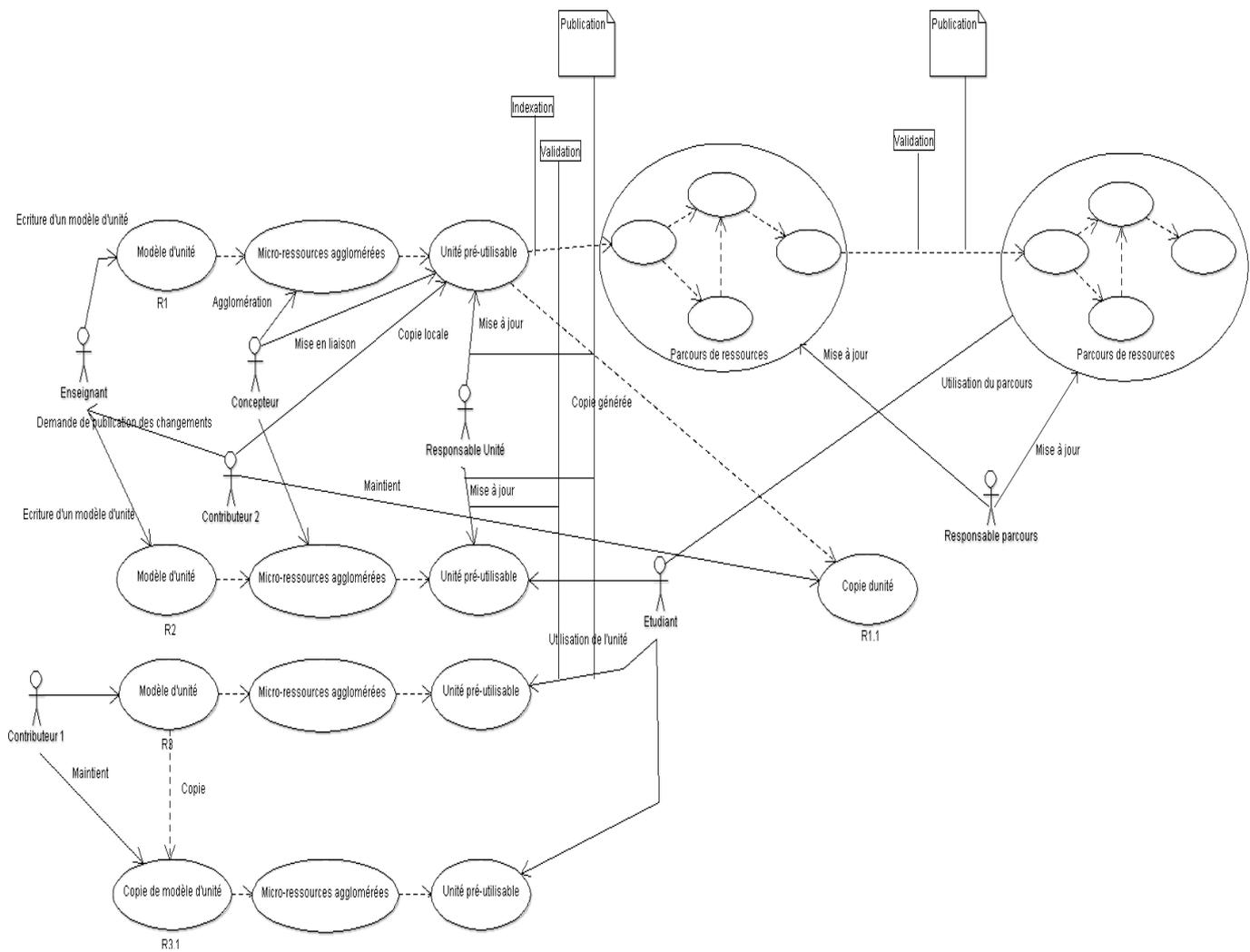


Figure 33 : processus de conception et d'édition collaborative

Le modèle formel de cas d'utilisation (présenté sur la Figure 34) illustre les différentes tâches supportées par les acteurs considérés :

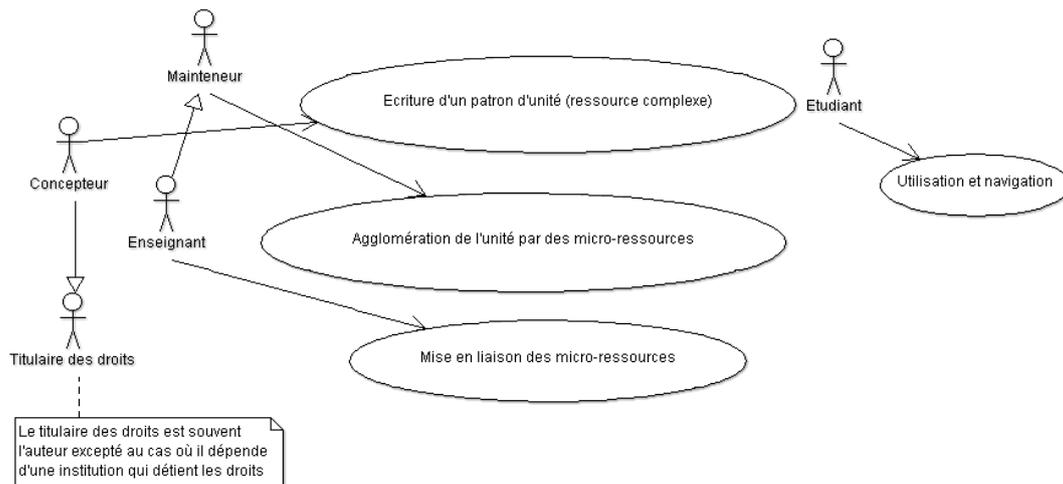


Figure 34 : diagramme de cas d'utilisation

Le processus étant fortement collaboratif, nous avons alors pensé à le modéliser à l'aide d'un diagramme de collaboration tel que présenté sur la Figure 35 :

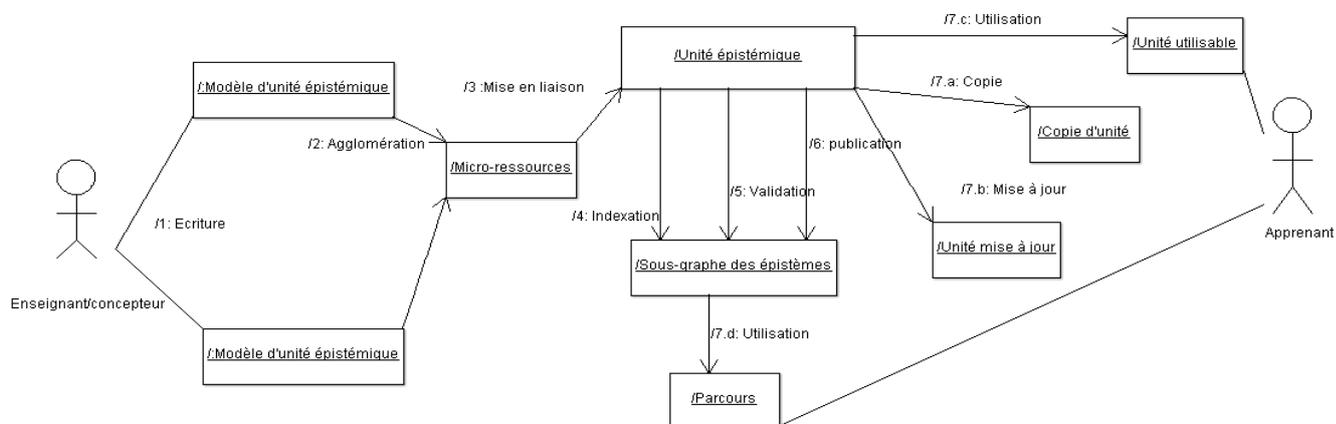


Figure 35 : diagramme de collaboration

2. Modèle de l'apprenant, de la tâche et du domaine

Hormis le modèle du domaine présenté précédemment à travers le graphe des épistèmes représentant les notions nécessaires à l'apprenant débutant (cf. le chapitre précédent de cette partie), le modèle de l'apprenant que nous utilisons ici se rapproche de celui du modèle à recouvrement (*overlay model*⁷⁷) pour ce qui est du diagnostic cognitif et basé sur les erreurs quand il est centré sur les tâches à accomplir (voir la section III). Parmi les systèmes les plus connus utilisant le modèle à recouvrement nous pouvons citer, *GUIDON*, *SCHOLAR* ou encore *WUSOR III*.

Notre proposition vise un modèle liant exercices (dans le sens des exercices France-IOI) et épistèmes. Le modèle de la tâche fait appel à des caractéristiques de difficulté variable (voir Figure 36) ou à des propriétés inspirées de la taxonomie de Bloom (i.e. compréhension, application, intégration...). Ainsi, un apprenant aura acquis un niveau de maîtrise des concepts associés selon la nature de la tâche considérée lorsqu'il l'a réussie : application directe, tâche de restitution, d'intégration ou selon la difficulté attribuée. Sur cette même figure, un apprenant consulte le contenu d'un épistème (au sens d'une unité du graphe épistémique), sachant qu'un épistème peut faire appel à un ou plusieurs concepts. Il résout un exercice qui peut faire partie d'un parcours prescrit ou suggéré par un tuteur/concepteur. Dans le cas le plus simple, lorsqu'un exercice faisant appel à un épistème est réussi par l'apprenant, le système valide les concepts associés aux épistèmes requis pour l'exercice.

⁷⁷ Représente les connaissances de l'apprenant comme sous-ensemble des connaissances expertes

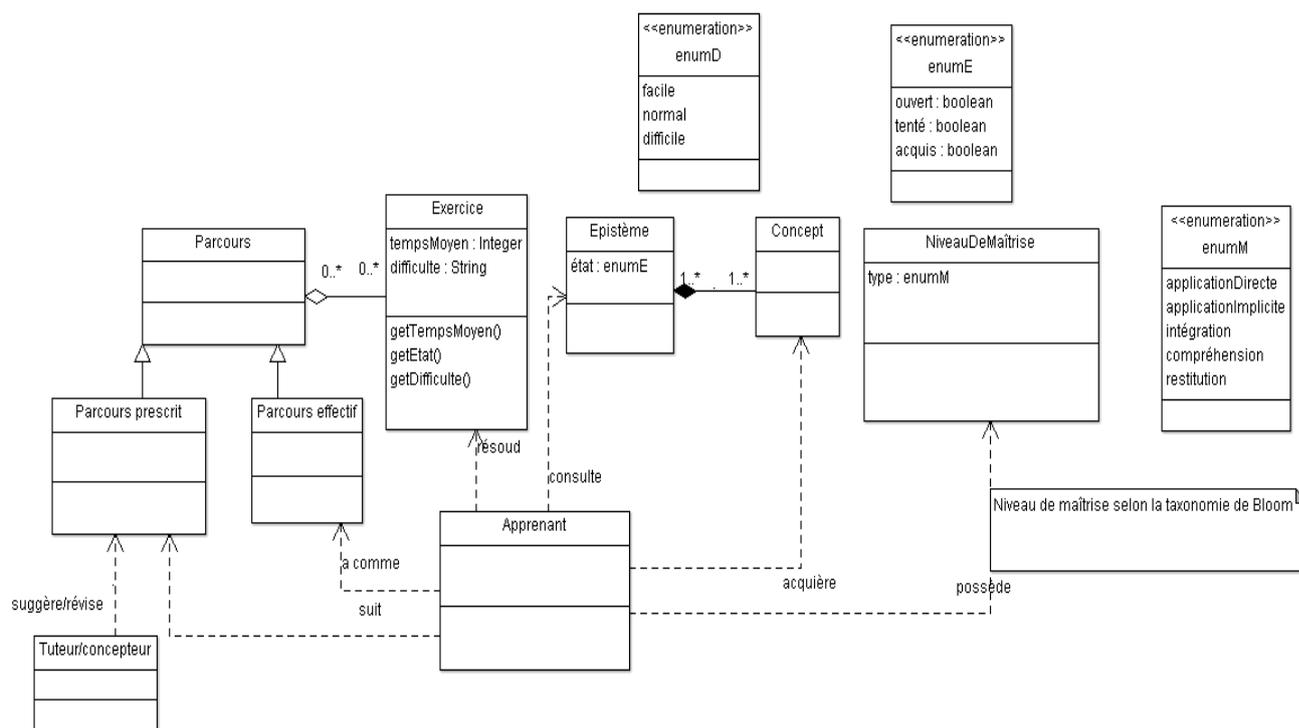


Figure 36 : exercice et épistèmes

3. Modélisation des états d'une unité

En ce qui concerne la validation pédagogique d'une unité, nous nous sommes inspirés des états fournis par ADL SCORM⁷⁸. Sur le diagramme d'état-transition (ou d'activité) ci-dessous (Figure 37), les rectangles désignent les différents états (de parcours) d'une unité. Les formes arrondies correspondent aux actions de l'apprenant. Celui-ci peut à tout moment s'évaluer (en effectuant le test) sans forcément parcourir le contenu d'une unité. Si le test est réussi, alors l'unité est réussie, sinon il est invité à obligatoirement parcourir l'unité avant de refaire le test pour le valider. Les actions suivantes sont optionnelles et consistent à soumettre du code⁷⁹ (si l'unité l'indique) et à consulter les composants de l'unité pour la parcourir en totalité.

⁷⁸ <http://www.adlnet.org/>

⁷⁹ Pour être spécifique à l'apprentissage de la programmation, nous avons ajouté un état supplémentaire (code submitted).

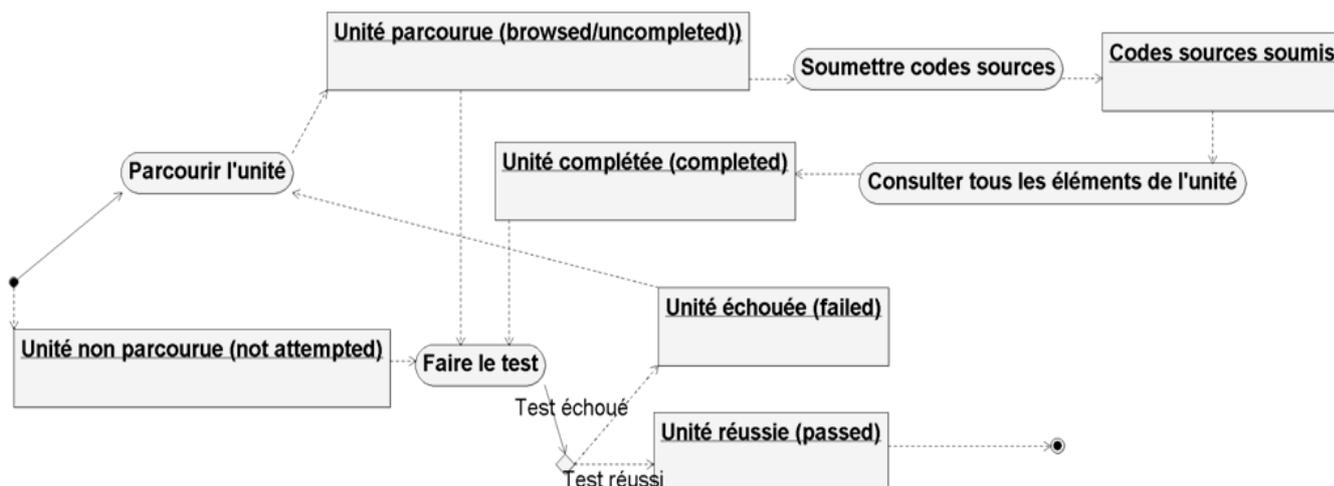


Figure 37 : états des unités

4. Modèle de rafraîchissement du graphe des unités selon l'influence des parcours antérieurs

Avant d'introduire les mécanismes de rafraîchissement envisageables pour maintenir un environnement évolutif, il est indispensable de spécifier les critères et les indicateurs qui serviront à l'évaluation. Parmi ceux-ci nous pouvons en distinguer deux sortes :

- Ceux qui s'appliquent aux unités épistémiques ou épistèmes en tant que micro-ressource, à savoir le nombre de passages (nombre d'utilisations quelle que soit l'action, lecture, ouverture simple etc..), la validation ou non de l'épistème et le nombre de tentatives.
- Et ceux qui concernent le niveau d'agrégation supérieur, en l'occurrence le parcours en tant que séquence spécifique. Dans ce cas, elle peut concerner une trajectoire élaborée de l'apprenant (voir la section III).

Ces indicateurs (sur le Tableau 3) sont certes assez généraux, mais ils peuvent être détaillés et spécifiés selon les besoins des enseignants ou des concepteurs.

Ressources	Variables / indicateurs sur l'historique
Epistèmes	<ul style="list-style-type: none"> - Nombre de passages ou visites (avec un seuil donné) - Nombre d'épistèmes validés - Nombre d'essais
Parcours	<ul style="list-style-type: none"> - Nombre de visites - Taux de respect du parcours conseillé ou recommandé par rapport à celui réalisé

Tableau 3 : indicateurs sur les épistèmes et les parcours

PARTIE II : D'UN GRAPHE STATIQUE DES ÉPISTÈMES VERS UN HYPERMÉDIA ÉPISTÉMIQUE ADAPTATIF

Chapitre 3 : Conception collaborative, navigation dynamique et modélisation des trajectoires

En utilisant ces derniers indicateurs, nous présentons sur le Tableau 4 les mécanismes de déclenchement ou de mise à jour de la ressource selon le niveau et les acteurs concernés.

Niveau de l'entité	Déclencheurs de rafraîchissement semi-automatique			Mécanismes
	Par la communauté		Par le système	
Épistème	Enseignant	Concepteur	Etudiant	<ul style="list-style-type: none"> - Nombre de visites/passages, - nombre d'essais
	<ul style="list-style-type: none"> - Commentaires sur l'unité - Evaluation d'une unité selon des critères prédéfinis : présentation, originalité, intérêt pédagogique... - Sélection d'un sous-ensemble d'unités à visualiser - Suppression d'une unité dans un clone - Fusion d'une unité avec une autre et renommage de la nouvelle unité - Masquage d'unités pour les apprenants - Recherche d'une unité épistémique (pour tous) - Validation selon le modèle (attempted, failed, passed, code submitted) 			

<p>Parcours/ ressource complexe</p>	<ul style="list-style-type: none"> - Commentaires sur le parcours - Annotations par d'autres enseignants - Evaluation d'un parcours sur des critères prédéfinis (présentation, originalité, intérêt pédagogique, cohérence, taux de réutilisation (quelles unités peuvent être réutilisées)) - Elagage (ou suppression des dépendances) - Sélection de parcours selon le profil d'application 	<ul style="list-style-type: none"> - Taux de respect du parcours conseillé/effectif (recommandé/réalisé) 	<ul style="list-style-type: none"> - Affichage des unités non suivies dans le parcours - Suppression d'un parcours - Génération de règles initiée par le système mais délimitée par les enseignants - Si le nombre de passages < seuil (treshold) ; prise en considération du changement - Sauvegarde des versions
<p>Macro-ressource</p>	<p>Profils similaires et contraintes d'édition et proposition selon les profils (domaine d'application)</p> <ul style="list-style-type: none"> - pouvoir cloner une macro-ressource (graphe complet) et modifier une copie locale 	<p>Taux de mise à jour inversement proportionnel à la taille de la ressource</p>	<p>Scinder une partie de la macro-ressource ou fusionner une autre partie selon les analyses de parcours généraux</p>

Tableau 4 : actions de rafraîchissement sur la ressource (selon le niveau et les acteurs concernés)

À partir du précédent tableau, nous pouvons résumer les aspects suivants :

- La communauté agit principalement par :
 - o des commentaires sur les unités (épistémiques), les parcours, avec possibilités d'annotations par d'autres enseignants ;
- Le système quant à lui utilise :
 - o des règles de déclenchement et des exécutions d'opérations automatisables :
 - suppression d'une ou de plusieurs unités, d'un ensemble de parcours pour l'ensemble des apprenants ou pour un groupe d'apprenants.
 - les valeurs de seuils sont tout de même délimitées par les enseignants ou les concepteurs.

Par ailleurs, la catégorisation des mises à jour doit être également prise en compte selon le degré « d'urgence » :

- imminente (ou urgente) ;
- recommandée ;
- souhaitable mais pas nécessaire ;
- optionnelle.

5. Mécanismes automatiques de mise à jour

Pour pouvoir appliquer les mécanismes de rafraîchissement de la ressource (quelle que soit sa granularité), nous pouvons faire appel à des mécanismes d'édition semi-automatiques. Nous avons vu dans la partie I (chapitre 2), qu'il y avait des outils supportés par MediaWiki, qui agissaient dans ce sens. C'est le cas par exemple de l'outil AutoWikiBrowser⁸⁰. Nous estimons que ce genre d'outil peut être utile surtout pour les tâches répétitives concernant la mise à jour du contenu de la ressource (par exemple après suppression de certains épistèmes ou dépendances, ou au contraire après l'ajout ou la fusion). Il est également possible de déléguer certaines tâches à des agents testeurs, par exemple en créant des « bots » sous PHP. Il est vrai que cela pose parfois des problèmes de sécurité car le serveur doit supporter le mode « cURL »⁸¹.

D'autres cas de modifications suite à des constats de navigation et d'utilisation peuvent parfois concerner la présentation (voir la section IV de ce chapitre).

III. Modélisation des trajectoires

Différents formalismes ont été introduits pour aider à la modélisation des processus de résolution de problèmes. Les graphes contextuels (Brézillon, 2003) peuvent être un moyen pour représenter la résolution de problèmes. Utilisant des notions de *context-awareness*, autrement dit des connaissances contextuelles et des pratiques contextuelles, ces types de graphes présentent les actions possibles avec différentes manières pour arriver à la résolution du problème.

Notre approche utilise un aspect de cette représentation dans le fait que les trajectoires les plus importantes à prendre en compte sont celles qui sont les plus fructueuses en termes de

⁸⁰ <http://fr.wikipedia.org/wiki/Wikip%C3%A9dia:AutoWikiBrowser>

⁸¹ Client URL Request Library, une interface permettant la récupération d'une ressource à partir de son URL, voir <http://curl.haxx.se/docs/copyright.html>

résolution d'une façon proche du Projet POEM⁸² (Louca et al., 2013). Il nous a semblé nécessaire d'introduire ce type de trajectoires après le processus d'adaptation du graphe épistémique (voir dans la partie II, le chapitre 2, section III) et l'intégration de l'hypermédia adaptatif HiPPY (voir la section V du chapitre 4 de cette partie II) dans la plateforme France-IOI.

1. Définitions

Considérant les traces primaires, nous pourrions les composer avec des opérations de sélection ou de filtrage afin d'élaborer des traces plus utiles et de plus haut niveau qui peuvent également être appelées traces modélisées (Djouad, Mille, et al., 2010). Dans un tel cadre, nous définissons des trajectoires à trois niveaux (Miled, Reffay, & Février, 2013):

a. La micro-trajectoire

Il s'agit de la séquence ordonnée d'événements (objets visités, soumission, navigation, etc.) pour un utilisateur spécifique entre la première visite (à $T = T_0$) de la description de l'exercice et la première soumission d'une solution correcte pour cet exercice (à $T = T_1$). Cette micro-trajectoire (Figure 38) contient en outre tous les événements concernant : la navigation dans les différents graphes d'épistèmes (notions associées, chapitre, cours complet), la visite et la validation de ces épistèmes.

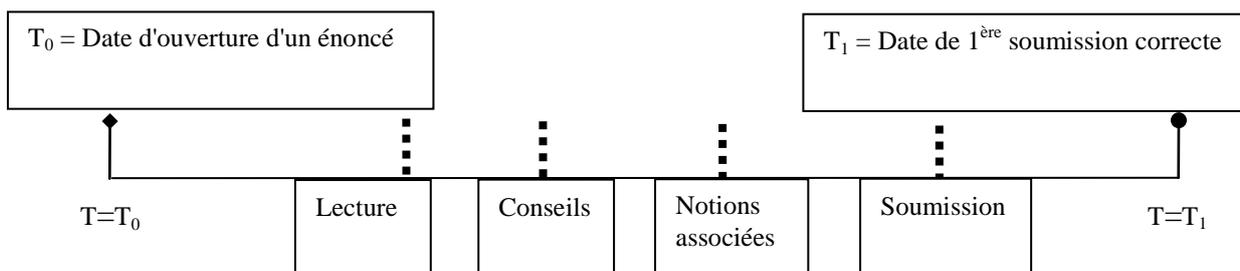


Figure 38 : structure d'une micro-trajectoire

b. La macro-trajectoire

Elle représente la séquence des exercices résolus pour un utilisateur spécifique sur une période donnée et pour un ensemble ordonné d'exercices. Comme le suggère la Figure 39, une séquence d'exercices peut être classée selon des objectifs d'apprentissage prescrits par l'enseignant ou selon des thèmes spécifiques. Par analogie aux états des épistèmes, chaque exercice peut avoir l'état visité, résolu ou en cours de résolution. Les exercices représentés par un disque blanc sont les exercices non ouverts, ceux qui ont été ouverts mais pas encore résolus contiennent des petits traits obliques et ceux qui contiennent des points sont ceux qui sont résolus. Les flèches continues désignent l'ordre séquentiel suggéré par la plate-forme. L'ordre effectif de résolution des exercices, calculable par les estampilles temporelles, peut être différent pour chaque utilisateur. Sur l'exemple de la Figure 39, les exercices E_1 , E_2 , E_3 font partie d'un thème particulier (ou chapitre faisant intervenir un ensemble d'épistèmes). E_3 , E_k , et E_t sont inclus dans l'ensemble obéissant aux objectifs décrits par l'enseignant. Cet objectif peut couvrir partiellement plusieurs thèmes. Compte tenu des instants de résolution des exercices E_i par l'apprenant A_1 notés $T_1(E_i, A_1)$, la macro-trajectoire de l'apprenant A_1 sur l'exemple serait la séquence : (E_1, E_t, E_k, E_n) .

⁸² Personalised Open Education for the Masses

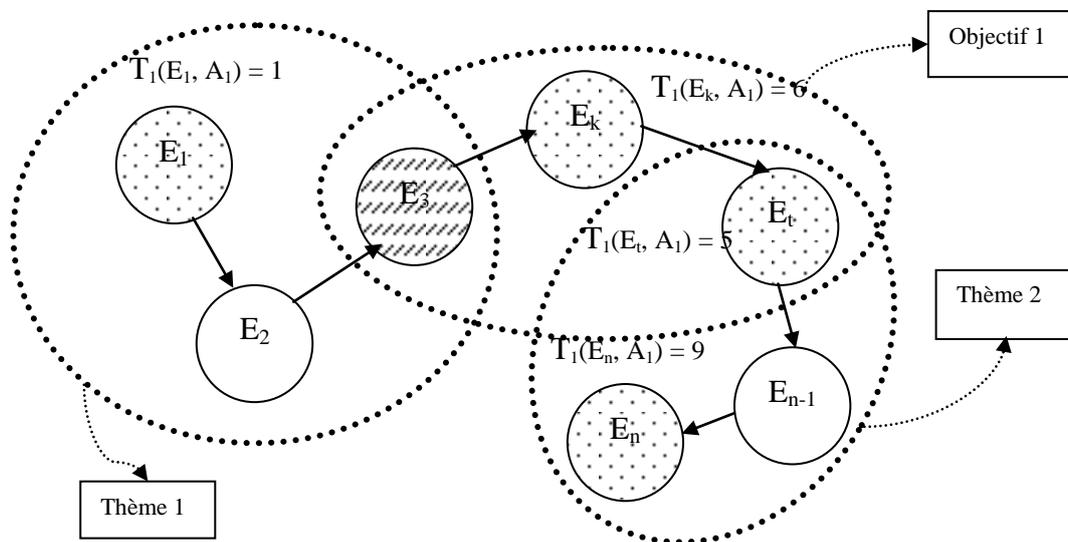


Figure 39 : ensemble ordonné d'exercices, thèmes, objectif et exemple de macro-trajectoire de l'apprenant A_1

c. La trajectoire épistémique

Cette trajectoire décrit l'ensemble des épistèmes utilisés pour un exercice donné (Figure 40). Les types d'événements concernés sont liés soit à la consultation de l'épistème (epiSeen, epiGraphDbClick, epiGraphClick, epiClick) soit à la navigation dans le graphe (recentrage de nœuds avec epiGraphClick).

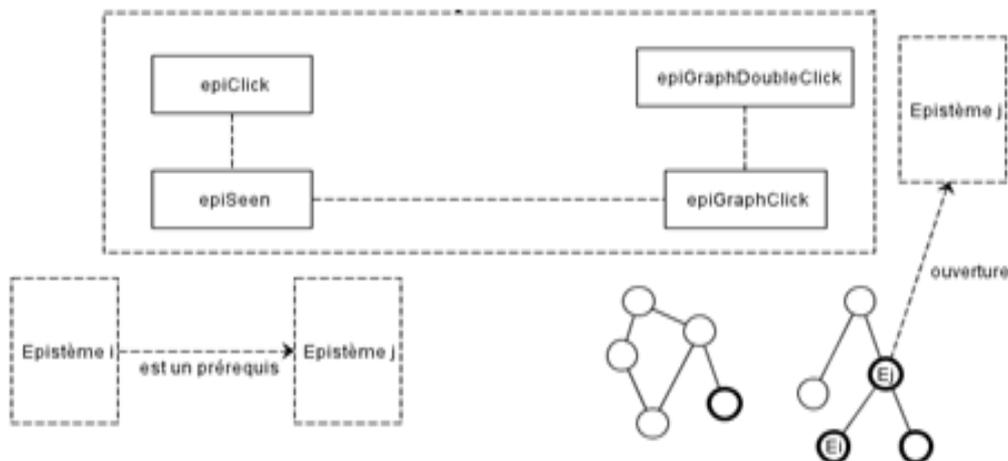


Figure 40 : composants d'une trajectoire épistémique

2. Comment mesurer et évaluer les trajectoires ?

Prenant en compte les élèves d'une même classe, les trajectoires individuelles seront comparées dans une classe afin de détecter et d'éclairer les motifs récurrents. Par exemple,

pour un exercice donné, quelques élèves auront besoin de visiter certains épistèmes (pour les aider à construire la solution) et d'autres non. Nous sommes curieux de voir s'il y a un impact sur le temps de résolution ou sur le nombre de tentatives :

- ✓ Quels sont les critères pour évaluer les trajectoires ?
- ✓ Qu'est-ce qu'une trajectoire efficace ?
- ✓ L'utilisateur a-t-il visité l'épistème recommandé par l'outil de navigation ?
- ✓ Cette visite a-t-elle eu un impact positif sur sa trajectoire ?
- ✓ Y-a-t-il une trajectoire pour des apprenants faibles et une autre pour des apprenants forts ?

Voici une vue d'ensemble des trajectoires à 3 niveaux (Figure 41) :

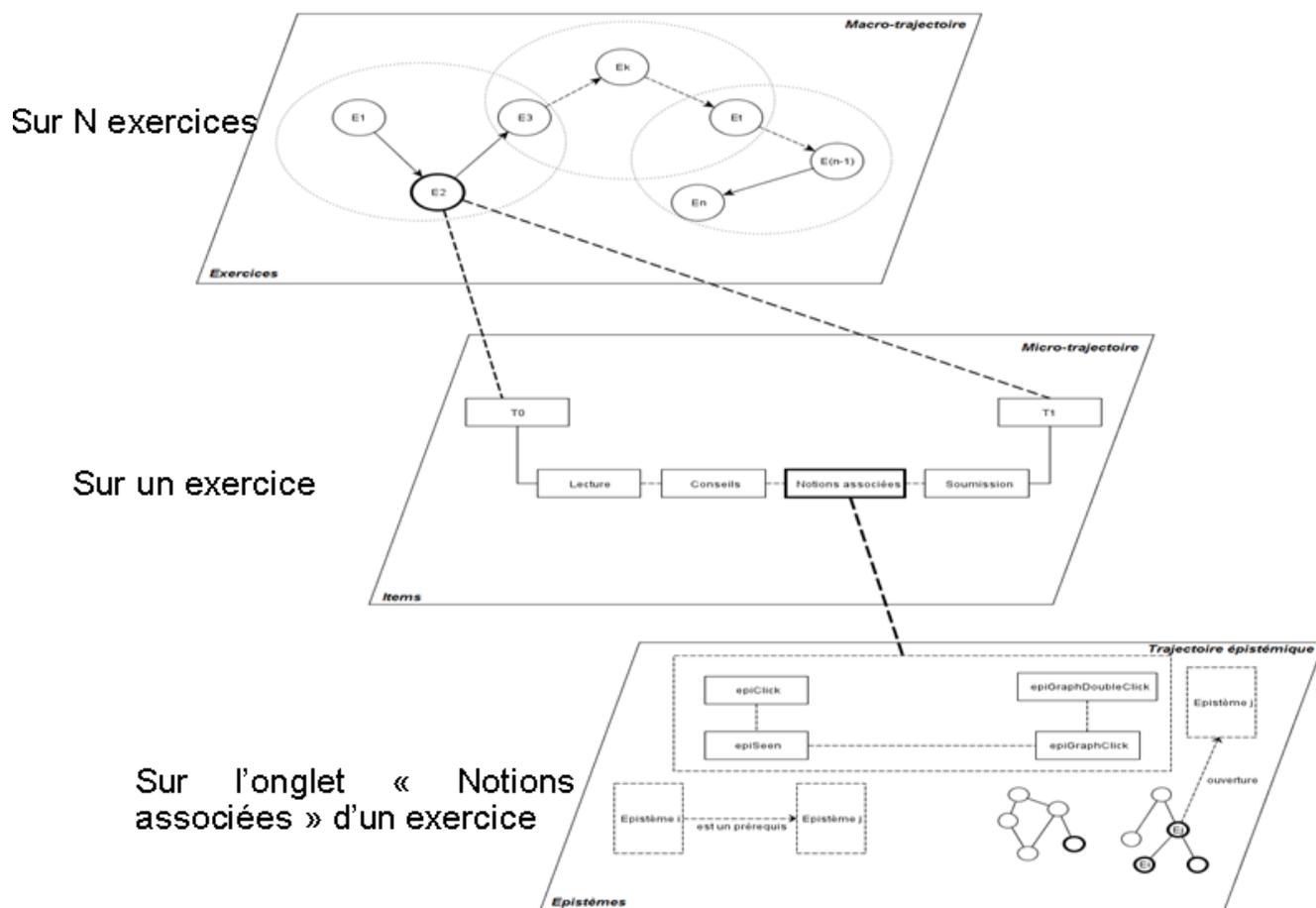


Figure 41 : structure des trajectoires à 3 niveaux

Un modèle plus formel présenté sur la Figure 42 donnerait d'une façon récapitulative le parcours effectif d'un apprenant comme faisant appel aux exercices, aux items visités et aux différents épistèmes utilisés de cette façon.

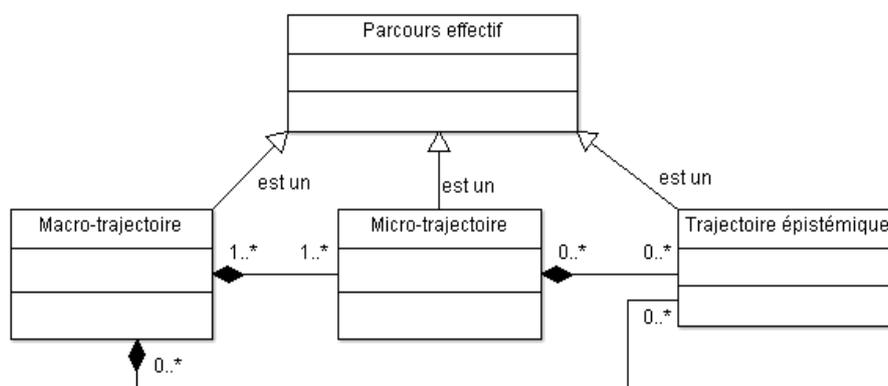


Figure 42 : modélisation d'un parcours effectif

3. Modélisation des trajectoires avec des réseaux sémantiques

Afin de représenter les trajectoires avec leurs différents niveaux, nous avons pensé utile de modéliser cela à l'aide d'une représentation sémantique RDF/OWL (Figure 43).

Nous avons dû également mettre en lien le profil d'un apprenant et ses propriétés.

Bien qu'à notre connaissance, il n'existe pas d'ontologie spécifique au cas France-IOI, nous considérons tout de même qu'il s'agit d'un réseau sémantique permettant de voir les dépendances entre les exercices, les épistèmes ainsi qu'avec les traces modélisées à l'aide de trajectoires.

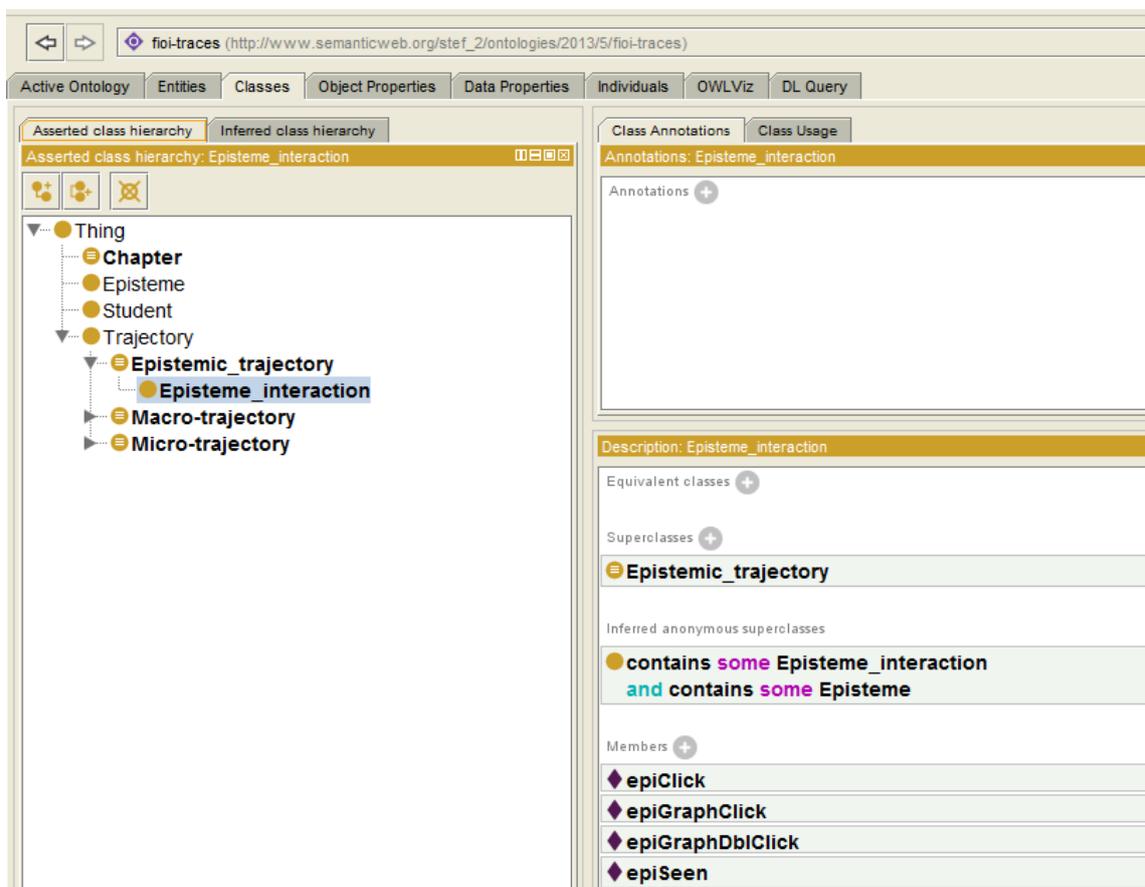


Figure 43 : réseau sémantique d'une trajectoire avec Protégé

Comme décrit plus haut, un parcours effectif (ou trajectoire) est composé obligatoirement d'au moins une macro-trajectoire ou plus, d'une micro-trajectoire ou plus et de zéro ou de plusieurs trajectoires épistémiques. Le réseau sémantique (cf. Figure 44) montre que :

- les différents exercices validés constituent une macro-trajectoire,
- les différents items ou onglets possibles font partie d'une micro-trajectoire,
- et les événements liés à l'interaction avec le graphe des épistèmes ou les épistèmes eux-mêmes sont inclus dans la trajectoire épistémique.

Le concept « Apprenant » (Student sur la figure) est relié au concept de trajectoire par la propriété `aCommeTrajectoire` « `hasAsTrajectory` ».

Ceci est une adaptation sémantique à la trajectoire à trois niveaux introduite dans la section précédente. Une vue partielle à l'aide de RDF Gravity⁸³ (en termes de concepts et de liens) sous forme de graphe est donnée sur la Figure 44. Certains liens tels que les exemples suivants ne sont pas visibles sur cette figure. Ainsi : le fait qu'un exercice nécessite des épistèmes mis en jeu (dépendances d'un exercice), qu'une trajectoire épistémique exploite des événements liés aux épistèmes ou encore qu'un parcours prescrit ou « chapitre » au sens France-IOI est une liste d'exercices. Notons par ailleurs que si un exercice réussi (*Passed*) contient dans ses dépendances des épistèmes (ex : E_{10} et E_{20}), nous considérons que ces épistèmes sont validés (i.e. : les concepts qu'il contient sont considérés acquis par l'apprenant). Si, de plus ces épistèmes (ex : E_{10} et E_{20}) sont récursivement liés à d'autre par la relation de prérequis (ex : E_{10} a pour prérequis E_3 et E_5 , E_3 a pour prérequis E_1 et E_{20} a pour prérequis E_{13}) alors tous les épistèmes liés (récursivement) sont également validés (ex : E_1 , E_3 , E_5 , et E_{13}).

⁸³ <http://semweb.salzburgresearch.at/apps/rdf-gravity/download.html>

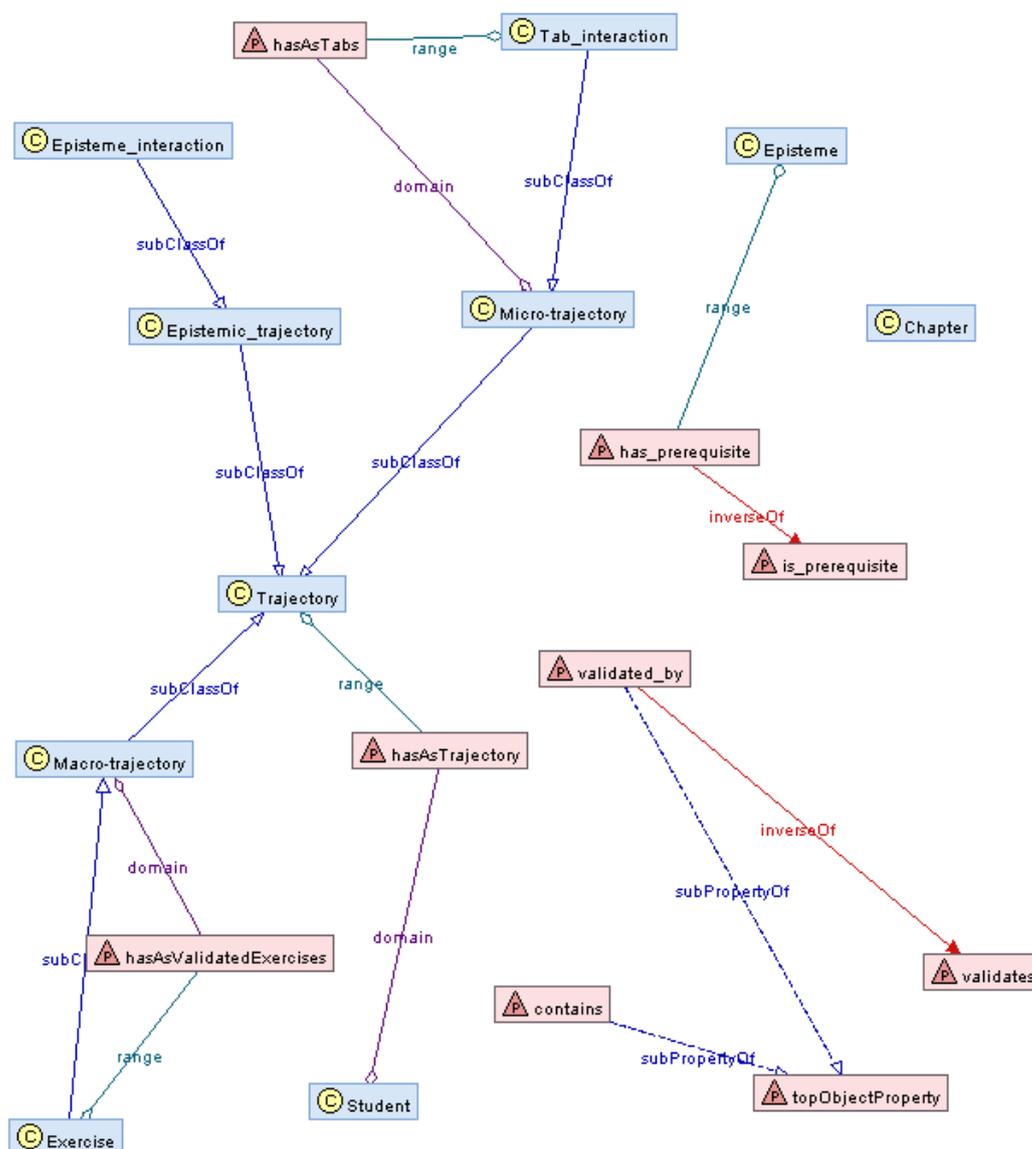


Figure 44 : graphe conceptuel partiel d'une trajectoire

Nous pouvons également avoir recours à une modélisation analytique pour lier trajectoire et apprenant. Soit $S = \{ S_1, S_2, \dots, S_N \}$ une liste d'apprenants et $E = \{ E_1, \dots, E_m \}$ la liste des épistèmes. Ci-contre un exemple de matrice donnant les épistèmes utilisés par les apprenants concernés.

$$[E_1 \quad \dots \quad E_m] \begin{bmatrix} S_1 & \dots & S_N \\ 1 & \dots & 1 \\ \vdots & \ddots & \vdots \\ 1 & \dots & 0 \end{bmatrix}$$

NB : Ceci est une correspondance entre les apprenants et les états des épistèmes (1 pour l'état validé, 0 sinon).

La fréquence d'usage des épistèmes visités pour chaque apprenant peut être également renseignée. Il est également possible de préciser la taille de la matrice (selon le nombre des élèves à suivre et la liste des épistèmes) ou même obtenir un vecteur par apprenant. Tous ces éléments seront nécessaires pour les exploiter dans l'analyse et la fouille de données sous forme de processus. (Voir le Chapitre 5). À titre illustratif, la matrice qui suit donne une fréquence d'usage des différents épistèmes : l'apprenant S_1 a utilisé 4 fois l'épistème E_1 . Par généralisation l'apprenant S_N aura utilisé k fois l'épistème E_m , avec m nombre total des épistèmes.

$$[E_1 \quad \dots \quad E_m] \begin{matrix} S_1 & \dots & S_N \\ \left[\begin{array}{ccc} 4 & \dots & i \\ \vdots & \ddots & \vdots \\ 9 & \dots & k \end{array} \right] \end{matrix}$$

IV. Instanciation de la dépendance entre édition et navigation

Comme le stipule l'approche liant les activités d'édition et de navigation, l'objectif est d'avoir des ressources dynamiques (ou évolutives) sensibles aux traces de navigation.

Nous verrons dans la partie 3, comment nous avons exploité (bien que partiellement) un trait de cette dépendance à l'aide de l'historique SVN⁸⁴. A titre d'exemple, un des éléments nécessitant la sauvegarde de plusieurs versions est le nombre des onglets (ou items) et leur ordre. L'idée étant de regarder à partir des traces apprenants certains comportements de navigation et de réguler l'interface de présentation ou de contenu selon les besoins réels ou les usages.

Comment la navigation à travers les traces peut-elle influencer le contenu ou la présentation ? Le fait d'avoir changé l'ordre des onglets (onglet notions associées) suite à ce qu'on avait observé sur les premières traces mais plus globalement, on pourrait généraliser l'impact de la navigation sur les ressources (graphe des épistèmes, exercices, ordre, présentation différente).

La Figure 45 présente le sous-modèle qui peut s'appliquer au cas France-IOI :

⁸⁴ Abréviation de Subversion, logiciel de gestion de versions.

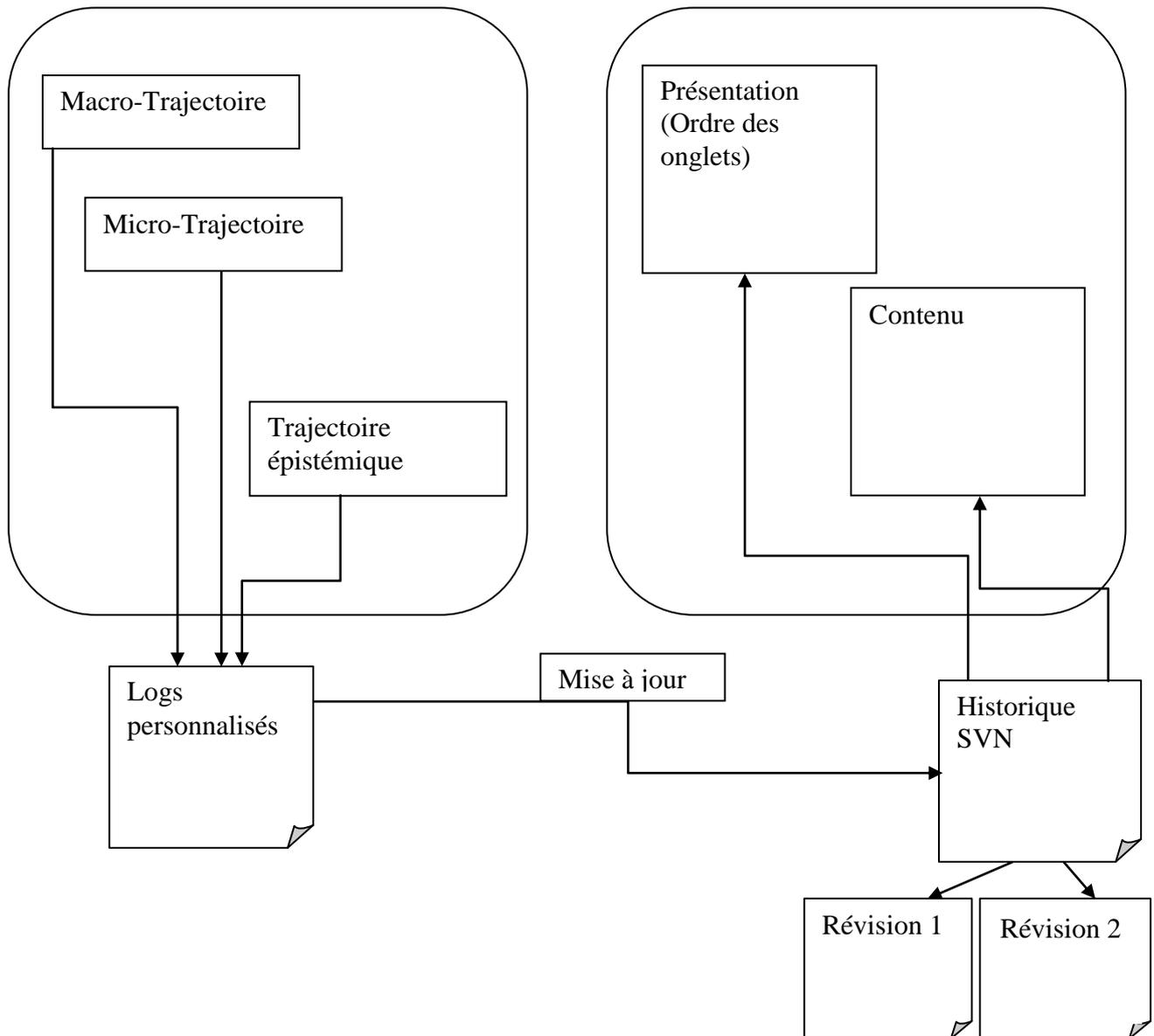


Figure 45 : processus de révision à partir des trajectoires

V. Conclusion

Nous avons présenté différents éléments ayant pour but de spécifier les aspects d'utilisation et d'édition collaborative. Nous avons présenté des moyens et des processus pouvant être utiles aux concepteurs pour éditer et maintenir l'ensemble des ressources dans un contexte collaboratif.

Pour ce qui est du modèle de l'apprenant et de ses interactions avec les tâches du domaine, nous avons opté pour un modèle reposant sur un modèle de recouvrement utilisant un sous-ensemble des connaissances expertes. Nous avons explicité la représentation des notions atomiques et leurs différents états selon leur utilisation par l'apprenant. Ces états se basent sur ADL SCORM auquel s'ajoute un état supplémentaire spécifique à l'apprentissage de la programmation qui est la soumission du code source. Cet élément nous a permis de faire le lien avec les parcours et les interactions entre les unités épistémiques à travers la définition des trajectoires à trois niveaux.

Nous tenterons de mettre en œuvre ces principes et ces choix de modélisation pour implémenter les prototypes. Cela fera l'objet du chapitre suivant.

Chapitre 4 : Architecture générale des prototypes

I.	INTRODUCTION	108
II.	UN OUTIL DE VISUALISATION COMME PIVOT A LA NAVIGATION DYNAMIQUE	108
1.	<i>Objectifs</i>	108
2.	<i>Intégration d'un outil de visualisation</i>	108
3.	<i>Test d'outils de visualisation des graphes</i>	108
III.	UNE MISE EN APPLICATION DES EPISTEMES SUR CLOUDSCORM.....	111
IV.	PROTOTYPE PRECEPTOR	112
1.	<i>Fonctionnalités et utilisateurs concernés</i>	112
2.	<i>Traces liées à l'édition et à la navigation</i>	113
3.	<i>Choix de l'implémentation</i>	114
4.	<i>Accès aux fonctionnalités</i>	116
a.	Dérivation d'une copie/Structure	117
b.	Evaluation des unités et des parcours.....	118
c.	Edition de métadonnées	119
V.	PROTOTYPE HIPPY	120
1.	<i>Modèles de données et éléments d'implémentation</i>	120
2.	<i>Navigation dans l'hypermédia épistémique</i>	122
3.	<i>Mise en place de l'évaluation épistémique individualisée</i>	124
a.	Modèle d'un épistème et ses primitives associées.....	124
b.	Mécanisme de traçage.....	125
c.	Individualisation de l'évaluation épistémique.....	125
d.	Construction de la fermeture transitive	126
VI.	CONCLUSION	128

I. Introduction

Du point de vue de l'utilisateur, il est parfois difficile de naviguer dans un grand ensemble de ressources, même s'il s'agit de liens hypertextes. Du point de vue du concepteur, l'édition de ressources de haut niveau que nous appelons également « ressources complexes » (cours, manuels de référence) nécessite des efforts et plus de collaboration pour en garantir la cohérence et éviter les redondances inutiles. Nous tenterons de mettre en œuvre un principe de navigation basé sur la visualisation dynamique de graphes.

L'objectif de ce chapitre est de tester la faisabilité des principes énoncés au chapitre précédent, à savoir les processus d'édition collaborative, la mise en œuvre du modèle à recouvrement de l'apprenant ainsi que ses interactions à travers les trajectoires spécifiques que nous avons définies. Nous présentons la mise en application de la navigation dynamique et la visualisation interactive grâce à la sélection de l'outil de visualisation des graphes. Nous testerons l'exécution et la navigation dans les épistèmes à l'aide de la plateforme CloudScorm illustrant un fonctionnement sur un LMS. Dans un second temps, nous décrirons le prototype PRECEPTOR et dans un dernier volet le prototype HiPPY dans le contexte de la plateforme France-IOI.

II. Un outil de visualisation comme pivot à la navigation dynamique

1. Objectifs

Nous souhaitons proposer une alternative de navigation surtout pour des ressources inter-reliées (ayant des objectifs communs) proposant une vue à la fois globale (grand graphe) et une vue locale (sur les parties qui concernent plus la tâche courante de l'utilisateur).

2. Intégration d'un outil de visualisation

Une carte de navigation est particulièrement utile aux apprenants pour naviguer dans les contenus pédagogiques (Ernst, 2008). Pour notre part, nous pensons que par extension, et avec le développement des technologies dynamiques de type HTML 5 sur tout support et plateforme d'exploitation, le recours aux cartes de navigation dynamiques pourrait constituer un atout.

3. Test d'outils de visualisation des graphes

Nous avons préalablement implémenté une portion du graphe des épistèmes sur Dhoston Springy⁸⁵ (cf. Figure 46). Même si cette bibliothèque présente de bonnes possibilités d'interaction (animation, déplacement selon le souhait de l'utilisateur), une des limites constatées est qu'elle ne permet pas une personnalisation adaptée, par exemple la vue peut devenir vite complexe et donc difficile à lire quand le nombre de nœuds et de liens est trop

⁸⁵ <https://github.com/dhoston/springy>

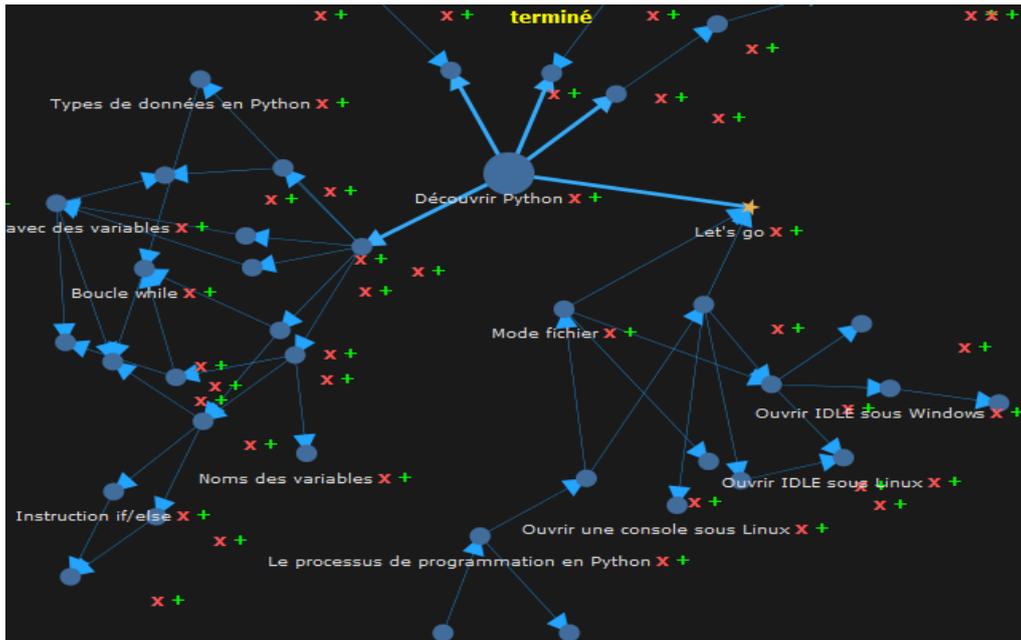


Figure 47 : une portion du graphe des épistèmes avec ForceDirected

Toutefois, là-aussi, il peut y avoir rapidement des problèmes de surcharge de vue avec un nombre plus important de nœuds.

L'utilisation de HyperTree (cf. Figure 48) quant à elle, permet de réduire le problème de surcharge en grisant les zones assez éloignées.

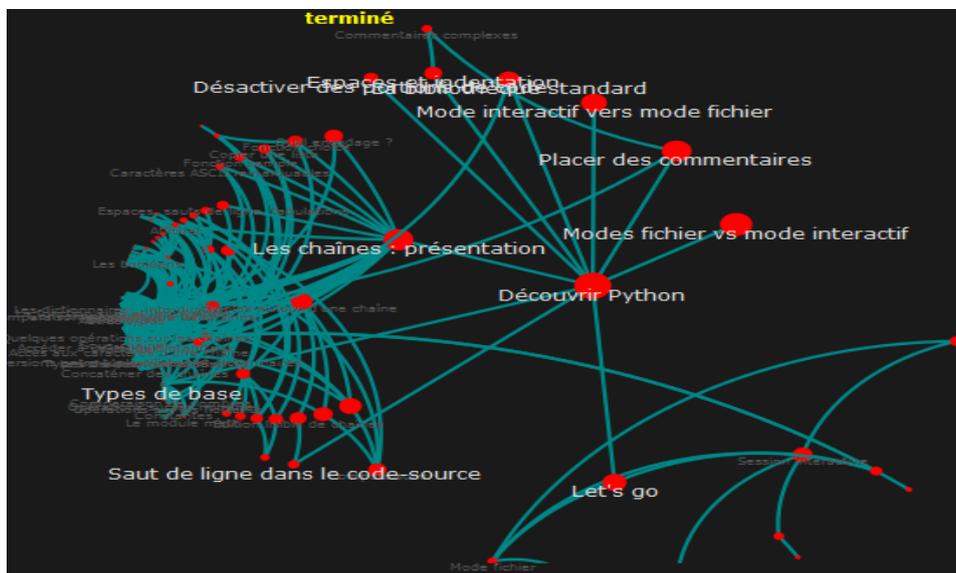


Figure 48 : une portion du graphe des épistèmes avec HyperTree

Cependant, comme le mentionne la structure, il s'agit d'un type créé nativement pour des arbres, et il devient difficile de transformer la vue en la rendant orientée. Nous verrons plus tard que le type RGraph (Figure 49), qui sera retenu pour le prototype HiPPY, peut être un compromis pour pallier certains problèmes de surcharge et de représentation des graphes acycliques orientés.

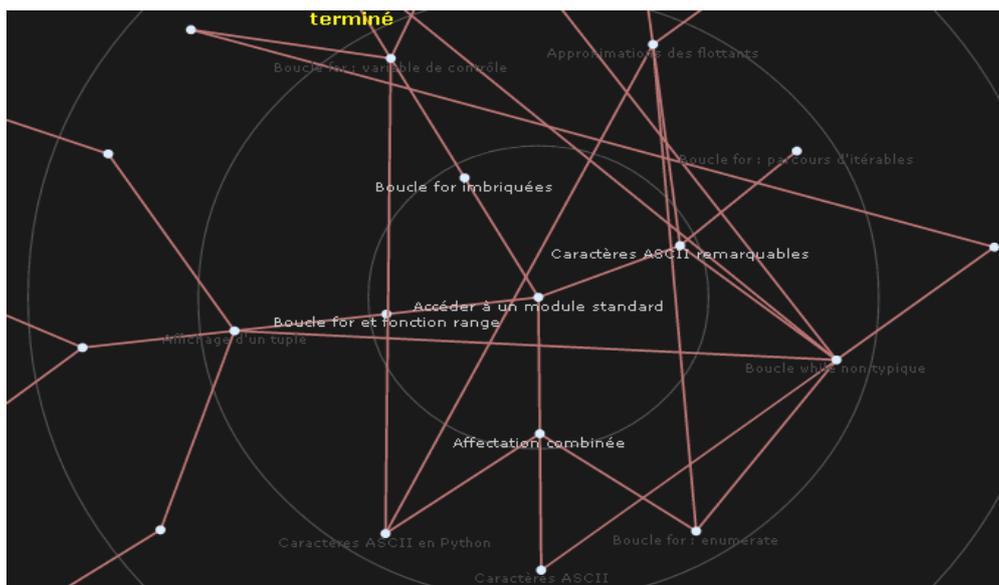


Figure 49 : une portion du Graphe des épistèmes avec RGraph

III. Une mise en application des épistèmes sur CloudScorm

La première étape a été de transformer une liste d'épistèmes en format exécutable sur les plateformes d'enseignement en ligne ou Learning Management System (LMS). Cette étape, hormis la possibilité de naviguer entre les différents épistèmes, nous a permis de rendre compte des états que pouvait avoir une unité épistémique. D'une manière simplifiée certes, cela nous a montré que nous pouvions exploiter les états de SCORM.

La séquence des unités (trente-quatre environ) python a été testée sur un LMS de type Claroline (Figure 50). Elle a été mise à la norme SCORM pour pouvoir être lue et importée. Elle a également été testée sur cloud.scorm.com. Le processus de « scormisation »⁸⁶ nécessite une archive compatible SCORM 2004 contenant un fichier manifest `imsmanifest.xml` (voir annexe III). L'éditeur Reload Editor⁸⁷ a été exploité pour rendre les épistèmes exécutables sur n'importe quelle plateforme compatible SCORM.

⁸⁶ plus d'informations sur [http://doc.claroline.net/en/index.php/How do I create SCORM content%3F](http://doc.claroline.net/en/index.php/How_do_I_create_SCORM_content%3F)

⁸⁷ <http://www.reload.ac.uk>

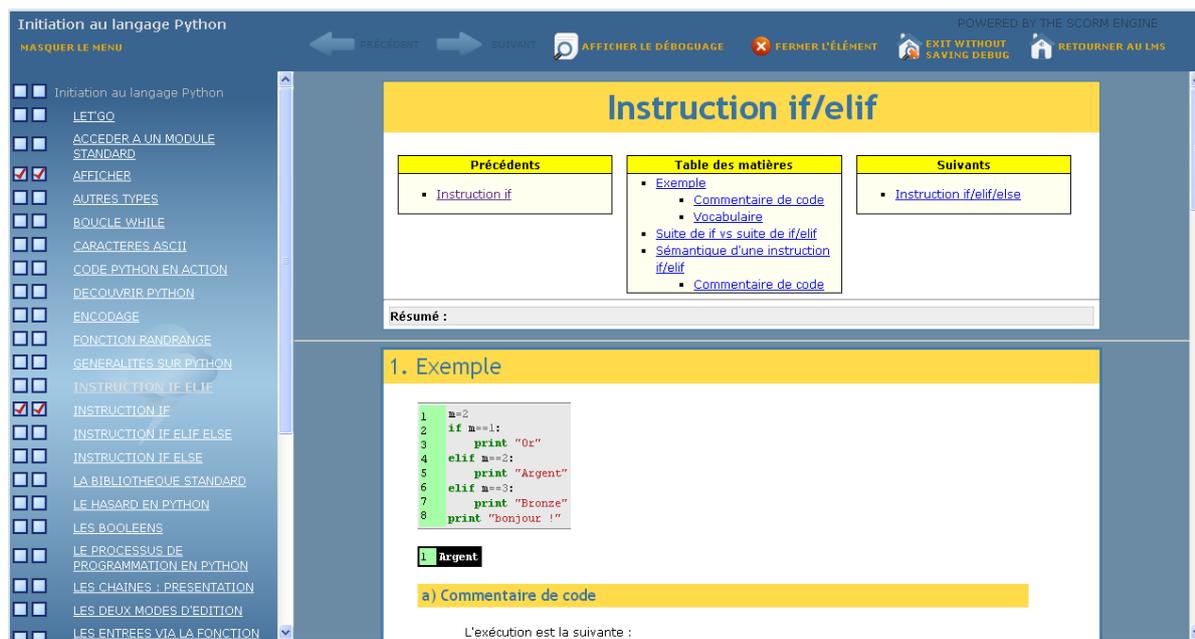


Figure 50 : liste d'unités épistémiques sur l'environnement CloudScorm

Le menu navigation se trouve à gauche et désigne l'ensemble des unités (ou épistèmes) potentiellement utilisables. Il s'agit d'une séquence simplifiée dénommée « initiation au langage Python ». L'état de chaque épistème renseigne s'il a été ouvert (cases à gauche de l'intitulé). Une table des matières interne se trouve en haut de l'épistème ainsi que les dépendances directes de chacun, que ce soient les prédécesseurs (prérequis) ou les successeurs ; à savoir qu'il est possible de caractériser la séquence (au sens SCORM) avec un seul SCO ou bien avec plusieurs. C'est pour ce dernier choix que nous avons opté puisque nous devons avoir pour chaque épistème un SCO correspondant et les traces associées.

IV. Prototype PRECEPTOR

Le principal objectif derrière l'élaboration d'un prototype qui va au-delà d'une simple séquence d'épistèmes au format SCORM, était de mettre à l'épreuve les types de visualisation, de vérifier la bonne intégration dans une application basée sur le web ainsi qu'assurer le mécanisme de suivi et de traçage à l'aide de cette bibliothèque. Il s'agit donc d'une nouvelle présentation dynamique des épistèmes avec la possibilité, en plus de pouvoir consulter les contenus correspondants, de pouvoir exécuter du code en langage Python grâce notamment à un service externe. Nous précisons que le but n'était pas de le soumettre aux utilisateurs et de l'expérimenter à grande échelle, mais plutôt de tester la faisabilité des principes énoncés précédemment.

1. Fonctionnalités et utilisateurs concernés

Les fonctionnalités supportées sont variables selon les privilèges d'utilisation et selon les types d'utilisateurs.

Cette plateforme supporte trois profils d'utilisateurs avec des privilèges graduels (étudiant, tuteur et enseignant/concepteur) :

- **Étudiant :**
 - Navigation dans le graphe des épistèmes avec ouverture des contenus des épistèmes et la possibilité de tester les codes des exemples par délégation sur un interprète⁸⁸ externe ;
 - exportation d'un nombre limité d'épistèmes sur une archive SCORM (exécutable sur un LMS) qui désigne une séquence minimale permettant de résoudre l'exercice type « plus ou moins » (cf. section VII.3 du chapitre 1 de la partie I);
 - éléments déclaratifs de validation.

- **Tuteur :** ayant les mêmes prérogatives qu'un étudiant, il peut également procéder à :
 - l'évaluation des unités et des parcours ;
 - et à l'affichage des tendances sur des graphes.

- **Enseignant/concepteur :** ayant les mêmes permissions que le tuteur, il a en plus la possibilité de procéder à :
 - l'édition d'une structure de graphe en JSON⁸⁹ : opération sur les graphes (ajout, suppression, agrégation, copies locales ...) ;
 - et à l'édition de métadonnées concernant les unités épistémiques (ou épistèmes).

Il est à noter la possibilité pour l'enseignant de modifier le contenu d'un épistème mais cette fonctionnalité n'est pas encore prise en charge.

2. Traces liées à l'édition et à la navigation

Le prototype⁹⁰ que nous avons développé sous Zend framework 1.11.3 propose de tracer les activités relatives à la navigation dans les ressources et d'autres spécifiques à leur édition (Figure 51). Nous avons opté pour Zend Framework pour plusieurs raisons. Parmi les principales raisons : nous pouvons citer le fait qu'il adopte une conception modulaire (Model-View-Controller) où il y a séparation entre la couche présentation (Vue), la couche métier (contrôleur) et la couche persistance (données). Ceci rend le code plus réutilisable, plus facile à entretenir et plus souple dans le développement. Ce framework open-source dispose d'une communauté importante et d'une documentation conséquente. D'autres points notables concernent : le support d'un grand nombre de bases de données (MySQL, Oracle, SQLite, PostgreSQL), la richesse de ses bibliothèques, le rythme assez fréquent des sorties (releases). Ces traces sont sauvegardées dans des tables MySQL. En termes de navigation, le mécanisme de génération récolte l'ensemble des unités visitées qui représente le(s) chemin(s) parcouru(s) (parcours effectifs) par l'apprenant. L'interface JIT qui est couplée avec le framework Zend doit permettre aussi de renseigner d'une manière fine sur les états de validation d'une unité (cf. Figure 37). Cette fonction n'est pas encore supportée. La mémorisation des activités d'édition permet de lister les évaluations des unités ou des parcours (suite d'unités), de modifier les métadonnées relatives à une unité donnée. La génération d'une copie/structure d'un graphe conformément à notre approche ainsi que l'exportation vers une archive SCORM sont

⁸⁸ www.codepad.org

⁸⁹ JavaScript Object Notation, format d'échange de données qui a l'avantage d'être facile à lire par les humains.

⁹⁰ Disponible en ligne sur www.preceptor-python.net

incluses. Ces éléments sont nécessaires pour pouvoir gérer l'évolution de la structure globale du graphe. Ainsi, nous pensons qu'une évaluation d'une ou plusieurs unités aura un impact sur l'évolution globale du graphe. Par exemple, si N évaluations globales (comportant plusieurs critères) font ressortir que l'unité U et que le parcours P doivent être modifiés et mis à jour, selon le degré d'imminence de la mise à jour, cette modification peut être faite soit par les contributeurs (ne concerne pas les apprenants) si cela concerne le contenu de l'unité, soit automatiquement s'il s'agit de réordonner un parcours. Une partie du graphe des épistèmes implémenté à l'aide de la bibliothèque JIT est visible sur la Figure 55 à la page 116.

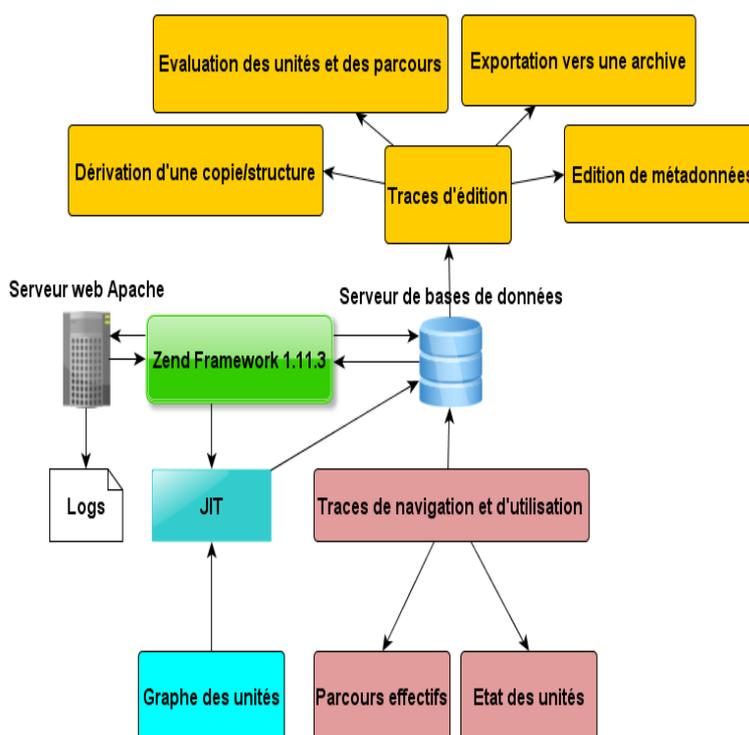


Figure 51 : processus de génération des traces d'édition et de navigation

Afin de pouvoir tester les fonctionnalités de base de notre prototype, nous avons eu recours à des tests automatiques pré-écrits. Il s'agit en fait d'un agent testeur (« *bot tester* »). Ce *bot* peut s'inscrire tout seul et évaluer une ressource. Par contre, lorsqu'il s'agit d'actions plus complexes comme la navigation et la sélection d'unités à visiter cela reste pour l'instant difficile à gérer. Une solution a été alors d'impliquer de vrais utilisateurs sur ces aspects de tâches. Nous avons eu l'occasion de le faire notamment en expérimentant le prototype HiPPY (présenté à la section V).

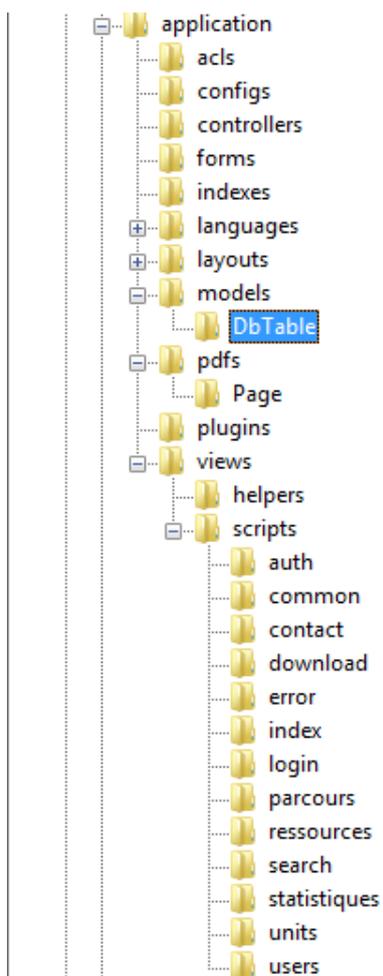
Le processus de traçage implique de stocker les données sur une base de données. Les traces d'édition sont directement capturées à l'aide du serveur. En revanche, les traces de navigation et d'utilisation font appel à des requêtes AJAX, puisque le graphe des épistèmes est matérialisé par la bibliothèque JIT qui utilise une interface Xhtml/JavaScript.

3. Choix de l'implémentation

Dans le but de mettre en œuvre les différentes propositions de modélisations de l'environnement que nous souhaitons développer, nous avons choisi de réaliser une

application web. La finalité étant de la publier sur le réseau internet afin qu'elle soit accessible de n'importe où.

Le choix s'est porté vers une application en langage PHP côté serveur et reposant sur la bibliothèque JQuery⁹¹ en JavaScript (côté client). Nous avons souhaité nous appuyer sur un framework de développement. Un framework ou cadre d'application peut être vu comme étant un ensemble de bibliothèques permettant un développement rapide d'application. Conçu d'une façon modulaire, il permet d'utiliser des composants (briques) qui sont en interaction les uns avec les autres.



Il en existe un certain nombre comme CakePHP⁹², CodeIgniter⁹³ ou encore Symfony⁹⁴.

Même si l'utilisation d'un framework peut faire gagner du temps (du moins pour une application conséquente) en tirant parti des connaissances déjà existantes, il est parfois déroutant de se familiariser avec l'architecture et le fonctionnement interne du framework. L'intégration d'un framework comme Zend, peut nécessiter également une longue configuration (voir annexe IV.1)

Figure 52 : structure de l'application Preceptor

L'application cible est une application 3-tiers en PHP/MySQL. Nous avons intégré des modules pour aider à appliquer certaines orientations de nos modèles : TinyMCE pour aider à écrire/éditer une unité préexistante; JSONEditor pour éditer le modèle de données du graphe. La structure arborescente (Figure 52) de l'application se compose des dossiers suivants : acls pour la limitation des accès (utile pour distinguer les privilèges des différents profils d'utilisateurs), configs, controllers pour les contrôleurs, forms pour les formulaires, indexes, models pour les modèles, languages pour gérer les aspects

linguistiques, layouts, pdfs, plugins, views. La persistance des données se fait à l'aide d'un mapping PDO⁹⁵ qui reprend les modèles utilisés et fait la correspondance avec la base MySQL. Cette plateforme à caractère expérimental permet de créer un compte s'il le faut et d'accéder par la suite selon le profil de l'utilisateur aux fonctions ciblées. Certaines des fonctionnalités présentées n'ont pas été menées à leurs termes comme l'édition du modèle du graphe des épistèmes, il est seulement possible d'afficher le modèle correspondant en JSON mais rien n'empêche de réutiliser cette structure ailleurs.

⁹¹ <http://jquery.com/>

⁹² <http://cakephp.org/>

⁹³ <http://ellislab.com/codeigniter>

⁹⁴ <http://symfony.com/>

⁹⁵ PHP Data Objects, a l'avantage d'accéder aux bases de données en faisant abstraction du SGBD utilisé.

4. Accès aux fonctionnalités



Figure 53 : page d'accueil du prototype PRECEPTOR

```
if(n.selected){
// alert(str);
$.ajax({
  type: "POST",
  url: "ressources/track",
  data: "name="+node.name,
  // datatype: "String",

  success: function(msg){
    alert( "Data Saved: " + msg );
    //$('#right-container').html(str);
  }
});
```

Figure 54 : requête Ajax de traçage

L'accès au graphe des épistèmes se fait à l'aide de l'onglet ressources et l'ouverture d'un épistème par un double-clic sur l'intitulé. Les petits symboles « + » et « x » (Figure 55) permettent d'adapter la vue en faisant respectivement apparaître et disparaître les liens de chaque nœud épistème. Lorsqu'un utilisateur utilise et ouvre un épistème, son action sera enregistrée à l'aide d'un appel Ajax (Figure 54) de type POST et sera transmis au contrôleur « ressources » et à l'action « track ».

Le mécanisme de capture garde en mémoire l'identifiant de l'utilisateur et la ressource utilisée, à savoir le nom de l'épistème ouvert.

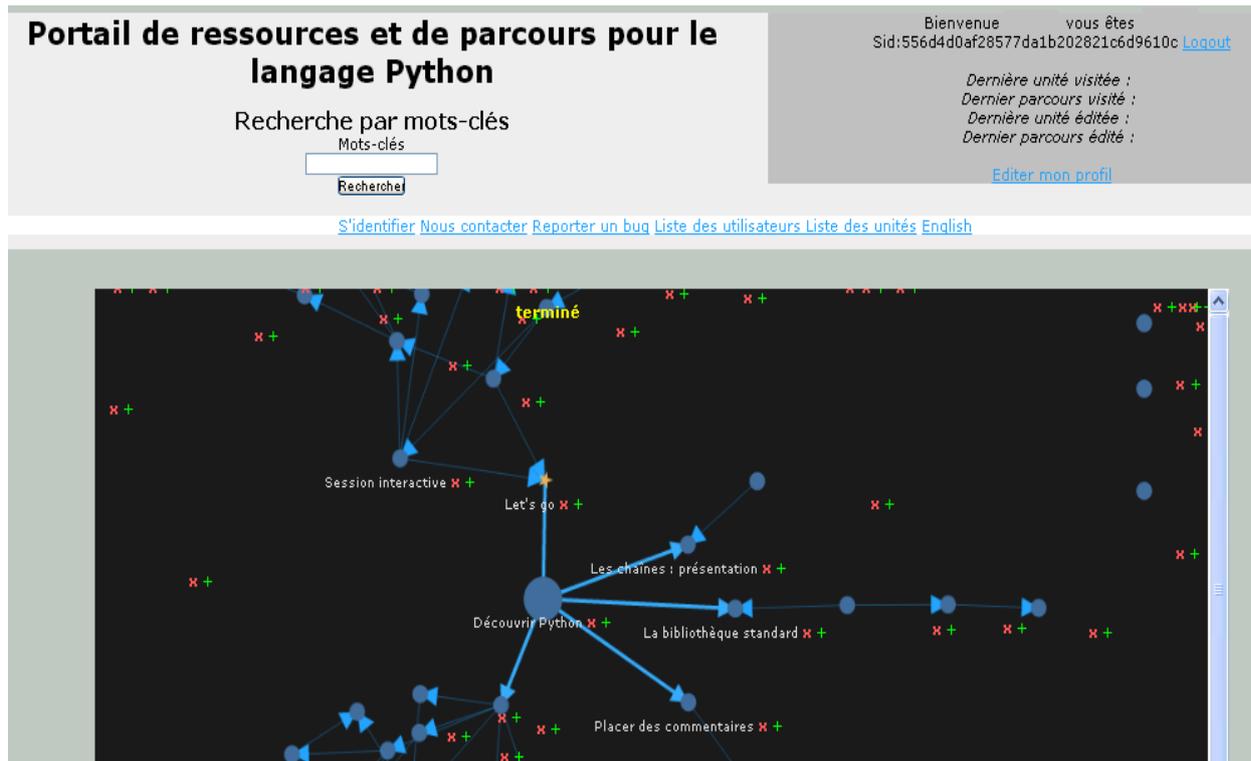


Figure 55 : une vue du graphe des épistèmes sur le prototype PRECEPTOR

a. Dérivation d'une copie/Structure

La capture d'écran (Figure 56) montre l'intégration et l'utilisation de JSONEditor⁹⁶ permettant à l'intégrateur ou au concepteur (éventuellement enseignant-concepteur) de visualiser le modèle de données du graphe des unités épistémiques (ou sous-graphes) en JSON. L'avantage de cet éditeur est de procurer la possibilité d'enrichir le modèle JSON en ajoutant des nœuds et des propriétés (sur le contenu et la présentation) graphiquement.

⁹⁶ http://www.thomasfrank.se/json_editor.html

The screenshot shows a web portal titled "Portail de ressources et de parcours pour le langage Python". It includes a search bar with the text "Recherche par mots-clés" and a "Rechercher" button. A user profile section in the top right corner displays the user's name "admin", a session ID, and a "Logout" link. Below the search bar, there are links for "S'identifier", "Nous contacter", "Rapporter un bug", "Liste des utilisateurs", "Liste des unités", and "English". A central section titled "json" contains a "Load an example:" dropdown menu set to "The source code of this JSON editor". Below this, there is a "Label:" field with the value "json" and a "Value:" field containing a JSON object. The JSON object is a list of nodes with properties like "nodeTo", "nodeFrom", "data", "color", "stype", "id", and "name". At the bottom of the JSON editor, there are buttons for "Data type:" (set to "object"), "Save", "Add child", "Add sibling", "Delete", "Rename", "Cut", "Copy", and "Paste". There are also two checkboxes: "Add children first/siblings before" and "Paste as child on objects & arrays".

Figure 56 : édition du modèle JSON

b. Evaluation des unités et des parcours

En ce qui concerne l'évaluation (Figure 57), l'utilisateur (tuteur ou enseignant) a la possibilité de donner une appréciation (de 0 à 5) sur les différentes unités (épistèmes) selon quatre critères : l'originalité, le côté ludique, l'intérêt pédagogique et la présentation. Il est également possible de voir la liste des évaluations passées. Ceci aura pour but de donner des indications éventuelles quant à la révision de certains contenus (épistèmes) de structures de dépendances ou encore de certains aspects de présentation (voir section II.4 du chapitre 3 de cette partie II).

Recherche par mots-clés
Mots-clés
Rechercher

Dernier parcours visité :
Dernière unité éditée :
Dernier parcours édité :
Editer mon profil

S'identifier Nous contacter Reporter un bug Liste des utilisateurs Liste des unités English

Evaluation des unités

[Lister les évaluations passées](#)

Evaluateur : admin
Auteur de l'unité : Pascal ortiz

Nom de l'unité
Afficher

Commentaires:

Originalité
0

Côté ludique
0

Intérêt pédagogique
0

présentation
0

Evaluer

Accueil Plan du site Ressources Editer une ressource Evaluer une ressource Parcours Evaluer un parcours Aide Statistiques

Powered by ZENDE FRAMEWORK

Figure 57 : interface d'évaluation d'une unité

c. Edition de métadonnées

Dans la fonctionnalité d'édition de métadonnées sur les unités (Figure 58), il peut être utile de mettre à jour le titre, le résumé du contenu, l'url de l'unité (si nécessaire) ou encore l'auteur (ou les co-auteurs s'il y en a). L'utilisateur, s'il le souhaite, peut également demander à exporter une archive compatible SCORM (peut être exécuté sur tout type de LMS), pour l'instant son contenu est figé, mais il est souhaitable que l'utilisateur puisse sélectionner les contenus qu'il souhaite exporter. L'accès à cette fonctionnalité se fait via l'url suivante <http://preceptor-python.net/preceptor/public/ressources/archive>.

Portail de ressources et de parcours pour le langage Python

Recherche par mots-clés
Mots-clés
Rechercher

155Bienvenue a, vous êtes enseignant.
Sid:d4a3a5859d42c4bcfabef94d75620bf4 [Logout](#)

Dernière unité visitée :
Dernier parcours visité :
Dernière unité éditée :
Dernier parcours édité :
[Editer mon profil](#)

[S'identifier](#) [Nous contacter](#) [Reporter un bug](#) [Liste des utilisateurs](#) [Liste des unités](#) [English](#)

graphid:
node19
Name:
Code Python en action
Author:
Pascal Ortiz
Summary:
Initiation au code Python

Url:
./units/code_python_en
Update

[Accueil](#) [Plan du site](#) [Ressources](#) [Editer une ressource](#) [Evaluer une ressource](#) [Parcours](#) [Evaluer un parcours](#) [Aide](#) [Statistiques](#)

Figure 58 : édition de métadonnées sur les unités

V. Prototype HiPPY

L'interface de la plateforme France-IOI peut paraître par moments linéaire et prescriptive. Comment alors réduire ces aspects linéaires ? Comment aider à la navigation dans un graphe d'épistèmes ? L'idée est de recourir à l'adaptation et la construction d'un « sous » graphe spécifique aux exercices France-IOI (dans le parcours lycée).

1. Modèles de données et éléments d'implémentation

Le prototype HiPPY (an epistemic HyPermedia to learn PYthon) dont une version préliminaire a été présentée dans (Reffay, Miled, Ortiz, & Février, 2013) est le fruit de l'intégration et de l'adaptation du graphe des épistèmes au contexte de la plateforme France-IOI. Hormis le graphe des épistèmes, ce prototype se base sur une navigation dynamique et une évaluation épistémique individualisée. Le changement sur l'interface s'est matérialisé par l'ajout d'un nouvel onglet « Notions associées » en ce qui concerne le sous-graphe orienté-tâche. Une section sur les graphes non-orientés tâche a été ajoutée pour chaque série d'exercices (chapitre au sens de la plateforme France-IOI). Il s'agit d'un module en PHP orienté objet intégrant la bibliothèque JIT et utilisant une approche MVC (modèle-vue-contrôleur) spécifique à la plateforme France-IOI. Le type de visualisation RGraph (cf. section II de ce chapitre) a été sélectionné pour la navigation dans le graphe des épistèmes intégré à ce module.

Voici les différentes tables MySQL liées entre elles constituant le modèle conceptuel de la base (Figure 59) :

- Submissions : identifie chaque soumission de code par rapport à un chapitre, une tâche et à un utilisateur. Un iScore à 100 indique que la soumission est réussie.
- Epistemes_relation : définit les relations entre les épistèmes (directe ou distante) et l'identifiant source et destination.
- Epistemes : fait référence au contenu de l'unité épistémique comme rappel de notions ou référentiel ponctuel.
- Epistemes_tasks : lien entre l'épistème associé à chaque tâche (une tâche peut nécessiter plusieurs épistèmes).
- Users : contient en plus le nombre de tâches résolues. sLangProg représente le langage par défaut. Ici nous considérons que sLangProg = 'Python'.
- Groups_members : indique si l'utilisateur fait partie d'un groupe formel (classe ISN) ou informel.
- Tasks : désigne la tâche ou l'exercice. Parmi les principaux attributs nous avons l'identifiant de tâche, l'énoncé (sTextId), le titre, la solution (en termes de code source attendu).

D'autres tables comme logs_user_epi et logs_user_task servent essentiellement à la capture d'événements selon les trajectoires décrites précédemment.

- Logs_user_task : enregistre les événements autour des différents onglets utilisés par rapport à une tâche ou un exercice. Cela concerne par conséquent la micro-trajectoire.
- Logs_user_epi : se charge de mémoriser les différentes actions internes au graphe épistémique. Il s'agit de la trajectoire épistémique.

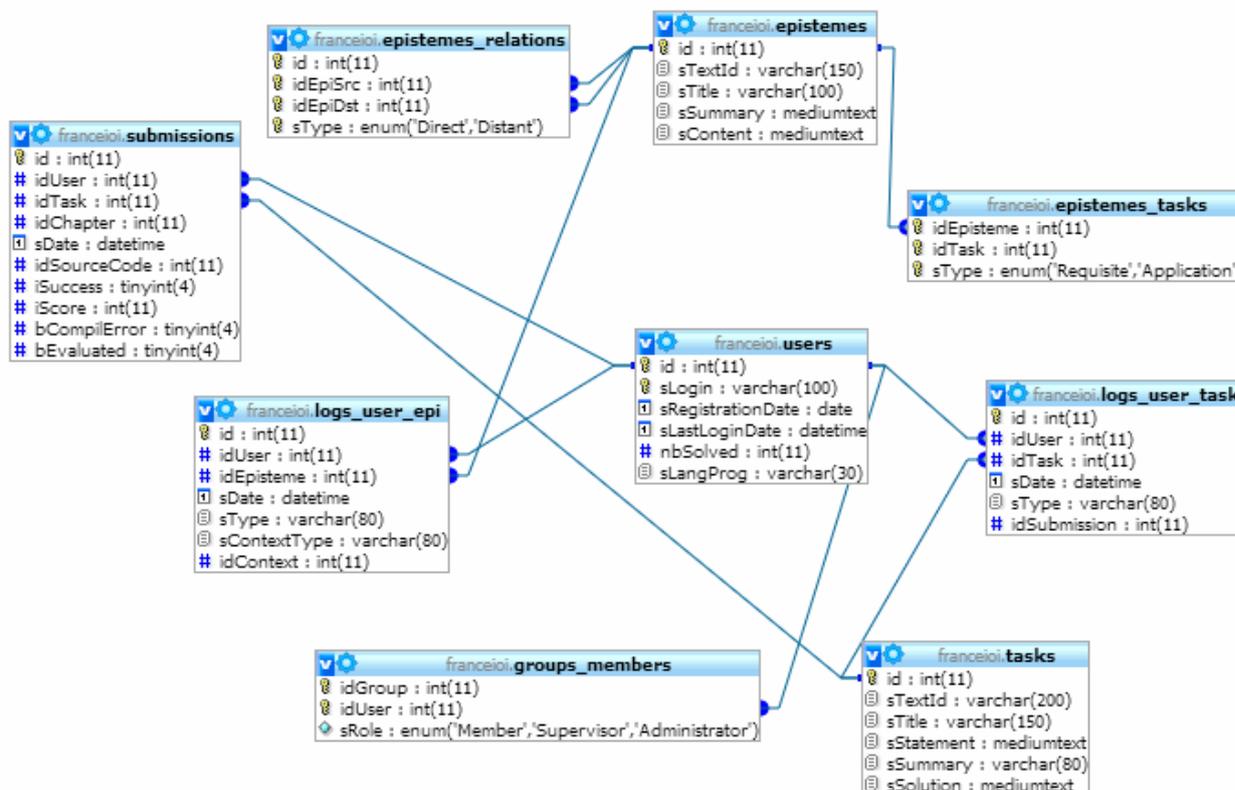


Figure 59 : modèle conceptuel de données de HiPPY

Intéressons-nous à présent aux aspects de navigations dans HiPPY.

2. Navigation dans l'hypermédia épistémique

Le processus de navigation dans les épistèmes peut se décliner sous deux formes :

- Une forme **non-orientée tâche** (toutes les notions utiles) : l'élève accède au graphe des épistèmes d'un chapitre (série d'exercices dans un objectif spécifique) ou de tout le cours par un lien prévu à cet effet. Il peut alors naviguer dans ce graphe (Figure 60). Les différents états des épistèmes illustrent en partie la modélisation introduite en chapitre 3 (Partie II).

Notions utiles du chapitre (expérimental)

Vous trouverez ci-dessous un graphe interactif (expérimental) des notions utilisées dans ce chapitre.

Vous pouvez **cliquer** sur une notion pour centrer la vue sur celle-ci, ajuster la vue en déplaçant le graphe ou en le faisant **glisser** (avec la souris) et **double-cliquer** sur la notion centrale pour en afficher la présentation détaillée. Vous pouvez également si nécessaire **agrandir/réduire** la vue en utilisant la molette de la souris.

Une flèche (pré-requis) de A vers B indique que la notion A est nécessaire pour comprendre la notion B.

NB: L'affichage du graphe est optimum avec Google Chrome et Internet Explorer (version 9). Certaines flèches sont inversées sur Firefox (version 19).

N'hésitez pas à [nous signaler](#) les problèmes éventuels.

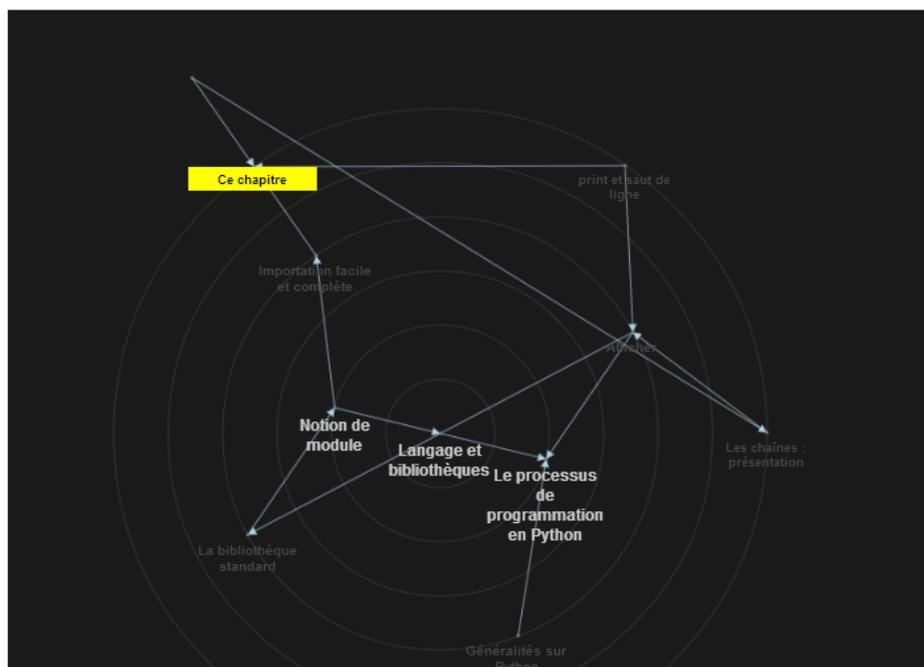


Figure 60 : exemple de graphe non-orienté tâche

- Une forme **orientée tâche** (notions associées) : tout comme pour le graphe complet, la navigation se fait d'une manière similaire (sélection et ouverture) mais le graphe présenté est réduit et ne contient que les épistèmes utiles à la résolution de l'exercice en cours (cf. Figure 61). Les nœuds en blanc (sans aucun symbole) représentent les épistèmes *non encore validés*, et ceux accompagnés d'un symbole vert ceux qui sont *validés* (mais pas nécessairement visités). Un simple clic permet de dégager la vue et de ne faire apparaître que les dépendances relatives à l'épistème sélectionné. Un double clic affiche le contenu de l'épistème. L'élève peut recourir également aux épistèmes ancêtres (prérequis) directement à partir du contenu ouvert.

problèmes >> Affichage de texte, suite d'instructions >> Labyrinthe

← Python ✓ Labyrinthe Validation →

Sujet Résoudre Conseils **Notions associées** Activité Correction

Vous trouverez ci-dessous un graphe interactif (expérimental) des notions travaillées dans cet exercice.

Vous pouvez **cliquer** sur une notion pour centrer la vue sur celle-ci, ajuster la vue en déplaçant le graphe ou en le faisant **glisser** (avec la souris) et **double-cliquer** sur la notion centrale pour en afficher la présentation détaillée. Vous pouvez également si nécessaire **agrandir/réduire** la vue en utilisant la molette de la souris.

Une flèche (pré-requis) de A vers B indique que la notion A est nécessaire pour comprendre la notion B.

NB: L'affichage du graphe est optimum avec Google Chrome et Internet Explorer (version 9). Certaines flèches sont inversées sur Firefox (version 19).

N'hésitez pas à nous **signaler** les problèmes éventuels.

```
graph TD; A[Cet exercice] --> B[Importation facile et complète]; B --> C[Langage et bibliothèques]; C --> D[Le processus de programmation en Python]; C --> E[Notion de module]; D --> F[Afficher]; D --> G[Généralités sur Python]; E --> H[La bibliothèque standard];
```

Figure 61 : sous-graphe de la tâche "Labyrinthe"

3. Mise en place de l'évaluation épistémique individualisée

Pour assurer une navigation à la fois dynamique et s'appuyant sur une évaluation épistémique individualisée, nous pouvons nous demander alors :

- comment peut-on adapter la navigation ?
- comment mettre à jour le profil ?

a. Modèle d'un épistème et ses primitives associées

Un épistème peut avoir une relation directe ou indirecte avec un autre épistème, être un épistème requis ou d'application dans une tâche. Les primitives associées (dont la plupart interrogent directement la base) à un épistème permettent notamment:

- d'importer le contenu d'un épistème ;
- d'importer les données qui lui sont associées (en termes de dépendance avec les autres épistèmes ou les autres tâches) ;
- de retourner tous les épistèmes prérequis ;

- d'obtenir tous les épistèmes nécessaires à la résolution d'une tâche (ceci constitue ainsi le sous-graphe orienté-tâche) et tous les épistèmes utiles à un chapitre (sous-graphe minimum couvrant l'ensemble des concepts utiles à la liste des tâches d'un chapitre) ;
- de générer le modèle individualisé de graphe au format JSON (détaillé ci-après) ;
- de construire la fermeture transitive du graphe (détaillé ci-après).

D'autres fonctions en lien avec la mise à jour et l'édition du graphe concernent :

- l'ajout de dépendance entre deux épistèmes ou bien entre un épistème et une tâche ;
- la suppression d'une dépendance entre deux épistèmes ou entre un épistème et une tâche.

b. Mécanisme de traçage

Le mécanisme de traçage concerne principalement les différentes actions liées au graphe épistémique (trajectoire épistémique) ainsi que les différents onglets utilisés (tracés dans la micro-trajectoire). Des appels Ajax récupèrent les données du côté client et sont transmises au serveur pour être ensuite stockées dans l'une des tables logs_users_epi ou logs_user_task.

Voici un exemple d'appel (Figure 62) qui permet de transmettre l'identifiant de l'épistème courant en train d'être ouvert (état epiSeen).

```
$.ajax({
  type: "GET",
  url: url + "&idEpisteme=<?echo $id;?>&sAction=epiSeen",
  datatype: "String",
  success: function(msg) {
  }
});
```

Figure 62 : envoi de l'identifiant de l'épistème ouvert

Ci-dessous (Figure 63) un exemple de sauvegarde avec une action en rapport avec la trajectoire épistémique (sAction contient le préfixe epi) dans la table MySQL logs_users_epi de l'identifiant de l'épistème, de l'identifiant de l'exercice (idContext), du sContext (s'il s'agit d'un graphe par chapitre ou global) après réception ainsi que l'identifiant de l'utilisateur (idUser), la date courante et le type d'onglet (ici TabEpistemes).

```
if (preg_match('/^epi/', $sAction))
{
  $idEpisteme = $_GET->get("idEpisteme");
  $idContext = $_GET->get("idContext");
  $sContext = $_GET->get("sContext");
  if ($sContext == "Global")
  {
    $idContext = -1;
  }
  $query = "
  INSERT INTO logs_user_epi(idUser, idEpisteme, sDate, sType, sContextType, idContext)
  VALUES($idUser, $idEpisteme, UTC_TIMESTAMP(), '$sAction', '$sContext', $idContext)";
  sql::query($query);
}
```

Figure 63 : exemple d'enregistrement de traces

c. Individualisation de l'évaluation épistémique

La fonction permet d'interroger les épistèmes validés (censés être maîtrisés directement ou indirectement) et de répercuter les différentes situations sur la présentation et de mettre à jour la visualisation du graphe courant des épistèmes. À titre illustratif (Figure 64), les épistèmes « Notion de module » et « Afficher » sont validés. Un épistème validé aura comme étiquette un symbole "✓" vert.



Figure 64 : adaptation de la navigation et évaluation épistémique

Voici ci-dessous le principe général (voir la fonction `getJitDataFromEpis` en annexe VI.1.a) :

Algorithme 1 : génération du modèle individualisée du graphe

Entrées (nœud racine, dépendances)

Si (le nœud racine est vide) Alors

il ne s'agit pas d'un épistème (il s'agit alors d'un exercice ou ensemble d'exercices)

Pour chaque nœud \in aux dépendances Faire

Si (Epistème est Validé) alors

EtatValidé \leftarrow vrai

Générer modèle JSON : initialiser les attributs identifiant,

nom et données de présentation : validation ou non)

Pour chaque successeur (est un prérequis de) du nœud courant Faire

Générer modèle JSON : initialiser les adjacences

d. Construction de la fermeture transitive

La construction de la fermeture transitive s'appuie principalement sur l'algorithme de Warshall (Warshall, 1962). L'idée derrière cela est de savoir s'il existe un chemin de x à y pour tout couple de sommets (x,y) du graphe. S'il existe un chemin de x à y ET s'il existe un chemin de y à z Alors il existe un chemin de x à z . En utilisant une matrice M d'adjacence ou matrice booléenne (1 pour dire s'il y a lien, 0 sinon), il est possible de générer une matrice M' renseignant sur tous les chemins possibles.

Algorithme 2 : calcul de la fermeture transitive (Warshall)

```

M' ← M
Pour i allant de 1 à N faire
    M'[i][i] ← 1
Fin Pour
Pour k allant de 1 à N faire
    Pour i allant de 1 à N faire
        Pour j allant de 1 à N faire
            M'[i][j] ← M'[i][j] OU (M'[i][k] ET M'[k][j])
        Fin Pour
    Fin Pour
Fin Pour

```

Ci-dessous notre adaptation de l'algorithme de Warshall pour ajouter les dépendances distantes.

Algorithme 3 : construction adaptée de la fermeture transitive du graphe

```

Supprimer de la table des relations entre les épistèmes tous les liens distants
Req ← Sélectionner les couples {épistème source, épistème destination}
ayant des liens directs
Pour chaque élément résultat de la requête Req Faire
    Stocker dans un tableau associatif aDirect
iMax ← identifiant maximum à partir de la table des épistèmes
Pour i allant de 1 à iMax Faire
    Si ( ∃ i ∈ aDirect) Alors
        Ajouter aDirect [i] comme tableau
aDistant ← aDirect
Pour n1 allant de 1 à iMax Faire
    Pour n2 allant de 1 à iMax Faire
        Pour n3 allant de 1 à iMax Faire
            Si (n2 ∈ aDistant[n1] ET n3 ∈ aDistant[n2]
            ET (n1 ∉ aDistant[n3])) Alors
                aDistant[n1][n3] ← n3
        Fin Pour
    Fin Pour
    Pour chaque élément du tableau aDistant comme src associé à dst Faire
        Si (l'élément dst ∉ aDirect[src]) Alors
            Ajouter relation distante entre src et dst

```

Ceci nous a été utile pour calculer les liens indirects (distantes) entre deux épistèmes donnés. Par conséquent, tous les liens distants possibles sont attachés aux différents épistèmes sauvegardés. Pour plus de détails, se reporter à la fonction setClosureDAG en annexe VI.1.b. Face aux trois prototypes construits, nous pouvons faire les constats suivants :

- Le premier prototype était une séquence plutôt simple d'épistèmes qui permettait une navigation sans pour autant s'adapter à l'utilisateur ni proposer des parcours par objectifs.
- Le second prototype PRECEPTOR exploite une navigation interactive se basant sur la visualisation des graphes en donnant la possibilité de tester du code grâce à un interprète externe.
- Le troisième prototype HiPPY s'intègre comme hypermédia dans la plateforme France-IOI. Il permet d'être utilisé en conjonction avec les exercices proposés. Outre les mécanismes de navigation dynamique, il offre un bilan épistémique individualisé.

VI. Conclusion

Nous avons montré à travers ce chapitre que l'aspect « visualisation » reste primordial pour aider à la navigation d'une manière dynamique et individualisée.

Nous avons passé en revue trois prototypes qui ont permis de mettre en pratique, certes à des stades assez variables, les principes énoncés précédemment.

Voici un tableau synthétique résumant les différentes couches supportées par les trois prototypes construits :

Prototype/ Environnement	Couche Ressources	Couche Parcours	Couche navigation	Couche Traçage	Couche Fouille de donnés et adaptation	Couche Réingénierie
Séquence sur CloudScorm	Oui (chaque épistème est une micro- ressource)	Non (pas de parcours prescrit), même si la séquence peut être assimilée à un parcours pour résoudre un exercice)	Mécanismes de navigation	À l'aide des SCO de SCORM, différents états et temps sont fournis	Non	Non
PRECEPTOR	Oui (chaque nœud du graphe est un épistème et donc une ressource)	Oui pour certaines sections	Navigation dynamique à l'aide de JIT	Oui, éléments consultés par les apprenants et traces d'édition	Non	Oui de manière limitée (édition de métadonnées, édition de modèle de graphes)
HiPPY	Oui	Oui (jumelée avec les exercices à résoudre)	Oui (JIT)	Oui	Oui	Oui, seulement sur certains aspects de la mise à jour du graphe

Tableau 5: couches supportées par les prototypes

Une fois construits et mis en place, il nous paraît vital de procéder à la mise à l'épreuve de ces prototypes pour :

- d'une part, tester et valider les composantes et ensuite valider les principes que nous avons énoncés
- et de l'autre étudier et analyser les différents usages possibles. Quels sont les outils et les opérations nécessaires pour y parvenir ?

La prochaine étape sera de poser les processus d'exploitation de traces et les mécanismes sous-jacents. C'est l'un des objectifs du chapitre qui suit.

Chapitre 5 : Vers des processus unifiés d'analyse de données en éducation

I.	INTRODUCTION	131
II.	SELECTION, MODIFICATION ET INTERROGATION DE DONNEES SUR DES REQUETES SIMPLES ET CROISEES 131	
1.	<i>Requêtes sur les macro-trajectoires</i>	131
2.	<i>Requêtes sur les micro-trajectoires</i>	132
3.	<i>Requêtes sur les trajectoires épistémiques</i>	133
III.	VISUALISATION ET EXPLOITATION DE DONNEES PAR DES OUTILS DE FOUILLE DE DONNEES	133
1.	<i>Importation de données issues de tables</i>	134
2.	<i>Adaptation des données</i>	134
3.	<i>Application des algorithmes de fouille</i>	135
IV.	CROISEMENT DE DONNEES ISSUES DE TRACES ET D'UN QUESTIONNAIRE : VERS LA MISE EN RELATION DES DONNEES DECLARATIVES ET OBJECTIVES	137
V.	CONCLUSION	137

I. Introduction

Pour pouvoir observer et ensuite analyser le comportement des utilisateurs et leurs interactions avec un système, il est utile de recourir à des mécanismes d'analyse de traces. La fouille de données à partir de traces, qu'il s'agisse d'une simple requête, de plusieurs requêtes croisées, voire même d'algorithmes de fouille plus élaborés (voir Partie I - Chapitre 5) est un élément majeur. Nous nous intéressons à présent au processus d'analyse des données d'usage issues du prototype HiPPY présenté dans le chapitre précédent. La première partie concerne les opérations de sélection, de modification et d'interrogation de données. La seconde montre l'exploitation d'outils de fouille de données (présentés dans le chapitre 5 de la partie I) et de processus utiles qui donneront lieu aux résultats d'analyse (issus de ces mêmes outils) de la partie III. La troisième partie pose les interrogations sur la méthodologie envisagée et sur la possibilité de croiser des données objectives issues de traces et des données déclaratives issues de questionnaires ou entretiens.

II. Sélection, modification et interrogation de données sur des requêtes simples et croisées

L'exploitation des données issues de traces peut faire appel à des moyens différents sur l'interrogation de la base. Ces données doivent être forcément en lien avec les indicateurs que nous chercherons à évaluer, et avec les traces modélisées sous forme de trajectoires multi-niveaux que nous avons introduites. Nous donnons quelques requêtes permettant de faire émerger un certain nombre des observables utiles à l'élaboration de ces indicateurs.

1. Requêtes sur les macro-trajectoires

Ces données sont certes déjà disponibles avant intégration du prototype HiPPY. Il est également intéressant de les observer une fois le prototype HiPPY intégré dans la plateforme France-IOI. Deux comportements nous semblent essentiels à ce niveau : l'inversion et le saut d'exercices. L'inversion désigne le fait qu'un apprenant fasse une liste d'exercices dans un ordre inversé. Le saut indique simplement que l'utilisateur évite certains exercices. Voici quelques requêtes permettant notamment d'interroger le nombre d'utilisateurs enregistrés entre le 1^{er} septembre 2012 et le 1^{er} septembre 2013 ayant résolu au moins un nombre minimum d'exercices (variable \$limit) appartenant à des groupes ISN et ayant choisi Python comme langage de programmation (voir Figure 65).

```
SELECT COUNT(*) AS count
FROM users
JOIN groups_members
ON groups_members.idUser = users.id
WHERE
  sLangProg = 'Python' AND
  sRegistrationDate BETWEEN '2012-09-01' AND '2013-09-01' AND
  nbSolved >= $limit";
```

Figure 65 : requête concernant les macro-trajectoires

Le calcul des exercices non faits peut s'effectuer par le principe suivant :

Algorithme 4 : calcul des exercices sautés

Pour chaque élément exercice ayant comme association (i, exercice) Faire
Si (user.aTasks[exercice.id] <> -1) Alors
 Min ← minimum(*Min*, *i+1*)
 Max ← maximum(*Max*, *i+1*)
 aAll[j] ← user.aTasks[exercice.id]
 j ← *j+1*
NbExercicesSautés ← *Max+1- Min - user.nbExercices*

Lorsqu'un identifiant d'exercice est à -1 c'est qu'il ne fait pas partie des exercices commencés (ou ouverts) par l'utilisateur.

Pour ce qui est des exercices inversés, la méthode suivante est utilisée :

Algorithme 5 : calcul des exercices inversés

aAll : tableau, *q* : tableau de longueur *aAll*
Pour k allant de 0 à longueur(aAll) Faire
 Max ← 0
 Pour j allant de 0 à k Faire
 Si (aAll[j] < aAll[k]) Alors
 Max ← maximum(*Max*, *q[j]*)
 q[k] ← *Max+1*
 Max ← 0
 Pour i allant de 0 à longueur(q) Faire
 Max ← maximum(*Max*, *q[i]*)
NbExercicesInversés ← user.nbExercices - *Max*

Il est ainsi possible de générer les macro-trajectoires précises en prenant en compte ces deux dimensions. Nous verrons dans la partie 3, l'importance de ces deux indicateurs (sauts et inversions) sur l'usage ou non du graphe épistémique.

2. Requêtes sur les micro-trajectoires

L'utilisation et l'exportation directe de la table logs_user_task permettent de renseigner sur les différents onglets utilisés lors de la phase de résolution de différents exercices. De ce fait, l'identifiant de l'utilisateur, l'identifiant de la tâche, la date et le type d'onglet avisent sur les temps de séjour sur chaque onglet selon la tâche suivie. Il est par ailleurs faisable d'intégrer d'autres attributs (Figure 66) comme l'identifiant, le résultat de la soumission correct (iScore=100) et l'onglet TabEpistemes (le type d'événement associé).

```
select s.id AS IdSub, s.idUser AS idUser, s.idTask, lut.sType
from epistemes e, chapters c, tasks t, users u, logs_user_epi lue,
logs_user_task lut, submissions s
where e.id = lue.idEpisteme and u.id = lut.idUser and
t.id = lut.idTask and u.id = lue.idUser and c.id = s.idChapter
and s.idTask = t.id and s.idUser = u.id and iScore = 100
```

Figure 66 : requête croisée sur les micro-trajectoires

3. Requêtes sur les trajectoires épistémiques

Avant de trouver des données sur les différents événements liés aux épistèmes (déjà décrits précédemment) à partir de la table logs_user_epi, il est peut être intéressant de regarder le nombre d'épistèmes utilisés par tous les utilisateurs ainsi que ceux qui ne sont jamais employés. À cet effet, la requête suivante (Figure 67) permet d'énumérer ceux qui ne sont pas utilisés (par identifiant et intitulé). Si l'on considère le nombre total de soixante-neuf épistèmes, on trouve un seuil inférieur à dix (ici par exemple sept n'ont pas été utilisés).

```
SELECT id, sTextId
FROM `epistemes`
WHERE id NOT
IN (SELECT idEpisteme
FROM logs_user_epi
WHERE sDate > '2013-04-04 18:30')
```

Figure 67 : requête donnant les épistèmes non utilisés

La requête ci-dessous (Figure 68) donne un aperçu quantitatif sur les différents événements rencontrés (epiGraphClick, epiClick, epiGraphDoubleClick et epiSeen) lors de la manipulation du graphe épistémique.

```
SELECT count(*) as nombre, sType
FROM `logs_user_epi`
where sDate > '2013-04-04'
group by sType
```

Figure 68 : aperçu quantitatif sur les événements de la trajectoire épistémique

III. Visualisation et exploitation de données par des outils de fouille de données

L'interrogation d'une base de données avec ses différentes tables ne paraît pas suffisante. L'utilisation d'outils spécifiques peut apporter des solutions, surtout s'il s'agit de données parfois hétérogènes. Nous décrirons dans cette section l'utilisation de trois outils : Weka, RapidMiner, et dans une moindre mesure l'outil Knime.

Comment alors unifier les processus d'analyse en fouille de données éducatives ? Certains penchent pour l'utilisation de métamodèles ou de processus de fouilles pour capturer des observables pertinents. Nous avons opté pour des processus d'analyse de données (ou workflows) et essayer de généraliser ces étapes quel que soit l'outil utilisé.

Sur les trois outils utilisés (Weka 3.6.8, Knime, RapidMiner), il y a la possibilité de créer, exécuter, réutiliser et exporter des processus pour pouvoir les réadapter et pour automatiser les tâches d'analyse ou bien pour valider des données. Nous pouvons entrevoir néanmoins dans ces outils quelques composantes communes : l'importation de données, l'adaptation des données et l'application des algorithmes de fouille.

1. Importation de données issues de tables

Une première étape est l'importation des données avec différents formats supportés : csv⁹⁷, arff⁹⁸ (plutôt pour Weka), ceci peut inclure l'étape de préformatage de données avec sélection ou exclusion de certains attributs. L'utilisation de PhpMyAdmin dans (Wamp/Lamp) peut aider à générer une version en csv des différentes tables sélectionnées. Le format csv nous paraît important comme élément générique, puisqu'il est facilement lu par tout outil de fouille de données. À titre d'exemple, l'interface générale de Weka (Figure 69) montre les différentes fonctionnalités : pré-traitements (avec Preprocess), classify pour appliquer des algorithmes de classification, clustering, associate pour les règles d'association par exemple ou encore croiser des visualisations de données avec Visualize.

2. Adaptation des données

Il y a ensuite une étape qui permet de s'adapter aux types de classifieurs (ou de régresseurs) sur les types (binomiaux, polynomiaux ou même entiers). C'est à cette étape qu'il est peut être judicieux de procéder à la sélection de certains attributs, ou à la transformation des types de données selon les algorithmes de classification, de règles d'association voire de motifs récurrents. Sur la figure suivante (Figure 69), l'importation des données se fait à l'aide de Open file et leur transformation à l'aide du filtre *NumericToNominal*.

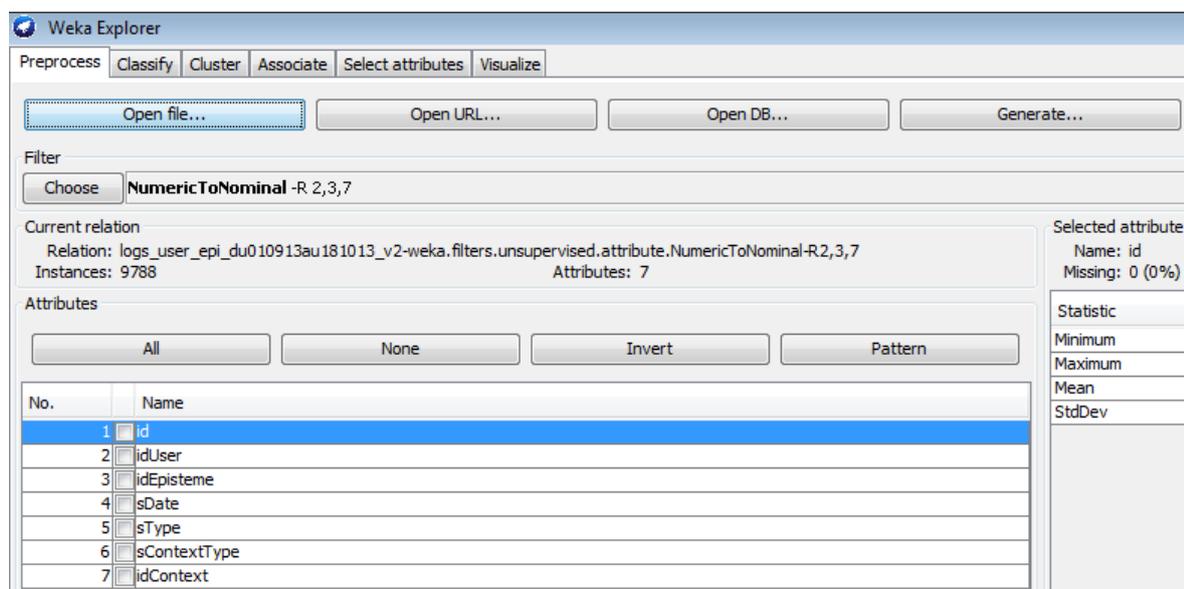


Figure 69 : transformation des attributs idUser, idEpisteme et idContext de numériques en nominaux

⁹⁷ Comma-Separated Value : représentation de données tabulaire sous format de valeurs séparées par un séparateur spécifique (virgule, point-virgule, etc...)

⁹⁸ Attribute-Relation File Format

3. Application des algorithmes de fouille

La plupart des outils déjà cités, propose des familles catégorisées d'algorithmes selon le type de données (déjà étiquetées ou non, apprentissage supervisé ou non, données continues ou discrètes, binomiaux, polynomiaux, numériques, etc...). Ci-contre (Figure 70) une exécution de l'algorithme Kmeans avec n=3 clusters et une distance euclidienne pour tous les attributs.

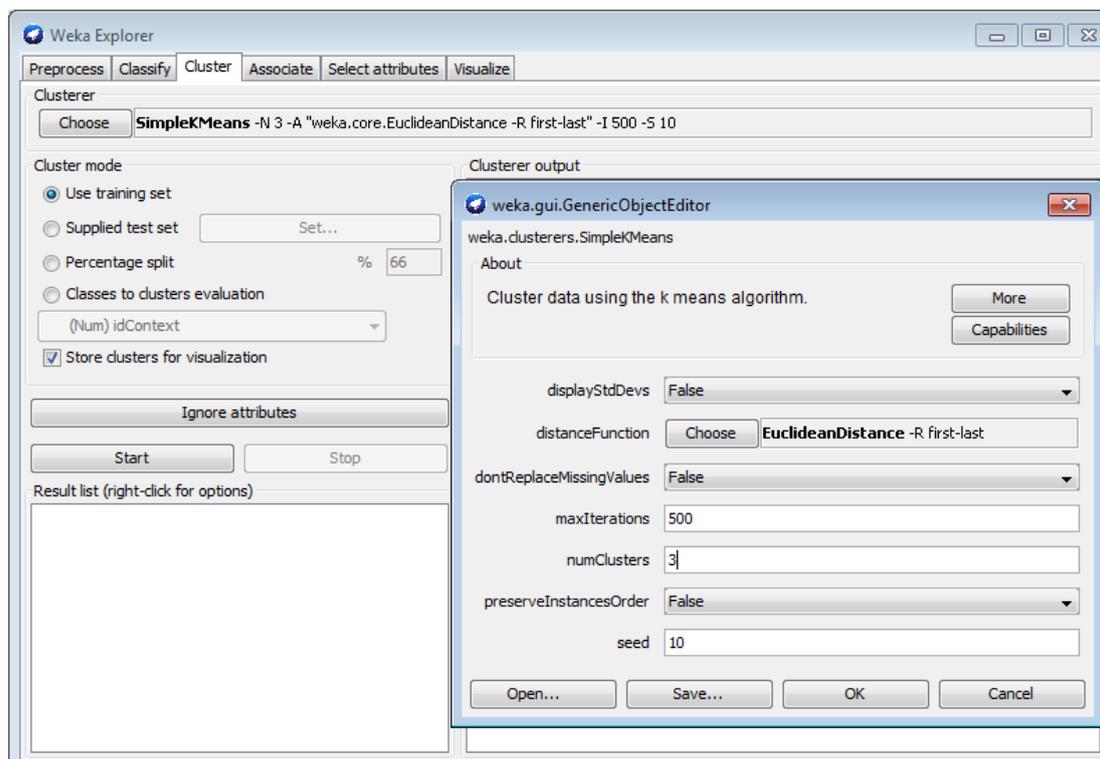


Figure 70 : exécution de Simple Kmeans dans Weka

Voici un exemple (Figure 71) de processus issu de RapidMiner : l'importation de données se fait à l'aide de l'opérateur *Read csv*, l'adaptation des données à l'aide de *select Attributes* et l'algorithme de fouille à travers *W-Hierarchical Clustering* (une version provenant de Weka)

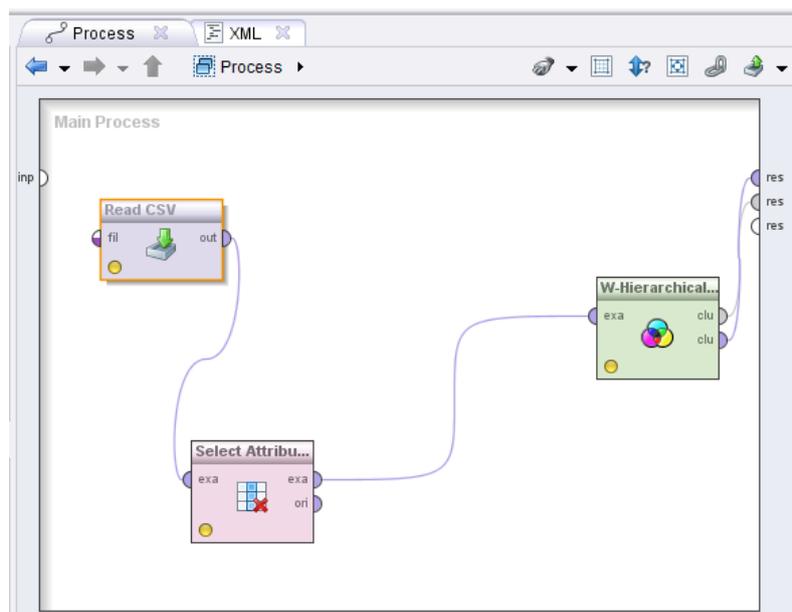


Figure 71 : exemple de processus avec RapidMiner

Pour pouvoir extraire des règles d'associations, nous opterons pour l'algorithme « *Apriori* » de Weka. Voici un paramétrage possible : `weka.associations.Apriori -N 100 -T 0 -C 0.6 -D 0.05 -U 1.0 -M 0.1 -S -1.0 -c -1`, où l'on demande 100 règles (paramètre N) avec une confiance minimale de 0,6 (paramètre C). Le delta (paramètre D) est le facteur de décroissance du support. La métrique choisie est la confiance (-T 0). Le niveau de signification (*significance level* : -S) est à 1.0 et la classe de l'index est à -1 signifie que le dernier attribut est pris comme attribut de classe (-c -1). La valeur minimale du support (lowerBoundMinSupport) est désigné par -M 0.1, et la valeur maximale (upperBoundMinSupport) par -U 1.0. Un pré-traitement « *NumericToNominal* » est nécessaire pour rajouter des attributs de type nominal. L'objet est de transformer des valeurs numériques en valeurs non-numériques (nominales). Cela peut être vu comme une manière sommaire de discrétisation, c'est-à-dire convertir des valeurs continues en valeurs discrètes. Sur Knime, il est également possible d'ajouter des opérateurs de présentation des résultats (comme sur la Figure 72) si cela est nécessaire par exemple à l'aide de l'opérateur *Scorer*. Sur cet exemple la génération d'un arbre de décision (avec ses composantes Learner et Predictor pour la prédiction).

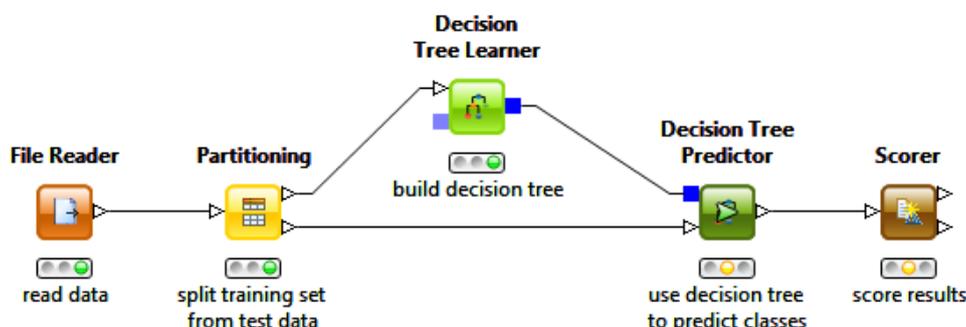


Figure 72 : exemple de processus avec Knime

L'arbre de décision résultant du processus décrit sur la Figure 72 sera partiellement présenté dans la partie III - chapitre 3 - section III.2.a.

IV. Croisement de données issues de traces et d'un questionnaire : vers la mise en relation des données déclaratives et objectives

L'objet sera d'exploiter les données sur les traces et les données déclarées par les utilisateurs. La partie III montrera plus en détail cette correspondance entre le questionnaire et l'analyse des trajectoires, pour comprendre l'utilité et la satisfaction générale.

La mise en place d'un questionnaire en ligne (voir annexe VIII) a généré une profusion de résultats (131 réponses en décembre 2013 ; 266 jusqu'au 6 octobre 2014). Les différentes réponses à ce questionnaire ont été récupérées et transformées pour être mises en correspondance avec des données de traces. Pour cela, nous avons transformé l'ensemble des items dans un format CSV.

Mais comment faire alors pour lier ces deux types d'information ? Tout en garantissant une anonymisation de chaque utilisateur, nous avons pensé intégrer l'identifiant dans le questionnaire qui sera décrit plus loin. Grâce à cet identifiant, il nous a été possible d'avoir des tables recrées qui contiennent par exemple un sous-ensemble des réponses ainsi que des indicateurs sur l'usage et les raisons ou les choix de certains items. À cet effet, une table « questionnaire » a été ajoutée à notre base de données locale qui comprend entre autres les tables présentées dans le chapitre précédent sur le prototype HiPPY.

Sur la Figure 73, la requête permet d'afficher un taux d'usage des épistèmes, un taux d'usage total des onglets ainsi qu'un champ spécifique au questionnaire sur l'utilité (q.P).

```
SELECT u.id, q.B, q.P, count( lut.sType ) AS EpiUsage, count( lutt.sType ) AS TabTotal
FROM `questionnaire` q, users u, logs_user_task lut
WHERE q.B = u.sLogin AND lut.idUser = u.id AND lut.sType = "TabEpistemes"
GROUP BY u.id
```

Figure 73: exemple de requête de croisement entre traces et questionnaire

V. Conclusion

Nous avons dévoilé dans ce chapitre différents moyens et outils servant à la fouille et à l'analyse de données concernant l'usage et la mise à l'épreuve du prototype HiPPY. De simples requêtes, en passant par des processus plus élaborés faisant intervenir des algorithmes de fouille fréquemment utilisés en apprentissage artificiel et jusqu'à la mise en correspondance de certaines données issues d'un questionnaire avec des traces, la visée demeure la même et consiste à identifier des indicateurs pertinents pour pouvoir d'une part : évaluer l'usage et l'appropriation de l'outil, et de l'autre, valider certaines hypothèses que nous tenterons d'énoncer plus tard.

Une des perspectives de cet aspect serait de concevoir un outil qui se baserait principalement sur différentes bibliothèques des outils (Weka, RapidMiner ou encore Knime) et qui utiliserait des moyens de transformation vers un format de type RDF. Cet démarche pourrait apporter une certaine genericité en utilisant des processus indépendants de l'outil et permettrait d'interagir avec des données annotées sémantiquement. Plus généralement, quelles sont les pistes de recherche appropriées pour favoriser la complémentarité entre métamodèles et fouille de données ?

Conclusion partie II

Cette partie a été l'occasion pour nous d'exposer nos principales contributions en termes de modélisation, de choix de conception, de prototypage et de méthodes visées pour l'analyse et la fouille de données. Les thématiques abordées ont naturellement porté sur nos majeures préoccupations de recherche : les mécanismes d'individualisation de la navigation et leurs aspects assistants.

Pour cela, nous avons d'abord introduit une approche générale liant les activités d'édition des ressources d'apprentissage aux activités de navigation. Il nous semble utile de suivre cette approche pour, d'une part, viser cette adaptation et personnalisation de la navigation, et de l'autre, assurer une évolution de la ressource d'apprentissage. Nous avons vu que le graphe des épistèmes constituait un socle primordial de la ressource et que le système à base de navigation dynamique JIT retenu par nos soins avait potentiellement de bons atouts pour pouvoir parvenir à ces objectifs. Les prototypes construits (séquence SCORM, PRECEPTOR et HiPPY), ont permis, à différents stades, d'appliquer certains de nos principes et d'éprouver certaines de nos modélisations.

Une des modélisations qui nous a permis de visualiser le processus de traçage et de navigation dans le graphe des épistèmes couplés aux exercices France-IOI est la trajectoire à trois niveaux. Cette trajectoire se compose d'une macro-trajectoire portant sur le niveau exercice, d'une micro-trajectoire (niveau intra-exercice où les différents onglets sont utilisés jusqu'à la 1^{ère} bonne résolution) et d'un niveau épistémique faisant concourir la multitude d'épistèmes.

Nous avons abordé la question fatidique de la collecte, l'adaptation et l'extraction de données utiles à partir de traces. Nous avons pu alors mobiliser divers moyens : requêtes simples ou croisées de bases de données ainsi que des processus de fouilles adaptés à différents outils tels que Weka, RapidMiner ou encore Knime. Même si certains outils identifiés proposent de décrire les processus d'analyse sous format XML, cela n'est pas toujours évident, s'il l'on veut les transposer dans d'autres outils. Cela pourrait amener à réfléchir sur des processus indépendants de l'outil ou de la plateforme d'analyse de traces destinés aux chercheurs ou aux enseignants.

Nous pensons tout de même que la collecte des traces et leur analyse est insuffisante pour évaluer les systèmes élaborés ou expliquer certains traits d'usage, c'est pourquoi nous avons estimé intéressant de croiser des données issues d'un questionnaire avec les traces obtenues.

Tous ces éléments s'avèrent comme préalables à l'expérimentation d'un système donné. D'autres objets de réflexion peuvent porter sur :

- des processus pré-conçus pour faire ressortir des indicateurs pertinents à partir d'un fichier brut csv non nécessairement nettoyé, pour pouvoir catégoriser les données utiles jusqu'à la phase finale de visualisation des indicateurs ;
- l'apport d'une vue d'ensemble pour servir dans un premier temps aux données issues du format spécifique France-IOI mais qui doit être extensible à un autre format (par exemple de données issues d'environnements plus massifs en incluant des données sur les interactions, d'autres types de ressources multimédia, etc..) ;
- et le fait de pouvoir transformer des données sous MySQL en données sémantiques afin d'exécuter plus facilement des inférences et faire ressortir et ainsi réutiliser notre modèle sémantique de trajectoires.

PARTIE III : MISE À L'ÉPREUVE, EXPERIMENTATION ET RESULTATS

Introduction de la partie III :

Nous avons introduit dans la partie précédente différentes contributions par rapport aux principes énoncés. Cette partie se veut :

- un espace d'expérimentation, de validation de certaines hypothèses formulées en lien avec nos principes énoncés ;
- une étape qui permet de mettre à l'épreuve les différentes questions de recherche qui nous ont motivés dans ce travail.

Le 1^{er} chapitre de cette partie III retrace le cheminement expérimental utilisé pour faire un test de performance sur certains types de visualisation de graphe de la bibliothèque JIT. Il a pour but également de donner quelques résultats qui sont des pistes et des éclaircissements qui peuvent être vus comme des recommandations.

Le 2^{ème} chapitre examine le terrain empirique avant d'évaluer les retours d'usage et les données utiles susceptibles d'être récoltées. Ceci inclut les différentes hypothèses à vérifier et les principes d'interaction homme-machine à prendre en considération.

S'agissant du 3^{ème} chapitre, il est destiné à éclairer le lecteur sur les différents résultats obtenus à différents stades d'expérimentation et selon différents modes d'acquisition et d'analyse. Voici l'organisation de la partie 3 (Figure 74).

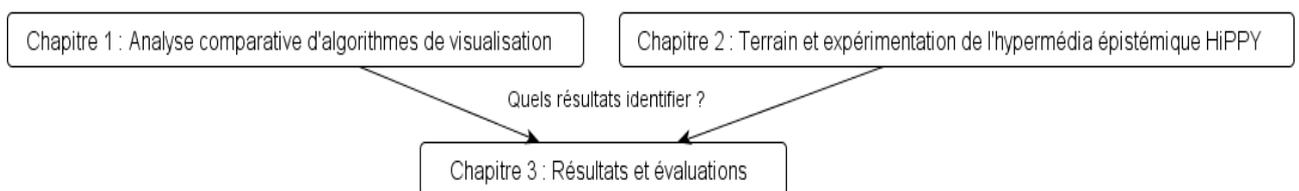


Figure 74 : liens entre les chapitres de la partie III

Chapitre 1 : Analyse comparative d'algorithmes de visualisation

I.	INTRODUCTION	141
II.	PROTOCOLE ET METHODOLOGIE SUIVIS	141
III.	MESURES DES TEMPS D'EXECUTION	142
IV.	COMPARATIFS ET RESULTATS	142
V.	CONCLUSION	145

I. Introduction

Avant d'utiliser telle ou telle technologie, bibliothèque ou tel autre composant logiciel, il est parfois avisé de recourir à une mise à l'épreuve en amont, afin de discerner les potentialités et les limites. Nous avons voulu procéder à cette mise à l'épreuve dans notre démarche avant d'intégrer une navigation dynamique à l'aide de l'outil JIT. Nous présentons dans ce chapitre le protocole suivi et les étapes sous-jacentes, l'étape de mesure des temps ainsi que les principaux résultats dégagés à partir de l'évaluation de trois algorithmes de visualisation.

II. Protocole et méthodologie suivis

Pour comparer les performances des différents outils de visualisation, nous avons suivi les étapes suivantes :

- la construction aléatoire de graphes $G(n, p)$ à l'aide de Gephi⁹⁹ avec n , le nombre de sommets et p , la probabilité de connexion ;
- la production de fichiers de données contenant des listes d'adjacence ;
- la transformation des listes d'adjacence au format JSON ;
- et l'exécution du type d'algorithme et de visualisation selon les données récupérées précédemment.

Le protocole (voir Figure 75) a consisté à générer dix graphes aléatoires et à comparer les exécutions pour les trois algorithmes de visualisation choisis : HyperTree, ForceDirected et RGraph avec :

$n = 10$ et $p = 0,5$;
 $n = \{50, 100\}$ et $p = 0,05$;
 $n = \{224, 500, 1000\}$ et $p = 0,01$;
 $n = \{2000, 3000, 4000, 5000\}$ et $p = 0,001$.

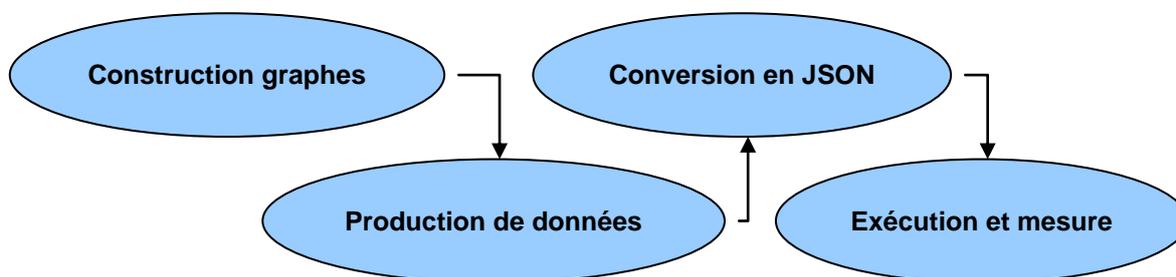


Figure 75 : étapes menant au benchmark

⁹⁹ <http://gephi.org>

III. Mesures des temps d'exécution

Nous avons mesuré les performances d'exécution de trois types de visualisation de la bibliothèque JIT (HyperTree, ForceDirected et RGraph) avec les caractéristiques suivantes : Windows XP SP3, 1Go RAM, 1,66 Ghz, et le navigateur Google Chrome. Cela s'est fait à l'aide de sondes et de transcription des données temporelles : les parties du code à tester ont été encadrées par des lectures de l'horloge (temps de début et temps de fin) pour déduire la durée totale en millisecondes et identifier le comportement de chacun des algorithmes en fonction de la taille des graphes. Nous avons eu recours (pour la majorité des cas) à une dizaine d'exécutions pour la même instance de test pour pouvoir calculer une moyenne de temps d'exécution. Néanmoins, ces valeurs restent indicatives et il se peut qu'il y ait des valeurs différentes selon la charge du processeur et sa vitesse, le navigateur et la quantité de mémoire disponible.

IV. Comparatifs et résultats

Les résultats comparatifs obtenus (Figure 76) donnent une estimation de la complexité moyenne opérationnelle et permettent surtout de discerner le comportement de chacun des algorithmes pour de plus grands graphes. Il eût été possible de calculer la complexité théorique de chacun de ces algorithmes, mais nous avons préféré les mesures effectives vu l'existence de différentes implémentations (autres que JIT). Par exemple, ForceDirected peut avoir une complexité théorique en $O(n^2)$, en $O(n)$ voire en $O(n \log(n))$. Les différentes estimations nous ont montré que les algorithmes RGraph et HyperTree supportent mieux le passage à l'échelle, tandis que le temps d'exécution de l'algorithme ForceDirected croît plus rapidement pour des valeurs élevées (presque 14 minutes pour $n=3000$, $p=0,001$).

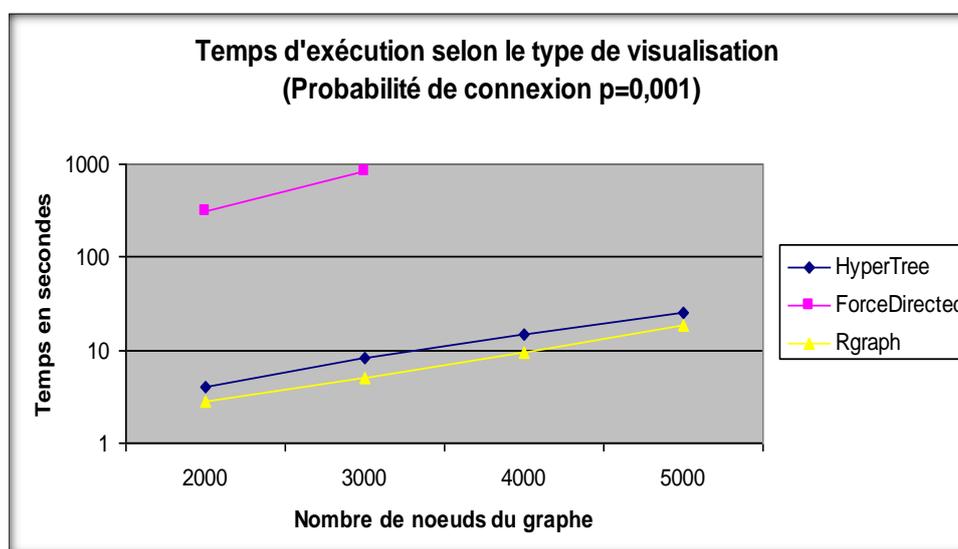


Figure 76 : temps d'exécution selon le type de visualisation

Cela peut facilement s'expliquer par une complexité accrue avec la simulation des lois des forces. Pour cet algorithme, nous n'avons pas pu relever les temps pour $n=4000$ et $n=5000$ (temps d'exécution supérieur à 30 minutes).

RGraph semble à peine plus performant que HyperTree. En revanche, ForceDirected ne peut être adapté qu'aux petits graphes ($n < 100$).

Nous avons utilisé une échelle logarithmique pour les temps d'exécution pour atténuer la distance entre les temps de ForceDirected et HyperTree/RGraph.

Dans les figures qui suivent (Figure 77, Figure 78 et Figure 79), nous avons détaillé les différents temps d'exécution pour différentes probabilités de connexion du graphe et selon le nombre de nœuds utilisés.

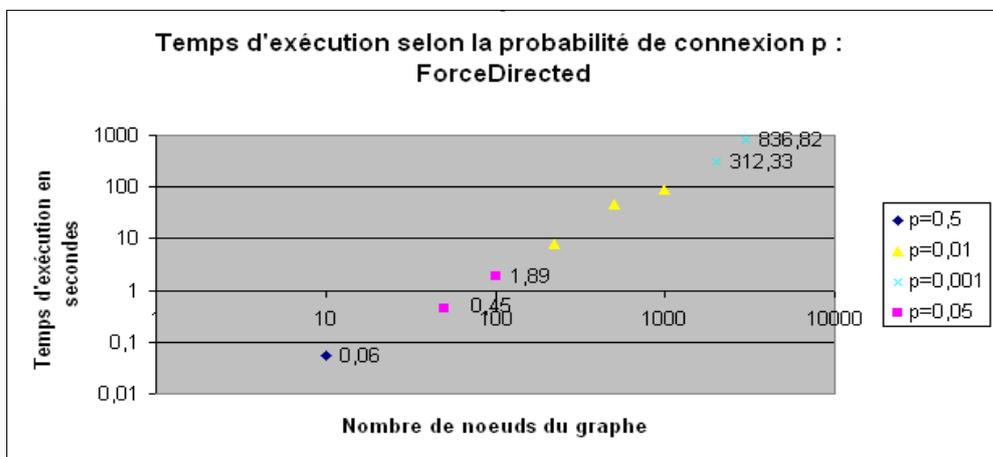


Figure 77 : temps d'exécution pour ForceDirected

Nous pensons que pour pouvoir visualiser l'ensemble des unités sous la forme d'un graphe, nous opterons plutôt pour le type HyperTree ou RGraph, vu que les valeurs de temps restent acceptables même pour n=5000.

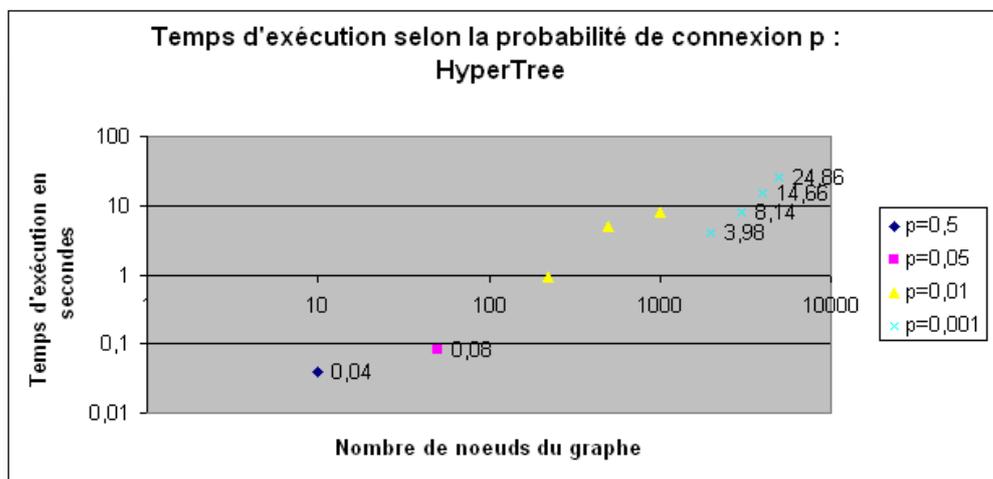


Figure 78 : temps d'exécution pour HyperTree

Nous privilégierons tout de même ForceDirected pour des sous-graphes que nous appelons parcours (ce type de graphe ne dépasse pas les 100 nœuds). Son intérêt réside essentiellement dans la malléabilité de la navigation où l'utilisateur peut déplacer les nœuds et ainsi modéliser et personnaliser la vue qui s'adapte le mieux à ses préférences.

Hormis les différents temps mesurés et les performances constatées pour ces différents tests, il s'agit surtout, et c'est ce qui nous intéressait en priorité, de voir comment il est possible d'exploiter ces résultats pour énoncer de nouvelles orientations.

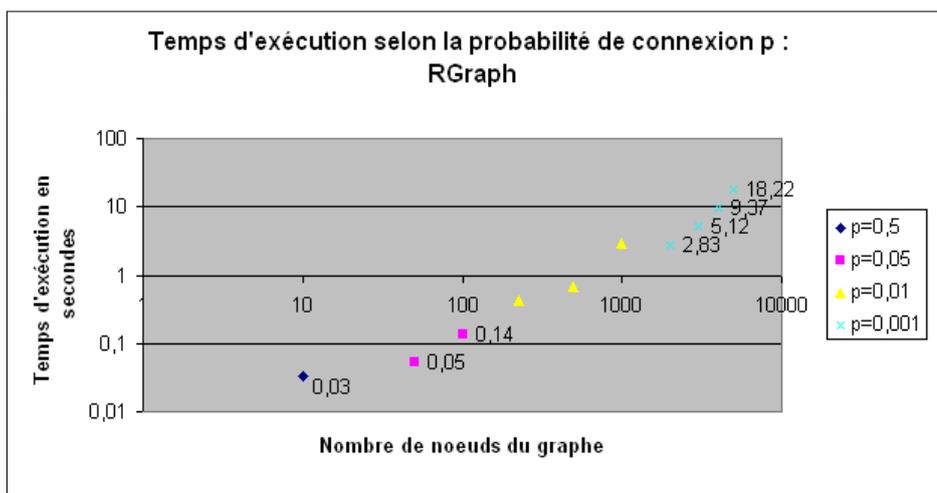


Figure 79 : temps d'exécution pour RGraph

Grâce aux temps d'exécution constatés et aux tests de passage à l'échelle, nous pouvons ainsi corréler un type de visualisation à :

- **une situation d'apprentissage :**
 - Rgraph ou encore HyperTree seraient préconisés pour une navigation libre (graphe complet des épistèmes) ;
 - ForceDirected pour une navigation guidée (parcours, sous-graphes), ce qui correspondrait à une navigation orientée tâche dans le cas France-IOI).

- **une situation d'édition :**
 - Pour une édition intra-graphe, Force Directed pourrait être plus stable dans des questions de suppressions/ajouts de liens ou de nœuds car il intègre nativement les graphes acycliques orientés ;
 - Pour une édition inter-graphes : HyperTree/Rgraph pourrait apporter plus de souplesse au niveau de la composition, mise en lien de plusieurs portions de graphes.

V. Conclusion

La mise à l'épreuve de plusieurs techniques de visualisation et de navigation dynamiques à l'aide de plusieurs types de graphes, nous a permis d'évaluer et de faire un comparatif en termes d'efficacité temporelle. Nous avons vu que ForceDirected, vu sa complexité, est plutôt conseillé pour les petits graphes ne dépassant pas les centaines de nœuds.

En revanche, HyperTree et RGraph peuvent être utilisés pour tous autres types de situations même s'il est parfois difficile de représenter un graphe orienté acyclique. Dans les prototypes que nous avons eu à implémenter, le graphe des épistèmes était composé de deux-cent vingt-et-un épistèmes pour le prototype PERCEPTOR et de soixante-neuf pour le prototype HiPPY. Une perspective d'utilisation et d'exploitation de ces résultats serait par exemple pour construire de grands et de très grands graphes lorsqu'il s'agit d'un domaine d'apprentissage complexe ou qui nécessiterait un découpage fin en épistèmes, et ainsi on avoisinerait les milliers de nœuds.

Chapitre 2 : Terrain et expérimentation de l'hypermédia épistémique HiPPY

I.	INTRODUCTION	147
II.	TERRAIN EXPERIMENTAL.....	147
III.	METHODOLOGIE	147
1.	<i>Hypothèses de départ</i>	147
2.	<i>Scénarios et tâches</i>	148
3.	<i>Utilisabilité et utilité en EIAH</i>	148
4.	<i>Moyens et mise en œuvre</i>	149
5.	<i>Contexte et étapes de l'expérimentation</i>	149
a.	Une pré-expérimentation.....	149
b.	Une expérimentation à grande-échelle	149
IV.	CONCLUSION	150

I. Introduction

Ce chapitre a pour ambition d'installer tous les éléments empiriques nécessaires au déroulement du processus expérimental qui aura pour but de mettre à l'épreuve nos différentes modélisations, choix de conception et autres orientations de recherche à travers le prototype HiPPY. À cet effet, nous présentons le terrain expérimental envisagé qui inclut les utilisateurs cibles et les conditions générales. Dans un second temps, nous décrirons la méthodologie suivie de recueil de données issues de traces et d'un questionnaire liés à nos hypothèses de recherche. Il y sera également question des tâches et de la manière dont seront évalués l'utilisabilité, l'utilité du prototype ainsi que les moments de cette expérimentation.

II. Terrain expérimental

Parmi les 3 prototypes réalisés (la séquence CloudSCORM, le prototype PRECEPTOR et enfin le module HiPPY), il nous a fallu choisir lequel mettre à l'épreuve dans le cadre d'une expérimentation authentique (par de vrais apprenants dans une situation d'apprentissage définie).

Concernant le terrain expérimental, nous avons finalement opté pour la plateforme France-IOI. Le principal avantage est la présence d'une communauté d'apprenants (sur 865 utilisateurs inscrits ayant opté pour le langage Python, plus de 721 apprenants ont réussi au moins 1 exercice entre septembre 2012 et décembre 2013) et également la participation de quelques enseignants (i.e. dans le cadre de l'option ISN). De surcroît, il y a une importante séquence d'exercices, chose qui manque dans le prototype PRECEPTOR et la séquence CloudSCORM.

Nous souhaitons tester l'utilisabilité et l'utilité de la carte de navigation et mettre en évidence les mécanismes d'aide à la navigation à travers : le graphe des épistèmes, la navigation dynamique et l'évaluation épistémique individualisée. Les facteurs expérimentaux sont : le prototype HiPPY et les traces de navigation sur le graphe

III. Méthodologie

1. Hypothèses de départ

En réalité nous disposons de trois familles d'hypothèses liées :

- aux aspects d'utilisabilité et d'ergonomie ;
- aux aspects potentiellement assistants de la carte épistémique (aide à la navigation, évaluation épistémique individualisée) ;
- et au potentiel d'évolutivité du graphe.

Elles sont destinées aux apprenants, mais parfois aux concepteurs/enseignants. Comment évaluer l'aide à la navigation dans une structure non linéaire (réseau des unités épistémiques du langage Python) ? Un moyen pour y parvenir est d'abord d'observer si le futur outil intégré est bien utilisé (taux d'usage par rapport aux autres onglets, temps moyens de passage, etc...) et recourir à un questionnaire pour justifier les différents choix et cheminements).

Voici maintenant les principales hypothèses qui devront être évaluées :

- H1: *la carte épistémique aide l'apprenant dans la résolution de problèmes ;*
H2 : *la carte des épistèmes est un moyen de consolidation de connaissances ;*
H3 : *la carte épistémique permet de visualiser les concepts utilisés (et leurs dépendances) lors d'exercices d'entraînement. Ex : afficher des chaînes en mode interactif -> encodage. ;*
H4 : *la carte épistémique permet de voir les concepts maîtrisés d'une façon individualisée ;*
H5 : *plus l'exercice est difficile, plus l'apprenant aura recours à la carte épistémique.*

Pour évaluer ces hypothèses, nous nous proposons de recueillir et d'analyser les traces des apprenants. Mais pour éviter les inférences hasardeuses, nous avons aussi interrogé les apprenants par l'intermédiaire d'un questionnaire. Nous verrons (au chapitre 3 de cette partie III) quels futurs usages et stratégies liés au graphe des épistèmes sont identifiables. Un autre aspect de recherche concernerait l'influence de la navigation (selon les statistiques des autres apprenants ou l'historique) sur la présentation ou le contenu de l'outil intégré.

2. Scénarios et tâches

Les tâches sont les suivantes :

- chercher les unités contenant les concepts nécessaires à la résolution d'un problème donné.
- Les évaluer et les éditer si nécessaire par rapport à un commentaire d'un autre enseignant.
- Dériver une copie et la modifier.

Un enseignant qui donne déjà son cours peut, s'il le veut, l'enrichir avec les unités contenues dans HiPPY et il peut, s'il le souhaite, réviser certaines ou en insérer d'autres pour le faire partager à la communauté d'apprenants en cours de formation. Bien évidemment, ceci ne peut être fait que sous réserve d'acceptation et d'une manière volontaire.

Un étudiant travaillant en autonomie souhaiterait commencer par une navigation libre en choisissant une unité parmi celles proposées dans la carte épistémique, et commencer des exercices d'entraînement en s'aidant de cette carte. Par contre, il peut également recourir à des problèmes. Dans ce cas, la carte épistémique n'est dévoilée que sous forme de rétroactions pour le faire approcher de la solution. Le graphe des épistèmes peut servir également comme appui. C'est dans cette optique qu'il a été utilisé prioritairement.

Un exemple de tâche serait de commencer par ouvrir un énoncé d'exercice, parcourir les différents épistèmes utiles, proposer et exécuter une solution. Si la solution n'est pas correcte, il peut soit utiliser des indices ou bien faire des allers-et-retours sur les notions qu'il ne maîtrise pas encore à travers les épistèmes correspondants. Il peut également reprendre la résolution d'exercices d'entraînement assez proches (associés à ces épistèmes non encore maîtrisés) ou encore regarder quelles autres unités épistémiques sont utiles (directement ou indirectement) à la résolution si la solution n'est pas totalement satisfaisante.

3. Utilisabilité et utilité en EIAH

Plusieurs méthodologies existent pour tester l'utilité et l'utilisabilité d'un EIAH (Nogry, Jean-Daubias & Ollagnier-Beldame, 2004). Nous tenterons d'évaluer les questions d'utilisabilité et l'utilité pédagogique dans une moindre mesure.

Cela comprend certains critères d'utilisabilité en termes d'IHM : ergonomie, reprise sur erreurs, cohérence, connaissance, mémorisation, ou encore apprentissage.

En ce qui concerne l'utilité en tant qu'EIAH, une des applications pourrait être d'étudier précisément les stratégies utilisées pour résoudre les exercices et déduire ensuite la place qu'occupe le prototype HiPPY dans le processus de résolution.

En outre, pour l'enseignant, cela pourrait être un outil support de création de contenu et d'édition intégrable dans un LMS (concerne plutôt l'archive SCORM réutilisable produite sur CloudScorm).

Il faut veiller tout de même à certains points comme la réduction de la charge cognitive (pour l'étudiant, la relation entre les parcours imposés et libres), ce qui nous amènera à étudier l'utilisation dans un premier temps et l'usage dans un second.

4. Moyens et mise en œuvre

Comme nous l'avons énoncé précédemment, nos outils de recueil de données seront fondamentalement l'analyse des traces et l'interprétation des résultats issus du questionnaire. Ceci sera nécessaire pour pouvoir identifier la population, de préférence ceux qui n'ont pas ou que peu de connaissance en programmation en Python. Nous pouvons estimer que l'outil sera plus profitable à un profil débutant en programmation vu que le graphe des épistèmes initiale a été pensé dans ce but.

Les autres types de questions peuvent concerner :

- la flexibilité de la vue : avec quelle facilité est-il possible de changer la vue ? Quels degrés de liberté chez l'utilisateur ? (déplacement nœuds, arcs, déplacement global, zoom, focus...);
- le degré d'utilité de la dissimulation de certaines régions du graphe (éviter la surcharge cognitive notamment dans les vues RGraph utilisées par HiPPY?).

Comment est-il possible d'évaluer l'interactivité avec l'objet graphe épistémique ? Le type de visualisation permet-il d'exploiter les unités sans en connaître les intitulés ?

5. Contexte et étapes de l'expérimentation

a. Une pré-expérimentation

Une pré-expérimentation a été prévue tout de suite après l'intégration du prototype HiPPY pour valider les fonctionnalités générales et surtout de mettre à l'épreuve le graphe des épistèmes.

b. Une expérimentation à grande-échelle

Il s'agit de l'expérimentation à proprement dite qui concerne d'une part des apprenants appartenant à un groupe ISN et de l'autre les utilisateurs inscrits d'une façon volontaire. En revanche, tous les apprenants ciblés par cette expérimentation ont choisi comme langage de référence le langage Python.

IV. Conclusion

Cette section a posé les jalons nécessaires pour le recueil de données expérimentales et cela à travers la collecte des traces et la soumission d'un questionnaire en ligne. Nous avons énoncé les caractéristiques du terrain expérimental envisagé, les différents éléments d'utilisabilité, d'utilité à évaluer ainsi que certaines hypothèses en relation avec le graphe des épistèmes et son éventuelle assistance à la navigation individualisée. Toutefois, cela reste ébauché sans les résultats effectifs issus de cette expérimentation.

Une piste serait de destiner cet outil aussi aux enseignants pour :

- pouvoir dire si la carte épistémique est parfois utile pour pouvoir éditer une portion du graphe ou du réseau en gardant une cohérence globale -> contraintes du graphe des unités ;
- voir si des unités non prévues dans le parcours pré-suggéré ont été visités -> encourager à visiter d'autres unités non prévues.

Le chapitre suivant nous donnera les résultats majeurs et les principales conclusions en rapport avec nos questions de recherche.

Chapitre 3 : Résultats et évaluations

I.	INTRODUCTION	152
II.	EVALUATION ANALYTIQUE AVEC LES CRITERES IHM	152
III.	RESULTATS IDENTIFIES A PARTIR DES TRACES	152
1.	<i>Données générales sur les trajectoires</i>	152
a.	Données pré-existantes sur les macro-trajectoires.....	152
b.	Données sur les résultats de la pré-expérimentation et de l'expérimentation à large échelle	153
2.	<i>Résultats des algorithmes de fouille</i>	156
a.	Clustering des performances des apprenants.....	156
b.	Règles d'association.....	157
c.	Arbres de décision.....	158
d.	Fouille de motifs (Pattern mining)	159
3.	<i>Constats et discussion</i>	161
4.	<i>Une première mise à jour des ressources à partir des données de navigation</i>	161
IV.	RESULTATS DU QUESTIONNAIRE	162
1.	<i>Un questionnaire destiné aux apprenants</i>	162
a.	Identifier le profil démographique des utilisateurs et leurs expérience en programmation	162
b.	Evaluer l'utilisabilité, l'efficacité et l'utilité	163
c.	Expliquer les actions et les préférences de l'utilisateur.....	163
d.	Résultats directs, croisés et discussion	163
e.	Principaux retours d'utilisateurs sur l'onglet « Notions associées »	167
▪	Problèmes	167
▪	Apports	167
f.	Principaux retours sur le système de navigation	168
▪	Problèmes	168
▪	Apports	168
g.	Propositions d'amélioration	168
V.	COMBINAISON DE RESULTATS ISSUS DU QUESTIONNAIRE AVEC LES DONNEES SUR LES TRACES	169
VI.	CONCLUSION	171

I. Introduction

Par son caractère crucial, la visée de ce chapitre est de couronner tout notre cheminement suivi dès le début de nos propositions, en passant par la construction du prototype HiPPY jusqu'au recueil des données et l'analyse des résultats. La première partie présente une évaluation analytique selon des critères normalisés d'interaction homme-machine. La deuxième montre les principaux résultats issus de données de traces. Ceci inclut évidemment l'identification de trajectoires selon les différents niveaux, l'utilisation d'outils et d'algorithmes de fouille. La troisième porte sur les résultats venant du questionnaire. L'ultime partie s'intéresse à la combinaison des données du questionnaire avec les traces.

II. Evaluation analytique avec les critères IHM

Nous rappelons les critères décrits par la norme ISO 9241-10 : l'adaptation à la tâche, le caractère auto-descriptif, le contrôle utilisateur, la conformité aux attentes de l'utilisateur, la tolérance à l'erreur, l'aptitude à l'individualisation, la facilité d'apprentissage. L'équivalent en AFNOR se présente sous forme de guidage, de charge de travail, de contrôle explicite, d'adaptabilité et de gestion des erreurs. Mesurer l'utilisabilité selon la norme ISO 9241 consiste à mesurer le degré selon lequel un produit peut être utilisé, par des utilisateurs identifiés, pour atteindre des buts définis avec efficacité, efficacité et satisfaction, dans un contexte d'utilisation spécifié. Nous tenterons de répondre à certains aspects d'évaluation à travers les données recueillies par le questionnaire (cf. section IV).

III. Résultats identifiés à partir des traces

Les résultats présentés dans cette section sont structurés selon les deux moments de l'expérimentation (Miled, Reffay, & Laroussi, 2014) :

- une pré-expérimentation qui a commencé en avril 2013 ;
- et l'expérimentation à grande échelle qui a débuté en septembre 2013.

Les résultats partiels obtenus à la phase de pré-expérimentation ont été confirmés lors de l'expérimentation à large échelle. Nous donnons en premier lieu des résultats généraux, puis nous procédons à l'analyse avec des algorithmes de fouille plus spécifiques.

1. Données générales sur les trajectoires

Voici les résultats selon les niveaux de trajectoires (macro, micro et épistémique). Certaines données pré-existantes (notamment sur les macro-trajectoires) ont été revues et rediscutées.

a. Données pré-existantes sur les macro-trajectoires

Sur 865 utilisateurs inscrits depuis septembre 2012 et ayant opté pour Python, 721 ont validé au moins un exercice, 438 en ont validé au moins six (correspond au nombre d'exercices par pour le 1^{er} chapitre) et 220 en ont validé au moins seize (nombre des exercices des deux 1^{ers} chapitres) et jusqu'à 104. Les deux premiers chapitres regroupent seize exercices. On constate que sur les 220 en ayant réussi au moins seize, 75 ont suivi strictement l'ordre prescrit sans

éviter aucun exercice (soit 34%). 66 ont sauté ou inversé entre 1 et 4 exercices (30%). Les 79 restants en ont sauté ou inversé cinq ou plus (soit 36%). Ces données initiales montrent une présentation plutôt séquentielle ne donnant pas une indication sur les dépendances entre les concepts. En dépit de cela, plus des deux-tiers des utilisateurs de la plateforme France-IOI qui ont résolu au moins seize exercices ont divergé de la séquence suggérée. Nous pensons que ces utilisateurs peuvent (1) bénéficier directement des notions associées relatives aux différents exercices et (2) leur permettre de mieux cerner leurs notions non encore validées.

À titre d'illustration (Figure 80), à partir des 721 utilisateurs ayant au moins réussi 1 exercice, nous avons tracé le nombre d'exercices résolus (ou réussis) (axe des abscisses) selon le nombre d'utilisateurs (axe des ordonnées). Ainsi, 57 utilisateurs ont résolu exactement 1 exercice, 24 ont résolu 2 exercices, 58 utilisateurs en ont réussi 3. Ce que nous pouvons noter, c'est que le maximum est atteint pour 4 exercices résolus (106 utilisateurs). Nous remarquons également que l'allure tend à décroître à partir de 15 exercices résolus et c'est à ce seuil que le nombre d'utilisateurs reste inférieur à 10. Nous faisons remarquer que le nombre d'exercices tentés mais non réussis (ou non résolus) n'entre pas ici en considération.

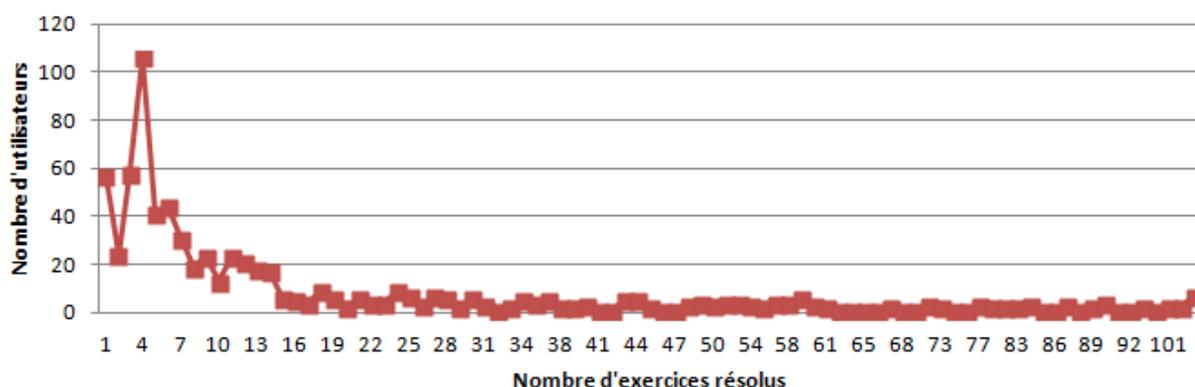


Figure 80 : nombre d'utilisateurs en fonction du nombre d'exercices résolus

b. Données sur les résultats de la pré-expérimentation et de l'expérimentation à large échelle

Le nombre des utilisateurs ayant utilisé l'onglet « Notions associées » a atteint 200 jusqu'au 18 mai 2013 (32 jusqu'au 12 avril, 80 jusqu'au 24 avril). Commençons par regarder les indications sur les trajectoires épistémiques. La Figure 81 présente un exemple de trajectoire. La première colonne est l'identifiant de l'utilisateur (21372), la seconde est l'identifiant de l'épistème, la troisième est l'estampille temporelle, la quatrième est l'événement associé à l'épistème en cours d'utilisation. La cinquième colonne distingue entre un graphe orienté-tâche (Task) ou générique et la dernière est l'identifiant de la tâche ou exercice (ici 1979). 62 épistèmes sur 69 ont été mobilisés. L'épistème « Afficher » est le plus utilisé. En huit jours seulement, sur les 567 instances (de trajectoires épistémiques), 530 étaient orientées-tâche, contre 37 génériques. Sur ces instances, 299 portaient sur l'événement epiGraphClick, 136 sur epiSeen, 78 sur epiClick et 54 sur epiGraphDoubleClick.

21372	14	05/04/2013 09:20	epiClick	Task	1979
21372	14	05/04/2013 09:20	epiSeen	Task	1979
21372	20	05/04/2013 09:22	epiClick	Task	1979
21372	20	05/04/2013 09:22	epiSeen	Task	1979

21372	63	05/04/2013 09:22	epiClick	Task	1979
21372	63	05/04/2013 09:22	epiSeen	Task	1979
21372	45	05/04/2013 09:23	epiClick	Task	1979
21372	45	05/04/2013 09:23	epiSeen	Task	1979
21372	7	05/04/2013 09:23	epiGraphClick	Task	1979
21372	63	05/04/2013 09:23	epiGraphClick	Task	1979
21372	6	05/04/2013 09:24	epiGraphClick	Task	1979
21372	26	05/04/2013 09:24	epiGraphClick	Task	1979
21372	61	05/04/2013 09:24	epiGraphClick	Task	1979
21372	63	05/04/2013 09:25	epiGraphClick	Task	1979
21372	4	05/04/2013 09:25	epiGraphClick	Task	1979

Figure 81 : extrait d'une trajectoire épistémique pour l'utilisateur 21372

Cet extrait de micro-trajectoire correspond au temps de passage sur l'onglet « Notions associées » représenté par TabEpistemes comme indiqué sur la Figure 82. On remarque bien qu'il s'est écoulé environ cinq minutes avant de passer à l'onglet TabEditor qui est l'éditeur intégré. Le détail des actions est donc facilement identifiable par la trajectoire épistémique vue précédemment sur la Figure 81.

21372	1979	05/04/2013 09:20	TabEpistemes	11/04/2013 12:11	69	13
21372	1979	05/04/2013 09:25	TabEditor	11/04/2013 12:11	69	13

Figure 82 : extrait de micro-trajectoire de l'utilisateur 21372

À partir de logs_user_task entre 2013-04-04 18:09:29 et 2013-04-12 18:05:24, 9737 instances ont été capturées : le nombre de tâches utilisées était de 113 / 155 (certaines tâches ne sont pas associées à des épistèmes en Python). D'autres ne nécessitent pas de soumettre de code, on aura comme état 'Read'.

Voici quelques données quantitatives détaillant les différentes distributions :

Read 1305, TabEditor 2735, TabEpistemes 541 (on doit pouvoir retomber sur ce même nombre d'instances en choisissant le bon intervalle temporel à l'aide de logs_user_epi), TabTask 1710, TabCorrection 809 (n'est généralement donné qu'après avoir soumis au moins une proposition de code), Submitted 1607, TabHelp 270, TabActivity 181: renseigne sur les tentatives des soumissions et leur résultat associé (iScore à 100 : validé), AdviceRequested 30, HelpRequested 3.

La Figure 83 présente une micro-trajectoire suivie pour résoudre le premier exercice (« Hello World !»). L'utilisateur 5379 a pu résoudre la tâche en 3 minutes. Il a également consulté les notions nécessaires après la correction. Plus bas, sont présentés les onglets correspondants aux différents événements tels que l'utilisateur les utiliserait pour résoudre cette même tâche : Read (correspond à sujet), TabEditor (est représenté par l'onglet) Résoudre et TabEpistemes (Notions associées). Nous verrons plus loin quel rôle pourrait jouer l'onglet « Notions associés » par rapport au processus de résolution.

No.	idUser	idTask	sDate	sType	sRegistrationDate	sLastLoginDate	nbSolved	iScore
79	5379.0	1869.0	07/04/13 06:41	Read	29/10/06	09/04/13 16:01	1355.0	100.0
80	5379.0	1869.0	07/04/13 06:41	Read	29/10/06	09/04/13 16:01	1355.0	100.0
81	5379.0	1869.0	07/04/13 06:43	TabEditor	29/10/06	09/04/13 16:01	1355.0	100.0
82	5379.0	1869.0	07/04/13 06:43	TabTask	29/10/06	09/04/13 16:01	1355.0	100.0
83	5379.0	1869.0	07/04/13 06:43	TabEditor	29/10/06	09/04/13 16:01	1355.0	100.0
86	5379.0	1869.0	07/04/13 06:43	Submitted	29/10/06	09/04/13 16:01	1355.0	100.0
84	5379.0	1869.0	07/04/13 06:43	TabCorrection	29/10/06	09/04/13 16:01	1355.0	100.0
85	5379.0	1869.0	07/04/13 06:44	TabEpistemes	29/10/06	09/04/13 16:01	1355.0	100.0



Figure 83 : extrait de micro-trajectoire de l'utilisateur 5379 pour la tâche 1869 (Hello World !)

Nous avons pensé également à mesurer les durées de passage dans les différents onglets. Selon la Figure 84, nous avons trois utilisateurs identifiés par 21134, 21867 et 21909 pour différentes tâches (1877, 1878, 1880 et 1881). Ces tâches sont les premières que les utilisateurs rencontrent (et donc les plus faciles en théorie), s'ils suivent le chemin prescrit. C'est pourquoi les durées sont parfois relativement faibles. La durée maximale dans cet extrait est d'environ sept minutes pour l'état de soumission de code (Submitted) qui fait partie de l'onglet TabEditor sur l'interface. Nous pouvons considérer que les temps inférieurs à trente secondes ne sont pas à prendre en compte du fait qu'ils peuvent expliquer une manipulation parfois hasardeuse chez l'apprenant.

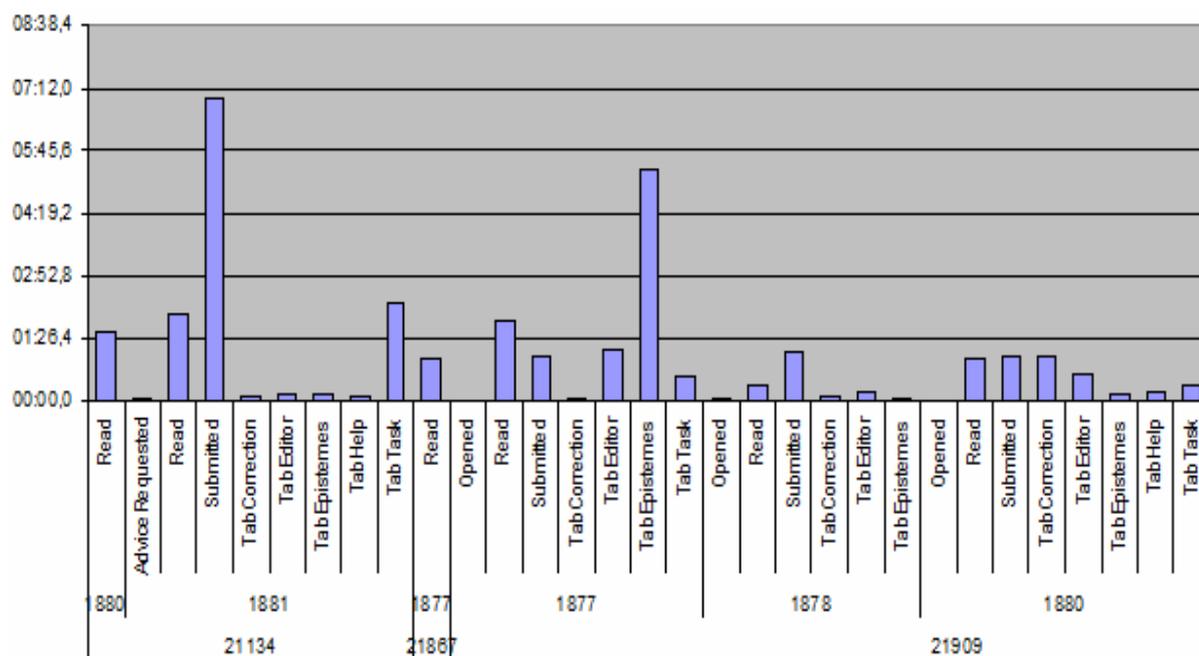


Figure 84 : temps passés sur les différents onglets pour trois utilisateurs (21134, 21867 et 21909)

Cette observation des temps de passage pourrait s'avérer insuffisante, c'est pourquoi nous avons pensé utile d'analyser l'ordre des onglets dans le but d'identifier des motifs pertinents faisant éventuellement intervenir TabEpistemes (voir la section III.2.c).

Quant à l'expérimentation à proprement dite, le nombre d'utilisateurs ayant au moins utilisé un épistème a dépassé les 858 (entre le 1^{er} septembre et le 18 octobre 2013). Ceci représente une bonne proportion (plus de 31%) par rapport aux 2762 inscrits sur la plateforme. 2105 ont sélectionné comme langage de référence Python. 1018 ont réussi à résoudre au moins 10

tâches. Sur cette même période, 8901 instances (90,94%) étaient orientées-tâche, 887 appartenaient au graphe générique. Ceci confirme l'usage du graphe des épistèmes principalement en rapport avec les exercices à résoudre. Comme sur la pré-expérimentation, l'épistème « Afficher » (ayant l'identifiant 8) était le plus utilisé. Sur cette même période, 66 sur les 69 épistèmes ont été employés. Les résultats sur la distribution des différents événements faisant intervenir le graphe ont confirmé (comme sur la pré-expérimentation) la domination de l'événement `epiGraphClick` avec 5549 (56,69%), 1181 pour `epiClick`, 2085 `epiSeen` et 973 `epiGraphDoubleClick` sur un total de 9788 instances. Cet aspect d'utilisation, montre que les utilisateurs ont tendance à naviguer dans le graphe sans forcément ouvrir le contenu d'un épistème.

2. Résultats des algorithmes de fouille

À l'aide des outils décrits dans la section III.3 du chapitre 5 de la partie précédente, nous avons eu principalement recours à la classification ou « *clustering* », aux règles d'association, aux arbres de décision et à la fouille de motifs.

a. Clustering des performances des apprenants

Outre le *clustering* hiérarchisé et agglomératif (Dendogram), nous avons opté pour l'algorithme K-means.

Dans nos requêtes énoncées et croisées, nous avons tenté de corrélérer le nombre d'exercices réussis ou tâches résolues (`nbSolved`) au nombre d'épistèmes mobilisés. Si l'on regarde le graphique généré par l'algorithme Kmeans-fast de RapidMiner (Figure 85) avec $n=4$, le nombre de classes (ou clusters), nous pouvons remarquer qu'il n'y a pas forcément de liens entre le nombre d'exercices résolus et le nombre d'épistèmes mobilisés.

En revanche, nous pouvons percevoir les classes que nous avons énoncées : ceux qui utilisent rarement le graphe des épistèmes et qui ont réussi moins de 25 exercices (en rouge), ceux qui ont réussi plus de 100 exercices et qui ne les utilisent pas (en bleu), ou encore ceux qui réussissent avec un usage assez élevé des épistèmes (en vert pâle et turquoise). Nous tenons à préciser qu'il peut y avoir certains biais, puisque l'intégration s'est faite au cours du mois d'avril 2013 :

Ce graphique comprend par exemple ceux qui étaient déjà inscrits avant l'intégration et qui n'ont pas forcément utilisé le graphe des épistèmes, et puis cela ne prend pas forcément en compte la progression de chaque utilisateur et la date d'inscription dans ce cas. Bien que les classes matérialisées par ce graphique soient indépendantes, une des spécificités de l'outil RapidMiner est de représenter les deux axes (nombre d'exercices résolus et nombre d'épistèmes mobilisés).

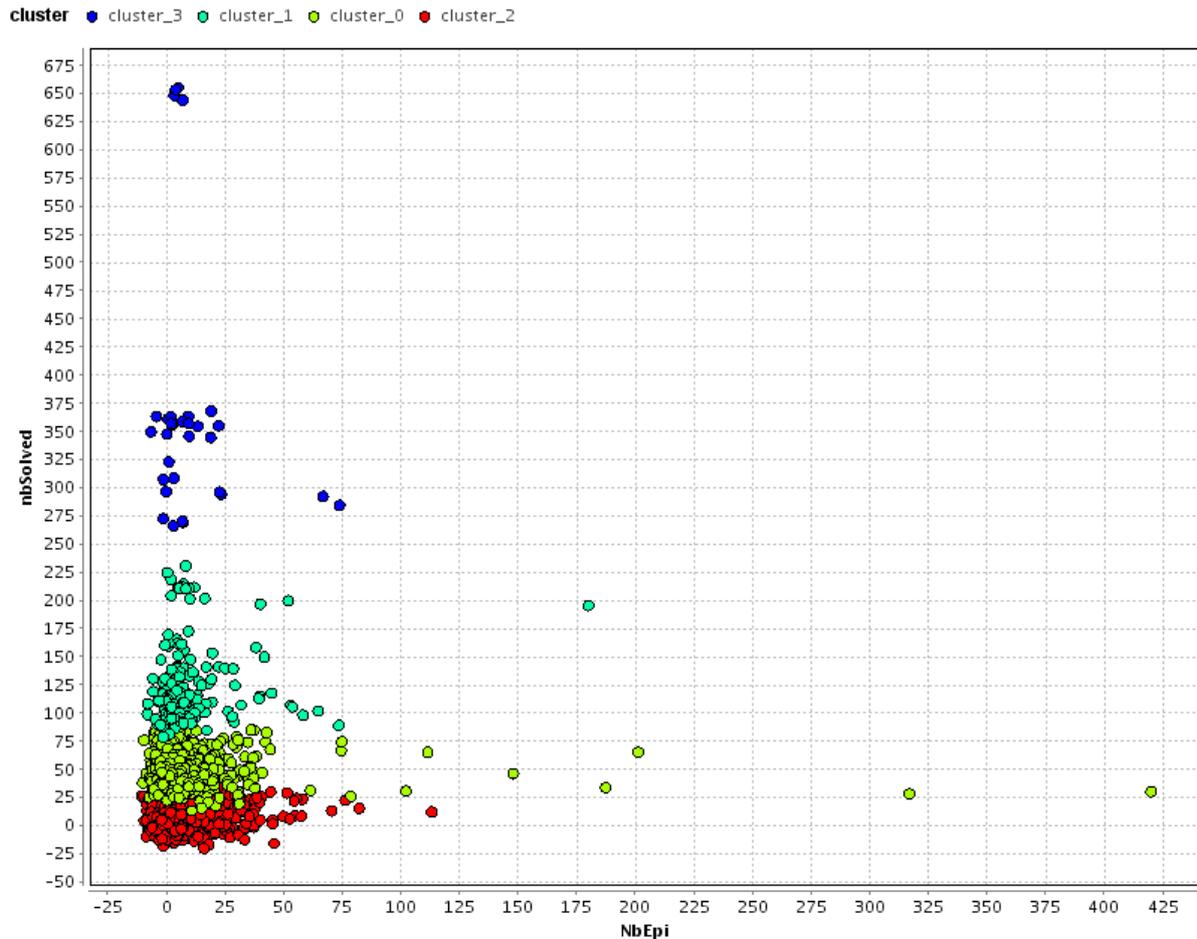


Figure 85 : résultat de Kmeans fast avec n=4

b. Règles d'association

Le principal objectif derrière l'extraction de règles d'association, est de pouvoir corréler certains faits constatés ou des tendances particulières qui mériteraient d'être observées. Ce qui nous semble le plus judicieux, est de percevoir des corrélations inattendues ou peu prévisibles. Pour cela, nous avons appliqué l'algorithme « *Apriori* » de Weka avec les paramètres suivants : la métrique utilisée est la confiance ayant une valeur minimale de 0,6 (voir la section III.3 du chapitre 5 de la partie II).

Nous avons utilisé un ensemble de 3083 instances avec un support minimum de 0,1 (308 instances), exploitant 4 attributs, le seuil de 14 exercices a été utilisé par rapport au nombre d'utilisateurs qui passe sous la barre des 10 utilisateurs (voir la section III.1.a), 15 épistèmes représente une valeur moyenne des dépendances d'un exercice de difficulté moyenne, 50 exercices est un choix arbitraire qui désignerait près de la moitié du nombre total des exercices considérés dans notre observation :

- sType : le type d'événement utilisé dans le graphe épistémique,
- I_50_NbSolved : désigne si le nombre d'exercices résolus est inférieur à 50,
- I_14_NbSolved : désigne si le nombre d'exercices résolus est inférieur à 14,
- et I_15_NbEpi : désigne si le nombre d'épistèmes utilisés est inférieur à 15.

Voici une sélection de règles obtenues :

1. $sType=epiGraphDoubleClick \ I_{50_NbSolved} < 50$ (346 instances) $\implies I_{15_NbEpi} \leq 15$ (338 instances) conf:(0.98)
2. $sType=epiGraphDoubleClick$ (428 instances) $\implies I_{15_NbEpi} \leq 15$ (413 instances) conf:(0.96)
3. $I_{15_NbSolved} \leq 14$ (1298 instances) $\implies I_{15_NbEpi} \leq 15$ (1225 instances) conf:(0.94)
4. $sType=epiClick \ I_{15_NbSolved} \geq 14$ (363 instances) $\implies I_{15_NbEpi} \leq 15$ (337 instances) conf:(0.93)

Nous pouvons remarquer certains faits déjà observés indirectement dans la phase de « *Clustering* ». Sur cet échantillon, la règle 1) nous apprend que ceux qui utilisent l'événement *epiGraphDoubleClick* et qui ont résolu un nombre inférieur à 50 exercices ont de fortes chances d'avoir un nombre d'épistèmes utilisés inférieur ou égal à 15 (la confiance est de 0,98). La règle 2) précise que les utilisateurs de l'événement *epiGraphDoubleClick* ont généralement utilisé moins de 15 épistèmes (cette règle s'applique avec une confiance de 96%). La règle 3) nous dit avec une confiance de 94% que ceux qui ont réussi à 14 exercices ou moins ont tendance à utiliser moins de 15 épistèmes indépendamment du fait qu'ils utilisent ou non l'événement *epiGraphDoubleClick*. Pour les utilisateurs ayant recours à l'événement *epiClick* et qui ont déjà résolu 14 exercices ou plus, le nombre des épistèmes utilisés serait inférieur ou égal à 15 (confiance de 0,93 selon la règle 4). Ces derniers éléments confirment partiellement la classification décrite dans la section précédente III.2.a.

c. Arbres de décision

La Figure 86 présente une portion d'arbre décisionnel généré par Knime qui repose sur une base de règles impliquant l'identifiant du contexte (*idContext*), c'est-à-dire l'identifiant de la tâche, l'identifiant de l'épistème (*idEpisteme*). Il y a également une indication sur les différents événements liés au graphe des épistèmes (*epiGraphClick*, *epiSeen*, *epiClick* et *epiGraphDoubleClick*). Que peut-on alors déduire ? Même si cette portion ne donne pas de notables éclaircissements, nous pouvons remarquer certains faits triviaux. Si le contexte (identifiant de la tâche) ≤ 646 alors l'identifiant de l'utilisateur > 17965 , nous pouvons penser qu'il s'agit de tâches moins difficiles à résoudre et que cela concerne les derniers utilisateurs enregistrés (tout utilisateur inscrit aura un identifiant supérieur à un utilisateur inscrit avant). Si le contexte de la tâche est supérieur à 646, l'utilisateur aura plus tendance à regarder le contenu des épistèmes (événement *epiSeen* à 25,3% contre 14,1% si le contexte de la tâche est inférieur ou égal à 645). Cela peut nous indiquer que pour les tâches les plus difficiles l'utilisateur aura sensiblement plus recours aux contenus des épistèmes.

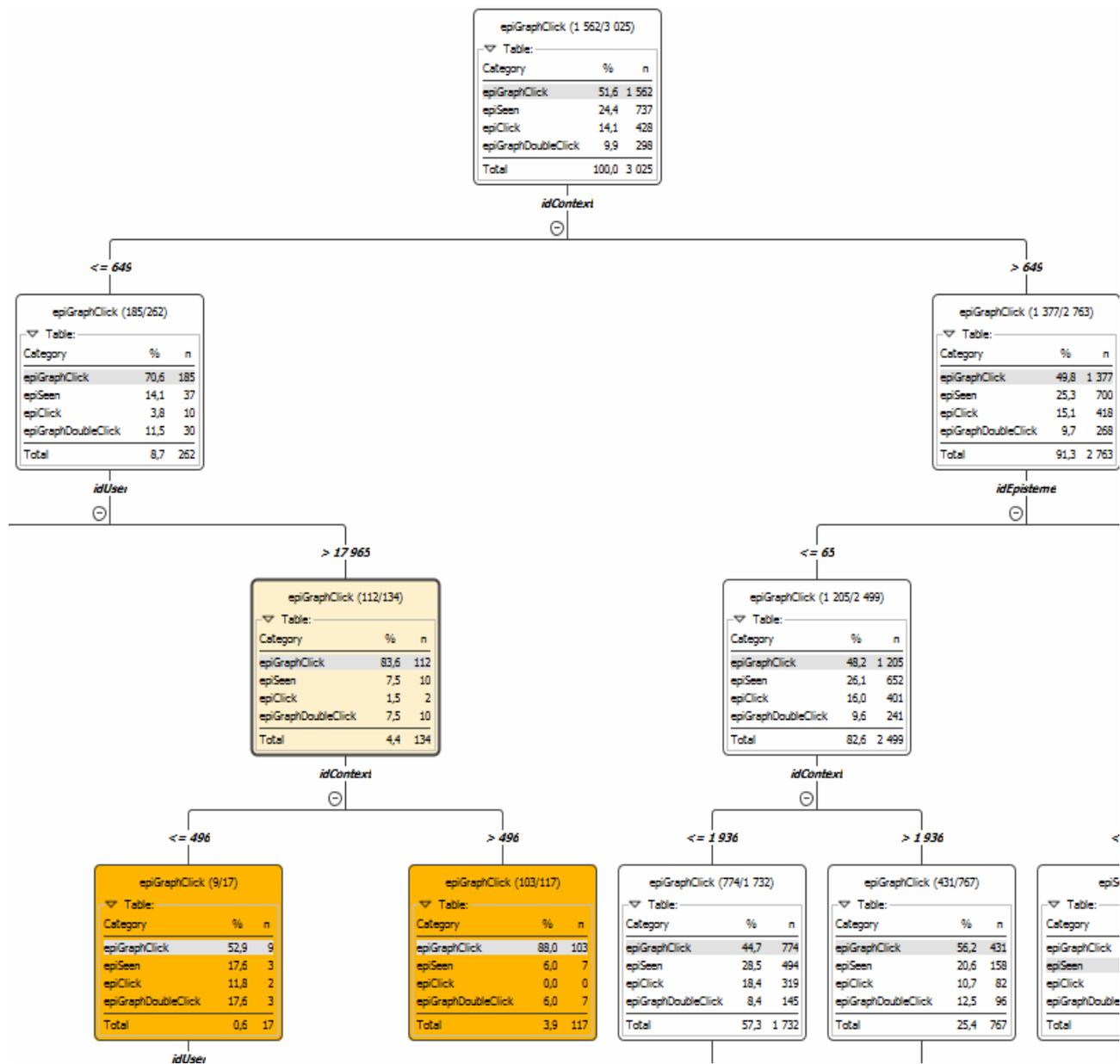


Figure 86 : arbre partiel de décision généré par Knime

d. Fouille de motifs (Pattern mining)

La Figure 87 décrit une analyse limitée (à 50 micro-trajectoires) des différents onglets utilisés. L'objectif de cette analyse est de mettre en évidence les motifs les plus fréquemment utilisés. On peut voir sur la Figure 87 les différents onglets visités après la lecture de l'énoncé de la tâche (Onglet Read). Sur les cinquante cas, six utilisent l'onglet TabEpistemes ou encore TabCorrection après une lecture (Read). 29 fois sur 50, c'est l'onglet « TabEditor » (pour écriture du code) qui suit tout de suite la lecture de l'énoncé. Si l'on continue le cheminement, sur les vingt-neuf cas, dix concernent l'état de soumission (Submitted) et sept pour TabEpisteme qui est suivi par TabCorrection. Les séquences les plus fréquentes sont donc : (*, TabEpistemes, TabCorrection) et (*, TabCorrection, TabEpistemes). Cela montre à l'évidence deux potentiels usages ou stratégies impliquant l'onglet TabEpistemes :

- Une **aide à la résolution** : (avant soumission) :
C'est un moyen pour avoir plus d'informations sur les concepts nécessaires à une tâche. Cette stratégie est liée au motif (*, TabEpistemes, TabCorrection) ;
- Une **aide à la consolidation** : (après soumission) : Il s'agit d'une série d'actions visant plus à vérifier des concepts manipulés après une soumission réussie. Cela correspond au motif (*, TabCorrection, TabEpistemes).

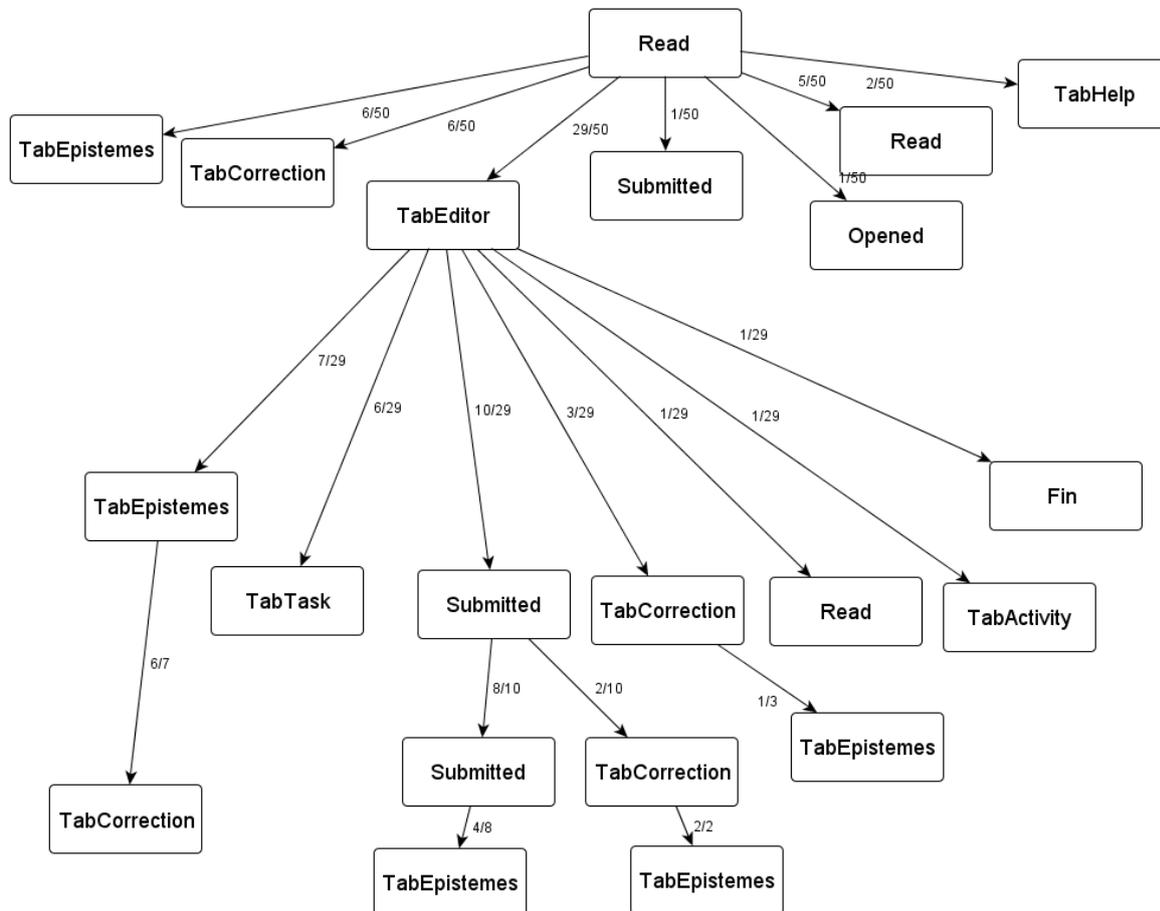


Figure 87 : motifs récurrents dans les micro-trajectoires

Voici maintenant un résultat de détection des motifs récurrents en utilisant GSP (General Sequence Pattern mining), ce résultat confirme que le motif (Read, TabEditor) est le plus fréquemment rencontré.

Nous rappelons que le support (ou couverture) est régi par la formule suivante :

$$\text{support (ou couverture)} = \frac{\text{Nombre d'items satisfaisant la règle}}{\text{Nombre d'items total}}$$

À gauche nous retrouvons le calcul du support et à la droite les items correspondants.

0.934: <Read> <TabEditor>
0.835: <Read> <TabCorrection>
0.851: <Read> <TabTask>
0.900: <Read> <Read>
0.646: <Read> <TabHelp>

Voyons maintenant les différents enseignements de ces résultats dégagés.

3. Constats et discussion

L'interface est plutôt prescriptive et soutenue par la présentation. Concernant les macro-trajectoires, la séquence des exercices semble linéaire. Pour ce qui est des micro-trajectoires, il existe des onglets systématiques (avec passage obligatoire), et le bouton « avancer » incite à suivre ces onglets d'une façon linéaire.

Quelle fréquence des items constituant la micro-trajectoire ? Fréquence de TabEpistemes ? C'est une question intéressante surtout dans le cas où la tâche n'est pas résolue dès la 1ère soumission.

Nous avons remarqué que les élèves avaient tendance à regarder les épistèmes après la correction : il s'agit du pattern (Submitted, TabCorrection, TabEpistemes); (TabEditor, TabCorrection, TabEpistemes); (TabCorrection, TabEpistemes). Globalement, l'ordre suivi est celui qui est prescrit (items dans une micro-trajectoire) et la liste de tâches.

Hormis les profils d'utilisateurs à analyser :

- ceux n'utilisant jamais le graphe des épistèmes,
- ceux l'utilisant rarement,
- ceux qui n'ont résolu que n tâches sous un certain seuil (indicateur nbSolved),
- ou encore ceux l'utilisant systématiquement,

Voici quelques autres éléments de réflexion :

- l'analyse de trajectoires : différents types de trajectoires au niveau micro et au niveau macro : le type de trajectoire est une stratégie potentielle de résolution ;
- l'efficacité d'une trajectoire se traduit par une trajectoire qui débouche sur une soumission réussie ou au moins sur un nombre de soumissions "raisonnable" ;
- et l'association d'un profil d'apprenant à une stratégie particulière ou à un type de trajectoire pour savoir quels épistèmes ont permis de débloquent la résolution.

4. Une première mise à jour des ressources à partir des données de navigation

Comme décrit dans le chapitre 3 de la partie II (cf. section IV), nous avons remarqué que les apprenants empruntaient des micro-trajectoires selon l'ordre prescrit (dans des proportions élevées). Il a donc été judicieux de changer l'ordre des onglets en avançant l'onglet « Notions

utiles » entre les onglets « résoudre » et « conseils ». Cela nous a permis également d'identifier deux stratégies d'utilisation des épistèmes (aide à la résolution et aide à la consolidation), alors que trivialement dans une première étape, seul l'aspect consolidation paraissait visible. Mais ce processus de révision nécessite d'être paramétrable pour permettre aux concepteurs pédagogiques de réorganiser l'interface de présentation ou de contenu en fonction des stratégies à privilégier.

IV. Résultats du questionnaire

Au-delà de pouvoir interroger les retours des utilisateurs et leurs attentes futures sur les principes de l'hypermédia HiPPY, notre objectif était également de croiser certaines données subjectives avec d'autres données plus objectives. En effet, le fait de croiser les traces de l'apprenant avec ses retours permet de mieux comprendre ses choix, ses orientations, chose qui n'est pas forcément détectable sur les traces seules ou sur les réponses aux questionnaires seules. Derrière cela, il y a aussi la volonté de conforter les hypothèses par la conjugaison des traces et des questionnaires.

Notre démarche stipule volontairement une soumission d'un questionnaire qui ne doit pas surcharger l'utilisateur avec un nombre important de questions (le nombre est inférieur à 18). Nous avons gardé tout de même quelques aspects d'évaluation systématiques comme les critères de l'INRIA. Ce que nous pouvons faire c'est comparer l'étude selon ces critères et faire des comparaisons, lorsque cela est possible, avec les retours des utilisateurs.

1. Un questionnaire destiné aux apprenants

Un questionnaire (voir annexe VIII) a été soumis aux utilisateurs pour évaluer et donner des éléments d'amélioration du prototype HiPPY. Ce questionnaire a été aussi utile pour valider certaines hypothèses sur le profil des apprenants (débutant, intermédiaire, avancé), sur le degré de pratique de la programmation (pas nécessairement avec le langage Python) et sur la proportion des utilisateurs appartenant à l'option ISN. Techniquement, il s'agissait d'un document Google où chaque utilisateur pouvait soumettre ses réponses en ligne. Nous avons toutefois conditionné l'accès au questionnaire par le fait qu'il devait au moins avoir résolu dix problèmes. Voici les principales questions posées :

a. Identifier le profil démographique des utilisateurs et leurs expériences en programmation

Il a été demandé aux utilisateurs de donner leur genre, âge (optionnel), et la dernière classe fréquentée ou le dernier diplôme obtenu. Nous avons également voulu savoir s'ils faisaient partie d'une classe ou d'un groupe ISN et quel était leur profil en programmation. Nous avons alors donné les questions suivantes : « quel est votre niveau de maîtrise en programmation avant d'étudier Python sur France-IOI ? », les différentes possibilités de réponses étaient : « j'avais déjà écrit plus de 10000 lignes de programme dans un langage », « j'avais fait quelques essais dans au moins 2 autres langages », « j'avais de bonnes notions en programmation mais pas de pratique », « je découvre la programmation pour la première fois ».

b. Evaluer l'utilisabilité, l'efficacité et l'utilité

Voici à présent quelques éléments pour évaluer la satisfaction de l'utilisabilité et l'efficacité de l'évaluation épistémique individualisée :

- « Le système de navigation (graphe dynamique) dans les "notions associées" vous semble (Non satisfaisant, peu satisfaisant, satisfaisant, très satisfaisant) ? »
- « Le système de navigation dans les "Notions associées" vous permet de voir les notions maîtrisées d'une manière (non satisfaisante, peu satisfaisante, satisfaisante, très satisfaisante) ? »
- Quant à l'évaluation de l'utilité de l'onglet « Notions associés », l'utilisateur devait choisir entre « inutile, parfois utile, utile ou très utile ».

c. Expliquer les actions et les préférences de l'utilisateur

Pour avoir quelques renseignements sur les choix, les actions et les préférences des utilisateurs, nous avons eu recours à deux questions.

- La 1^{ère} portait sur l'élément qui était le plus apprécié des utilisateurs. Les réponses possibles étaient : « l'animation et le centrage des nœuds », « l'affichage des concepts maîtrisés » et « les rappels sur une notion » ou « les prérequis d'une notions ».
- La seconde demandait la raison principale de leur utilisation de l'onglet « Notions associés ». Il était alors permis de choisir entre : « pour vérifier quel concept est lié à tel autre », « pour afficher les concepts assimilés », « pour voir les prérequis d'un exercice » et enfin « pour donner des rappels sur certains concepts » ou « autre raison à spécifier ».

d. Résultats directs, croisés et discussion

Nous avons recueilli 121 réponses (entre le 5 novembre 2013 et le 4 Décembre 2013) au lieu de 131 (dix valeurs en doublons ont été détectées). Nous rappelons que l'utilisateur doit répondre d'une manière unique avec la condition suivante : seuls les utilisateurs ayant au moins résolu (réussi) 10 problèmes en Python peuvent accéder au questionnaire en ligne.

Voici à présent les premiers résultats directs. Les données à plat donnent les proportions suivantes :

- 25% des utilisateurs sont de sexe féminin.
- 52% de l'ensemble des utilisateurs ont peu programmé (moins de 10 lignes de code), contre 21% entre 10 et 100.
- 41% découvrent la programmation pour la première fois et 22% ont une bonne connaissance mais sans pratique.
- 42% ont commencé la programmation en 2013.
- 38% du total des utilisateurs appartiennent à un groupe ISN¹⁰⁰.

Tous ces éléments dénotent un profil dominant de débutant en programmation. En ce qui concerne la satisfaction de la navigation dynamique (Figure 88), 50% sont satisfaits (ceci comprend la proportion des utilisateurs très satisfaits). Ce pourcentage est plus élevé (Figure 89) quand il s'agit d'utilisateurs de sexe masculin et n'appartenant pas à des groupes ISN (n = 58).

¹⁰⁰ Informatique et sciences du numérique : nouvel enseignement de programmation en France introduit en septembre 2012 en terminale S.

Le taux de satisfaction atteint alors 58% (avec 10% très satisfaits). Il passe à 59% (3% très satisfaits), quand il s'agit de sujets masculins ayant le bac et non appartenant à un groupe ISN (n=32).

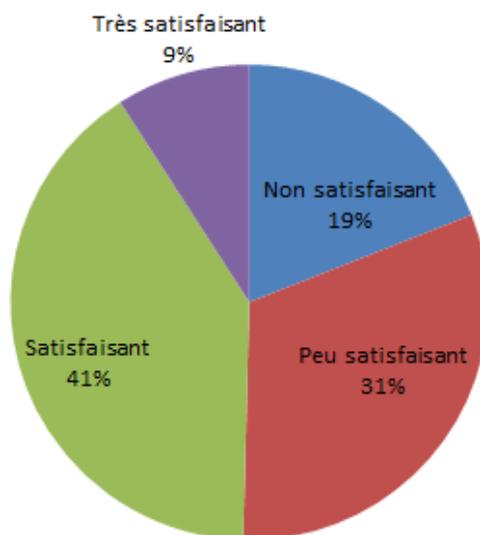


Figure 88 : évaluation de la navigation dynamique (n=121 : tous)

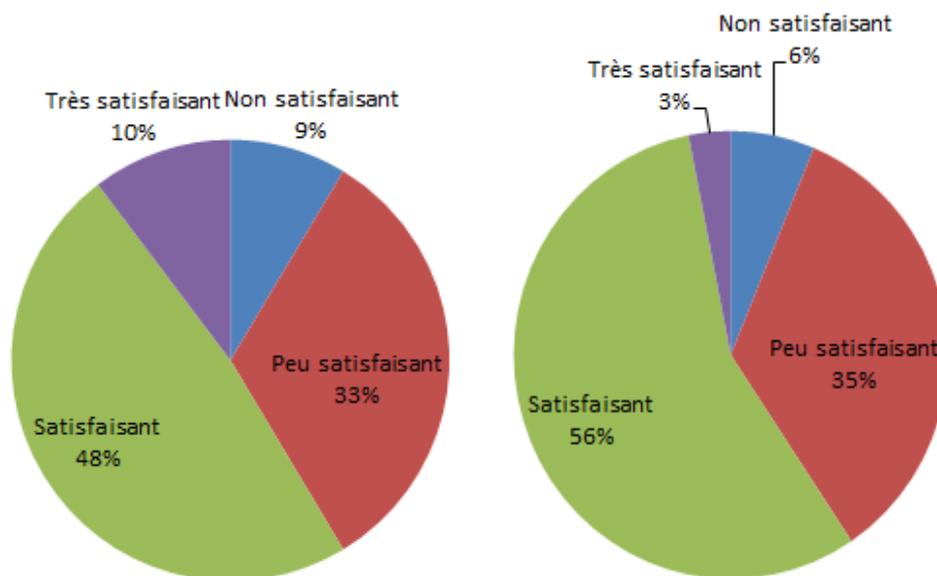


Figure 89 : évaluation de la navigation dynamique (à gauche n=58 : garçons hors ISN, à droite n=32 : garçons bacheliers hors ISN)

Pour les sujets masculins (Figure 90) ayant déjà écrit plus de 10000 lignes de code (n=14) le taux de satisfaction général est à 64% (dont 21% très satisfaits).

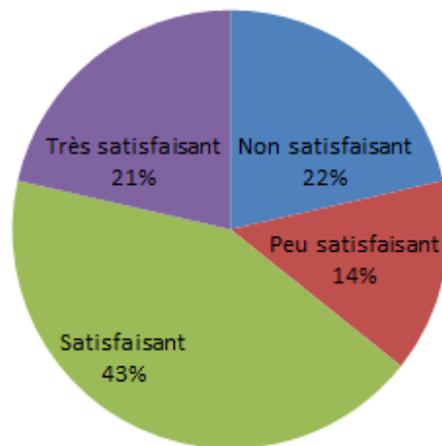


Figure 90 : évaluation de la navigation dynamique (n=14, programmeurs non débutants)

S'agissant de l'évaluation épistémique individualisée (Figure 91), 64% des utilisateurs (n=121) trouvent qu'elle est satisfaisante (dont 8% très satisfaisante). Pour les utilisateurs de sexe masculin n'appartenant pas à des groupes ISN (n = 58), l'évaluation épistémique individualisée est très satisfaisante à 9%, satisfaisante à 62%, peu satisfaisante à 22% et insatisfaisante à 7%.

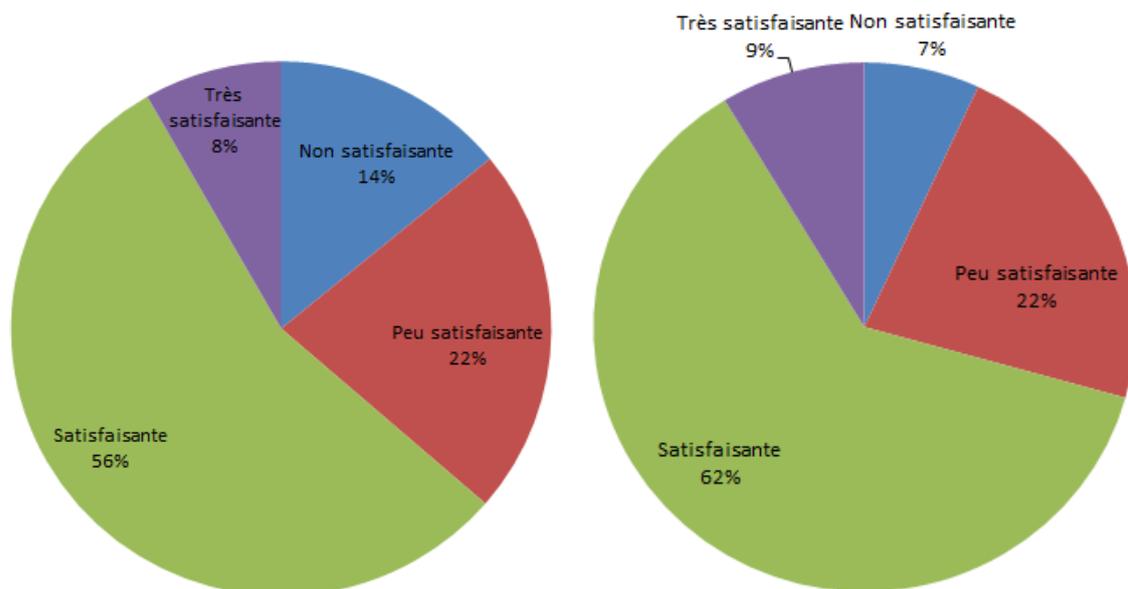


Figure 91: évaluation épistémique individualisée (à gauche n=12 : tous, à droite n=58 : garçons hors ISN)

Pour ceux qui n'appartiennent pas à un groupe ISN et qui découvrent la programmation pour la première fois (n=26), le taux général de satisfaction atteint 81% (15% très satisfaits).

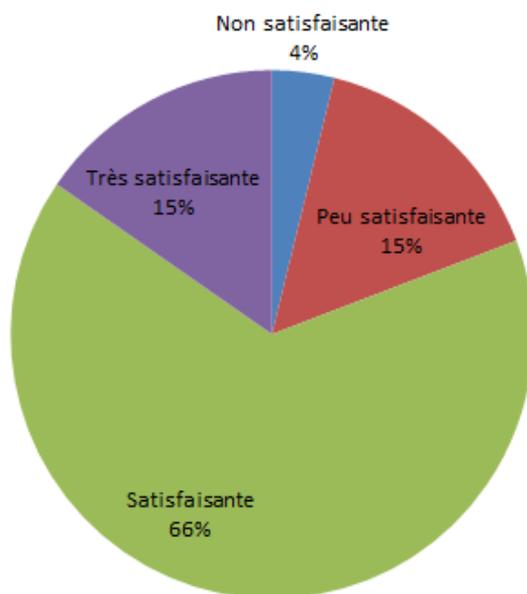


Figure 92 : évaluation épistémique individualisée (n=26 : débutants hors ISN)

Sur 121 utilisateurs, 33% trouvent l'outil HiPPY utile, 11% très utile, 35% parfois utile et 21% inutile, mais à l'inverse d'autres résultats, les utilisateurs appartenant à des groupes ISN le trouvent plus utile que d'autres, (42% utile et 12% très utile). Pour ceux qui estiment avoir de bonnes notions de programmation mais pas de pratique, n=26, 54% estiment qu'il est utile (et très utile). Notons tout de même que les sujets inscrits en option ISN (n=46) trouvent l'outil HiPPY plus utile que ceux qui n'en font pas partie (à 33% et 13% très utile).

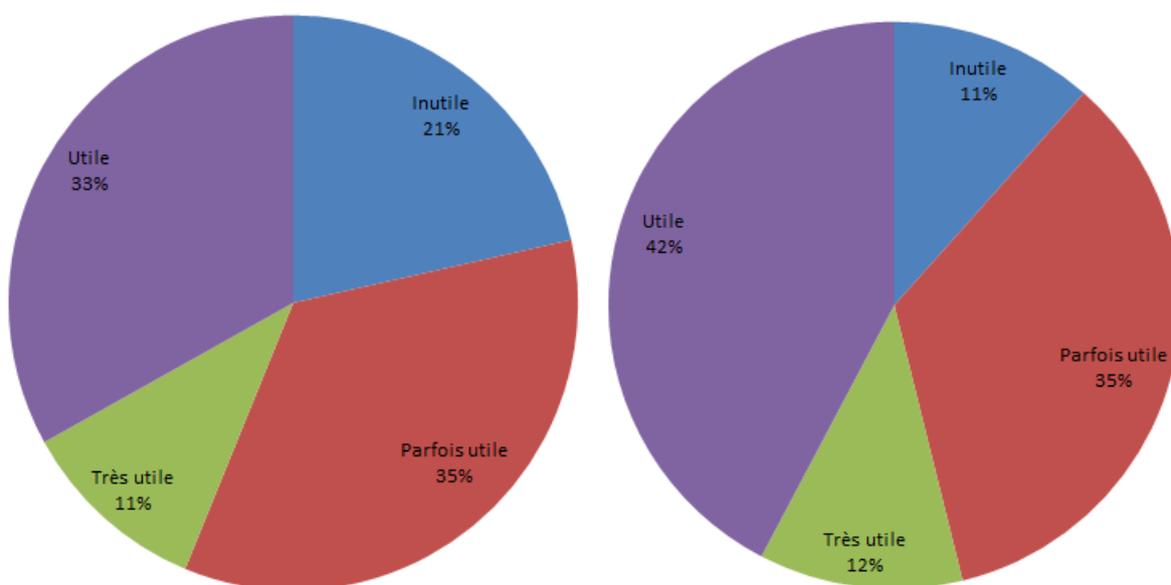


Figure 93 : évaluation de l'utilité de l'onglet « Notions associées » (à gauche n=121 : tous, à droite n=26 : bonnes notions mais pas de pratique)

Ces derniers résultats concernant l'utilité nous montrent que 44% trouvent utile l'onglet « Notions associées » avec ses différentes fonctionnalités, c'est pourquoi nous avons décidé d'investiguer plus longuement sur cet aspect et avons essayé de faire une sorte de validation croisée de ces résultats déclaratifs (Voir la section V).

L'analyse des différents résultats du questionnaire nous a clairement montré que certaines de nos hypothèses (concernant *la navigation interactive et dynamique*, *l'évaluation épistémique individualisée* et *l'utilité de l'onglet « Notions associées »*) ont été validées avec des degrés variables selon la population concernée. Nous récapitulons les résultats les plus notables :

- l'intégration de *la navigation interactive et dynamique* à base de graphes a été plutôt satisfaisante chez les utilisateurs. Même si la satisfaction générale est de **50%**, ce taux atteint **58%** chez les *utilisateurs masculins non appartenant à l'option ISN*, **59%** chez les *sujets masculins bacheliers hors ISN* et près de **64%** pour les *programmeurs non débutants* (plus de 10000 lignes de code) ;
- *l'évaluation épistémique individualisée* a bien été appréciée à hauteur de **64%** pour toute la population, à **71%** chez les sujets masculins hors ISN et à **81%** pour les utilisateurs découvrant la programmation pour la première fois et qui n'appartiennent pas à l'option ISN ;
- et *l'utilité de l'onglet « Notions associées »* était de **44%** pour tous les utilisateurs et de **54%** pour les utilisateurs déclarant avoir de bonnes notions en programmation mais pas de pratique.

Nous présentons à présent quelques éléments identifiés à partir des questions ouvertes (« de type : pourquoi trouvez-vous l'utilisation de l'onglet satisfaisant ou pas ?») sur les limites ou les problèmes constatés, les apports notoires ainsi que sur les axes d'amélioration.

e. Principaux retours d'utilisateurs sur l'onglet « Notions associées »

▪ Problèmes

Parmi les problèmes repérés, nous pouvons citer ceux inhérents à la *présentation* et au *contenu*. Sur la présentation, certains évoquaient un manque de lisibilité, une lenteur incompréhensible, un manque de clarté sur le graphe, plus de traits qu'il ne fallait. Un petit nombre le trouvait incompréhensible. Quant au contenu, quelques utilisateurs ne trouvaient pas selon eux d'information intéressante et ne s'en servaient pas pour les exercices faciles. Il y en a même ceux qui ne voient pas d'intérêt à lier les notions entre elles.

▪ Apports

Les deux apports majeurs semblent être *l'aide à la progression* et les aspects de *contenu*. Un bon nombre estime qu'il s'agit d'un bon complément pédagogique, une synthèse ou encore un résumé, un repère, une manière de montrer le parcours réalisé, la possibilité de retourner rapidement aux notions, comprendre les erreurs, donner un ordre d'apprentissage ainsi qu'un historique et permet de faire les liens. Ce qui a été apprécié aussi, c'est la manière de visualiser la complexité d'un problème.

Une vue plus détaillée (Figure 95) indique les liens qui rapprochent les mots-clés cités conjointement et peuvent renseigner sur les co-occurrences. Ainsi, la taille des nœuds et des liens donne des explications sur les liens entre les mots-clés énoncés. Ainsi, on voit par exemple une forte corrélation (épaisseur du lien) entre les mots-clés « notion » et « voir » ou encore entre « exercice » et « notion » (de moindre importance)

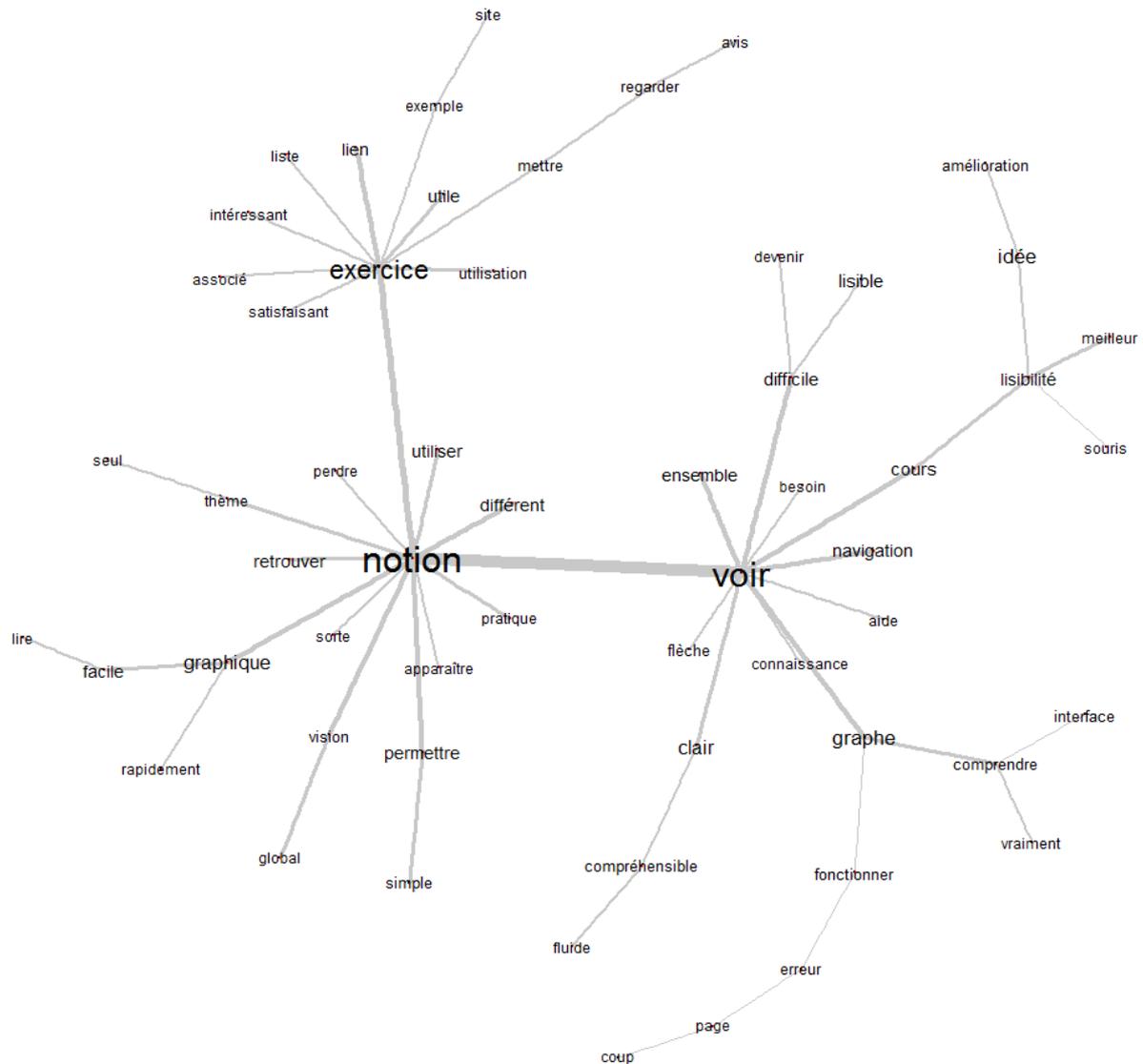


Figure 95 : graphe de dépendances des mots-clés issus des réponses ouvertes

V. Combinaison de résultats issus du questionnaire avec les données sur les traces

Suite aux résultats obtenus sur l'utilité de l'onglet "Notions associées", nous avons essayé de vérifier l'hypothèse suivante :

« Si un utilisateur a trouvé l'onglet "Notions associées" utile alors il a plus de chance d'utiliser plus fréquemment cet onglet. »

Cette hypothèse, si elle était vérifiée, pourrait être étayée par l'idée que les tâches difficiles exigent plus de rappels sur les concepts étudiés. Nous avons défini une nouvelle métrique

appelée "Taux d'Utilisation de l'Onglet Epistèmes" (TUOE), qui concerne toutes les tâches sans distinction et pour chaque utilisateur de la manière suivante :

$$TUOE = \frac{\text{Fréquence d'Usage de l'Onglet Epistèmes (FUOE)}}{\text{Fréquence d'Usage de Tous les Onglets (FUTO)}}$$

Dans la section IV du chapitre 5 de la partie II, nous avons donné les principaux moyens pour parvenir au croisement de cette métrique issue des traces avec l'indicateur déclaratif de l'utilité. Voici le graphique obtenu montrant les résultats déclaratifs (sur l'utilité de l'onglet « Notions associés ») et la fréquence effective d'utilisation (métrique calculée).

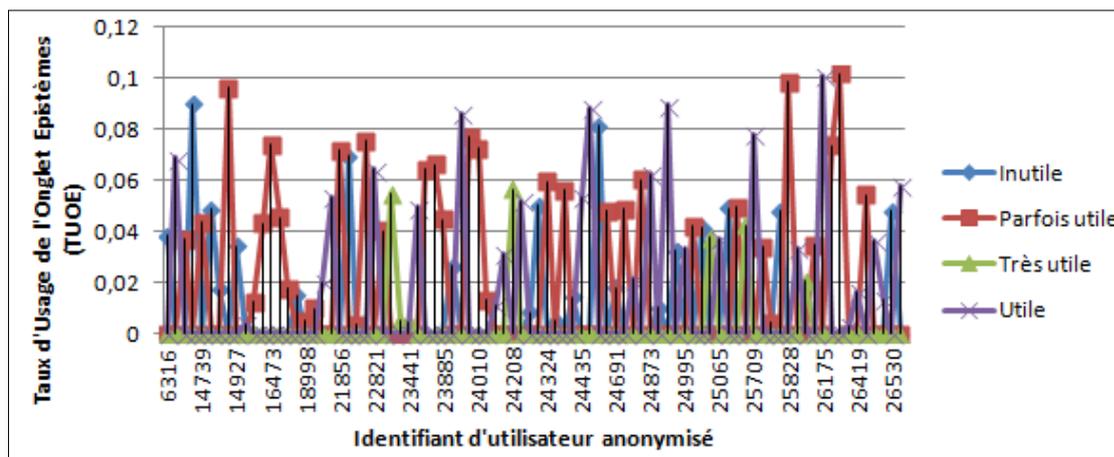


Figure 96 : relation entre l'utilité déclarée et l'usage effectif

L'axe horizontal représente les identifiants des utilisateurs anonymisés, l'axe vertical concerne les valeurs associées au TUOE. Nous nous attendions à obtenir des valeurs plus élevées de TUOE avec les marqueurs « très utile ». Ce n'était pas le cas ici. Or, ce que nous remarquons c'est que les valeurs les plus élevées sont clairement liées aux marqueurs « utile » et « parfois utile ». D'autres résultats contradictoires donnent des valeurs élevées du TUOE (0,09) pour le marqueur « inutile ». Voici les différentes valeurs (Tableau 6) de la Fréquence d'Usage de l'Onglet Epistèmes (FUOE), la Fréquence d'Usage de Tous les Onglets (FUTO) et le Taux d'Usage de l'Onglet Epistèmes (TUOE).

	FUOE	FUTO	TUOE
Moyenne	13,5	299,5	0,048028
Ecart-type	2,12132	129,400541	0,013668
Maximum	210	3218	0,101796
Minimum	1	181	0,003023

Tableau 6 : quelques chiffres sur FUOE, FUTO et TUOE

Même si ces résultats donnent des points de vue plutôt paradoxaux, il serait peut-être fructueux d'étudier plus longuement, par exemple les durées spécifiques en conjonction avec le TUOE. Cet aspect n'a pas été pris en compte ici. Nous pourrions également voir si ces

valeurs sont plus significatives avec les micro-trajectoires (c'est-à-dire jusqu'à la bonne résolution d'un exercice) éventuellement combinées avec d'autres indicateurs déclaratifs.

VI. Conclusion

Nous avons pu confirmer les premiers résultats de la pré-expérimentation lors de l'expérimentation à large-échelle. Nos résultats nous ont permis de valider l'outil intégré dans la plateforme. L'analyse des traces nous a permis de renseigner sur deux stratégies possibles constatées :

- **l'aide à la consolidation de notions** (après la correction de la tâche),
- **l'aide à la résolution** (rappels sur les épistèmes concernés)

Les mécanismes de l'individualisation de la navigation semblent bien appréciés, puisque la satisfaction globale de l'évaluation épistémique atteint 64%. Cela pourrait servir comme point commun avec un des critères d'ergonomie en Interface Homme Machine, voilà qui nous permet de joindre l'utilisabilité du point de vue IHM mais aussi du point de vue EIAH. Nous avons tout de même obtenu une indication partielle sur l'« utilité » (bien qu'il est difficile de la mesurer) qui était légèrement inférieure à 50% (44%). Il est à noter qu'il existe des rétroactions orientées sur la tâche en cours (indices sur le code, aide algorithmique sur la plateforme France-IOI, aide sur ce qu'il faut faire exactement). Cela ne paraît alors pas très surprenant. C'est sur ce dernier point que nous avons décidé de confronter traces effectives et données déclaratives pour vérifier leur cohérence et expliquer certains cheminements. Les résultats obtenus paraissent parfois contradictoires entre ce qui était déclaré et ce qui était effectif (d'après les traces).

L'ensemble de ces résultats pourrait ouvrir de nouvelles perspectives comme par exemple :

- détecter les parcours réussis (ou du moins ceux qui permettent d'avoir une bonne résolution d'exercice) : sur le plan pédagogique, ce serait une orientation utile aux enseignants et concepteurs/éditeur de l'hypermédia qui seront éclairés sur les points forts et les difficultés des apprenants ;
- détecter les parcours conduisant à un échec permet de déterminer quel profil est en corrélation avec la ou les stratégies employées selon les trajectoires utilisées ;
- en s'inspirant d'autres domaines comme le business intelligence, essayer de détecter en amont les situations d'échec pour y remédier à l'avance grâce à la détection en amont de certains motifs récurrents.

Conclusion partie III

Cette partie nous a conduit à un certain nombre de constats.

- L'utilisation d'une bibliothèque doit être testée avant de pouvoir être intégrée à un prototype qui aura pour vocation d'être manipulé par les utilisateurs. Non pas seulement pour éprouver les potentialités et les limites, mais aussi pour pouvoir détecter sa bonne adéquation aux situations à mettre en œuvre.
- Les tests de performance nous ont permis de dégager des préconisations sur le choix des méthodes de visualisation en adéquation avec des situations d'apprentissage ou plutôt avec des situations d'édition de la ressource.
- L'énonciation des hypothèses de recherche et le cadrage des éléments soumis semblent nécessaires à prendre en considération pour permettre d'installer d'une manière convenable le terrain expérimental, le recueil des données et les outils d'analyse.
- Les résultats d'une expérimentation peuvent faire appel à la collecte de traces pour pouvoir valider les fonctionnalités du prototype, analyser certains usages et vérifier l'utilisabilité ainsi que l'utilité, mais il est parfois préférable de croiser d'autres sources données comme celles issues d'un questionnaire en ligne tel que nous l'avons administré. Au-delà du constat que nous permet de faire les traces « brutes », les éléments de réponse au questionnaire doivent permettre d'éclairer ces constats par des profils récurrents ou des explications circonstanciés.

Conclusion générale

Dans le cadre d'un projet connexe visant l'élaboration d'un document numérique pour l'apprentissage du langage Python en autonomie, nous disposons d'un grand nombre (plusieurs centaines *a priori*) d'unités d'apprentissage (appelées épistèmes). L'ensemble de ces unités peut se formaliser sous la forme d'un graphe, où chaque nœud est une unité et chaque arc est une dépendance, notamment de type prérequis. Comment aider alors les apprenants à naviguer d'une manière individualisée dans cet ensemble de ressources d'apprentissage (sous forme de graphe) tout en permettant une certaine évolutivité et une adaptation à leur parcours et à leurs usages ?

L'idée de personnaliser ou d'adapter un contenu ou une présentation peut recourir, selon les degrés de complexité, à plusieurs techniques. Notre problématique centrale était de trouver un moyen pour aider à la navigation individualisée. Cela peut paraître simple à appliquer, mais la question peut se complexifier s'il s'agit de la lier à un environnement évolutif dépendant des traces collectives.

Pour cela, nous nous sommes fixés comme finalités de proposer des mécanismes d'aide à la navigation dynamique intégrés dans un prototype les mettant en œuvre, de suggérer dans une moindre mesure des processus (ou workflows) d'aide à l'édition collaborative des ressources pour (1) guider (à travers ce graphe) les apprenants vers leurs objectifs et (2) éclairer les concepteurs sur la cohérence et les révisions sur la méta-ressource.

L'approche consistant à corréler activités d'édition et de navigation n'est pas uniquement considérée comme un simple moyen, nous l'avons mise en évidence tout au long de ce travail, pour mettre à jour la ressource mais également pour mieux individualiser la navigation et les parcours. Nous pouvons parler d'artefact informatique qui doit évoluer même d'une façon minime sous l'effet de l'usage.

C'est ainsi que nous avons étudié la typologie des ressources et des parcours d'apprentissage et leur édition, la visualisation et la navigation dynamique dans les graphes, les mécanismes d'adaptation (hypermédia adaptatifs), l'exploitation ainsi que l'analyse de traces, fouille de données en éducation... Nous avons passé en revue de nombreux travaux qui ont traité des objets d'apprentissage, de leur conception et de leur édition, plus spécifiquement dans une optique d'ingénierie pédagogique. D'autres travaux se sont penchés sur l'accès et la navigation dans les ressources. Cependant, nous n'avons pas trouvé à notre connaissance, ceux qui s'intéressent à la relation entre ces activités d'édition et de navigation qui ont des objectifs différents, faisant intervenir concepteurs et apprenants. Nous avons donc proposé de mettre en relation ces deux activités.

Dans la littérature des EIAH, le recours aux mécanismes de traces a pour finalité la régulation ou l'auto-régulation de l'apprentissage, la modélisation et le profilage de l'apprenant, la personnalisation et l'adaptation de l'apprentissage ou encore la réingénierie des EIAH. Nous nous sommes proposé d'exploiter les traces issues des activités de navigation et d'utilisation (en plus des besoins d'individualisation chez l'apprenant), répondant à des besoins d'édition pour assurer, in fine, une meilleure évolutivité de la ressource dans son ensemble.

Conclusion générale

Conformément à l'architecture en couches suggérée par nos soins, nous avons adopté une méthodologie reposant sur un cycle de construction d'un hypermédia adaptatif, sur une étape d'analyse de traces par des techniques de data-mining et sur une évaluation par les utilisateurs. Cette démarche de recherche exploratoire utilise une conception et auto-évaluation des usages par les apprenants à travers l'analyse des traces collectives.

Parmi les objets d'investigation que nous avons retenus, nous pouvons rappeler, en particulier :

- les usages des apprenants utilisant le prototype (stratégies de résolution et démarches...);
- l'utilisabilité et l'utilité relative du prototype construit ;
- des principes mis en œuvre (navigation dynamique, évaluation épistémique individualisée...);
- et dans un autre volet, de moindre importance, la façon de mettre à jour le contenu/présentation à partir des usages (instanciation de la mise en relation entre édition/navigation).

Le contexte et le cadre de l'expérience consistait à proposer aux apprenants distants (certains suivent l'option ISN) l'utilisation du prototype en ligne sur la plateforme www.france-ioi.com.

Les différentes expérimentations (pré-phase et phase effective) nous ont permis de valider le mécanisme de navigation dynamique et d'évaluation épistémique individualisée. A cet égard, les utilisateurs ont globalement été satisfaits de ce système de navigation à base de visualisation interactive et ont bien apprécié le bilan de connaissances personnalisé. Nous n'avons pas pu mener à terme et expérimenter en entier ce cycle de réingénierie (couche 6 du modèle introduit), car nous nous sommes confronté à quelques freins :

- Les enseignants ou les concepteurs utilisent généralement des outils ad-hoc soit par habitude soit par préférence et il est par conséquent assez difficile de proposer un outil externe qui leur permettrait d'éditer des contenus du type épistèmes par exemple.
- Le fait de modifier ou de changer des pratiques, ne serait-ce que mineures, est parfois plus difficile que prévu ; les résistances au changement sont encore plus évidentes lorsqu'il s'agit de proposer des évolutions de type interface en termes de présentation et de contenu.

Nous estimons avoir obtenu des résultats satisfaisants sur l'intégration du prototype HiPPY du point de vue de l'acceptabilité, de l'appropriation et surtout de l'aide à la navigation individualisée.

Nous avons vu qu'une analyse de traces est parfois insuffisante pour expliquer les raisons et les choix qui poussent les utilisateurs à suivre telle ou telle démarche. De même, il est ressorti de notre recherche que l'utilisation d'un questionnaire seul n'est jamais une garantie pour évaluer l'importance de l'utilisation de certains aspects de HiPPY. Cet aspect nous a poussé à croiser ces deux types de données hétérogènes et nous a permis de relativiser certains constats.

Il est toutefois utile d'énoncer quelques limites identifiées qui feraient l'objet de perspectives de recherches futures :

- l'utilisation d'une bibliothèque préexistante peut être dans quelques situations inadaptées pour modéliser et représenter les données voulues (certains types de visualisation ne sont pas recommandés pour un graphe acyclique orienté).
- Ce prototype HiPPY n'est qu'un cas d'application au domaine de l'apprentissage de la programmation en Python. Il est, à nos yeux, tout à fait transposable à d'autres langages de programmation de paradigmes assez proches ou à d'autres domaines. Il est à noter le rôle principal de l'expert du domaine qui aurait à traiter cette tâche, pour rendre judicieux un découpage en épistèmes. Dans cette perspective, nous pouvons projeter ce type d'artefact dans des environnements de types cours en ligne ouverts et massifs (CLOM), ou encore dans un contexte de jeu sérieux pour circonscrire un domaine particulier. Il est possible, selon nous, d'appliquer certaines de nos modélisations, des méthodologies d'exploitation de traces et leur croisement avec des données déclaratives.

Il serait aussi opportun d'envisager une étape en amont qui pourrait être semi-automatisée en laissant le soin de créer, à certains moments automatiquement, un graphe des épistèmes, en partant de diverses sources du domaine et de produire des concepts « atomiques » pour un apprentissage destiné prioritairement aux débutants. Nous pensons également que ce n'est plus l'expert du domaine qui doit mettre à jour les ressources et les parcours, mais il appartiendra aux apprenants également, d'avoir la possibilité de modifier la ressource et de ne faire intervenir le tuteur qu'à des moments ponctuels et selon des besoins ciblés.

Index

A

apprenant · 10, 11, 12, 13, 14, 22, 28, 31, 39, 40, 49, 55, 56, 59, 60, 70, 78, 94, 101, 103, 106, 115, 164
Approche · 76
arbres de décision · 64, 67, 71, 158

C

CloudScorm · 14, 110, 113, 114, 130, 151
clustering · 63, 65, 66, 71, 136, 158
concepteur · 10, 11, 12, 49, 78, 92, 110
conception · 13

D

DAG · 13, 85, 86

E

édition · 13
éducation · 3
environnements d'accès aux ressources · 13, 31, 32
épistème · 13, 28, 86, 101, 102, 123, 125, 126
exercice · 28, 32, 52, 68, 72, 81, 86, 92, 95, 100, 101, 102, 125, 150
expérimentation · 13, 14, 85, 88, 140, 141, 148, 149, 151, 152, 154, 155, 157, 173, 174, 273

F

fouille de données · 3, 13, 14, 62, 63, 69, 70, 135, 139

G

graphe · 12, 14, 17, 25, 26, 27, 28, 36, 40, 43, 45, 46, 47, 48, 53, 56, 75, 78, 82, 83, 84, 85, 86, 87, 88, 89, 90, 92, 94, 96, 98, 99, 101, 106, 110, 115, 116, 117, 119, 122, 124, 125, 127, 145, 149, 150, 151, 152, 197

H

HiPPY · 12, 61, 73, 110, 122, 148, 164, 170
hypermédias · 13, 55

M

modèle · 14, 17, 19, 21, 22, 23, 24, 27, 28, 36, 50, 53, 55, 56, 57, 58, 59, 60, 64, 68, 69, 77, 79, 85, 92, 93, 94, 97, 102, 106, 108, 110, 111, 120

modélisation · 13, 14, 26, 49, 58, 59, 60, 75, 78, 82, 87, 90, 92, 103, 108
modélisations · 11, 12, 14, 117

N

navigation · 10, 11, 12, 13, 14, 15, 24, 31, 33, 42, 43, 44, 45, 47, 53, 55, 56, 57, 58, 60, 61, 69, 70, 73, 75, 76, 77, 78, 79, 80, 81, 84, 87, 90, 91, 92, 100, 101, 102, 106, 110, 115, 116, 118, 122, 124, 125, 126, 128, 145, 149, 150, 163, 169

O

objet d'apprentissage · Voir ressources d'apprenitssage

P

parcours d'apprentissage · 13
personnalisation · 13, 49, 55, 60
Preceptor · 114
programmation · 10, 11, 12, 28, 34, 39, 40, 49, 59, 95, 108, 151, 164, 167, 168, 197, 200
Python · 10, 11, 12, 13, 26, 28, 31, 34, 40, 77, 86, 87, 123, 149, 151

R

règles d'association · 63, 71, 136, 158
ressources d'apprentissage · 10, 11, 13, 19, 25, 31, 32, 44

T

traces · 10, 11, 12, 13, 14, 22, 28, 49, 52, 53, 78, 81, 92, 100, 106, 115, 116, 149, 150, 164, 173
trajectoire · 14, 100, 101, 102, 163

U

unité épistémique · Voir épistème, Voir épistème, Voir épistème

V

visualisation · 10, 11, 12, 13, 14, 39, 43, 44, 45, 78, 87, 110, 127, 140, 142, 143, 144, 151

Références bibliographiques

- Agrawal, N. (2006). Learning objects: Features and categories. Consulté à l'adresse <http://drtc.isibang.ac.in:8080/handle/1849/218>
- Ala-Mutka, K. (2012). Problems in learning and teaching programming. *Codewitz Needs Analysis*. Consulté à l'adresse http://www.cs.tut.fi/~edge/literature_study.pdf
- Aleven, V., McLaren, B., Roll, I., & Koedinger, K. (2004). Toward tutoring help seeking. In *Intelligent Tutoring Systems* (p. 227-239). Springer.
- Amershi, S., & Conati, C. (2009). Combining unsupervised and supervised classification to build user models for exploratory learning environments. *Journal of Educational Data Mining, 1*(1), 18-71.
- Auber, D., Chiricota, Y., Delest, M., Melançon, G., Domenger, J.-P., Mary, P., & others. (2007). Visualisation de graphes avec Tulip: exploration interactive de grandes masses de données en appui à la fouille de données et à l'extraction de connaissances (p. 147–156). Consulté à l'adresse <http://hal.archives-ouvertes.fr/lirmm-00203728/>
- Baker, R. S., Corbett, A. T., Koedinger, K. R., & Wagner, A. Z. (2004). Off-task behavior in the cognitive tutor classroom: when students game the system. In *Proceedings of the SIGCHI conference on Human factors in computing systems* (p. 383-390). ACM.
- Baker, R. S., Gowda, S. M., Wixon, M., Kalka, J., Wagner, A. Z., Salvi, A., ... Rossi, L. (2012). Towards Sensor-Free Affect Detection in Cognitive Tutor Algebra. In *Proceedings of the 5th International Conference on Educational Data Mining* (p. 126-133).
- Baker, R., Walonoski, J., Heffernan, N., Roll, I., Corbett, A., & Koedinger, K. (2008). Why students engage in « gaming the system » behavior in interactive learning environments. *Journal of Interactive Learning Research, 19*(2), 185-224.

Références bibliographiques

- Baker, R., & Yacef, K. (2009a). The state of educational data mining in 2009: A review and future visions. *Journal of Educational Data Mining*, 1(1), 3-17.
- Balacheff, N. (1994). Didactique et intelligence artificielle. *Recherches en didactique des mathématiques*, 14, 9-42.
- Baron, G.-L., & Bruillard, E. (2011). L'informatique et son enseignement dans l'enseignement scolaire général français. Enjeux de pouvoir et de savoirs. In *Recherches et expertises pour l'enseignement scientifique, Technologie, sciences, mathématiques* (J. Lebeaume, A. Hasni, & I. Harlé (Éd.), p. 79-90). De Boeck.
- Barritt, C., & Alderman Jr, F. L. (2004). *Creating a reusable learning objects strategy: Leveraging information and learning in a knowledge economy*. Wiley. com.
- Bouhineau, D., Luengo, V., Mandran, N., Ortega, M., & Wajeman, C. (s. d.). Conception et mise en place d'un entrepôt de traces et processus de traitement EIAH: UnderTracks. In *6e Conférence sur les Environnements Informatiques pour l'Apprentissage Humain* (p. 41). Consulté à l'adresse <http://www.irit.fr/EIAH2013/uploads/LivretPosterDemosEIAH2013.pdf#page=47>
- Brézillon, P. (2003). Représentation de pratiques dans le formalisme des graphes contextuels. *Actes des Deuxièmes Journées d'Etude en Psychologie Ergonomique–EPIQUE*, 263-268.
- Broisin, J. (2005). A Model driven Approach for the Management of Learning Object.
- Broisin, J., & Vidal, P. (2005). Un Environnement Informatique pour l'Apprentissage Humain au service de la Virtualisation des Objets Pédagogiques. *Revue STICEF*, 12, 177-204.
- Brusilovsky, P. (1996). Methods and techniques of adaptive hypermedia. *User modeling and user-adapted interaction*, 6(2-3), 87-129.
- Brusilovsky, P. (1997). Efficient Techniques for Adaptive Hypermedia (p. 87–129). Kluwer Academic Publishers.

- Brusilovsky, P. (1999). Adaptive and intelligent technologies for web-based education. *KI*, 13(4), 19-25.
- Brusilovsky, P. (2001). Adaptive hypermedia. *User modeling and user-adapted interaction*, 11(1-2), 87-110.
- Brusilovsky, P., & Sosnovsky, S. (2005). Individualized exercises for self-assessment of programming knowledge: An evaluation of QuizPACK. *Journal on Educational Resources in Computing (JERIC)*, 5(3), 6.
- Campbell, L. M. (2003a). Engaging with the learning object economy. In A. Littlejohn (Éd.), *Reusing online resources: a sustainable approach to e-learning* (p. 35–45). London: Kogan Page.
- Catteau, O. (2008a). *Le cycle de vie de l'objet pédagogique et de ses métadonnées*. Université de Toulouse. Consulté à l'adresse <http://hal.archives-ouvertes.fr/tel-00344701/>
- Catteau, O., Marquié, D., Vidal, P., & Broisin, J. (2009a). Une architecture orientée services pour la gestion des ressources d'apprentissage International E-MI@ GE. Consulté à l'adresse <http://hal.archives-ouvertes.fr/hal-00423815/>
- Chaabouni, M., Miled, M., & Laroussi, M. (2013). Un aperçu sur les traces dans les EIAH : utilisations, évolutions technologiques et défis. Présenté à CNRTICE 13 - 1ère Conférence Nationale sur la Recherche en TICE, Cité des sciences Tunis: Institut de la Francophonie pour l'ingénierie de la connaissance et la formation à distance.
- Cheng, R., & Vassileva, J. (2005). Adaptive Reward Mechanism for Sustainable Online Learning Community. In *AIED* (p. 152-159).
- Cheniti-Belcadhi, L., Braham, R., Henze, N., & Nejdil, W. (2004). A Generic Framework for Assessment in Adaptive Educational Hypermedia. In *ICWI* (p. 397-404).

Références bibliographiques

- Cheniti-Belcadhi, L., Henze, N., & Braham, R. (2008). Assessment personalization in the Semantic Web. *Journal of Computational Methods in Science and Engineering*, 8(3), 163-182.
- Cheung, R., Wan, C., & Cheng, C. (2010). *An ontology-based framework for personalized adaptive learning*. Springer.
- Choquet, C., & Iksal, S. (2007a). Modeling Tracks for the Model Driven Re-engineering of a TEL System. *Journal of Interactive Learning Research*, 18(2), 161-184.
- Clauzel, D., Sehaba, K., & Prié, Y. (2009). Modelling and visualising traces for reflexivity in synchronous collaborative systems. In *International Conference on Intelligent Networking and Collaborative Systems (INCoS 2009)* (p. 16–23). IEEE Computer Society. doi:10.1109/INCOS.2009.55
- Cram, D., Jouvin, D., & Mille, A. (2007). Visualisation interactive de traces et réflexivité: application à l'EIAH collaboratif synchrone eMédiathèque. *Revue Sticef. org*, 21, 07.
- De Bra, P., Aerts, A., Berden, B., De Lange, B., Rousseau, B., Santic, T., ... Stash, N. (2003). AHA! The adaptive hypermedia architecture. In *Proceedings of the fourteenth ACM conference on Hypertext and hypermedia* (p. 81-84). ACM.
- De La Passardière, B., & Grandbastien, M. (2003). Présentation de LOM v1. 0, standard IEEE. *Revue Sciences et techniques éducatives*, 211–218.
- Delozanne, E., Le Calvez, F., Merceron, A., & Labat, J.-M. (2007). Design Patterns en EIAH: vers un langage de Patterns pour l'évaluation des apprenants. *Revue des Sciences et Technologies de l'Information et de la Communication pour l'Education et la Formation (STICEF)*, volume 14.
- DePaula, R., Fischer, G., & Ostwald, J. (2001). Courses as seeds: Expectations and realities. In *Proceedings of the Second European Conference on Computer-Supported Collaborative Learning (Euro-CSCL '2001)* (p. 494-501).

- Dernoncourt, F., Taylor, C., O'Reilly, U.-M., Veeramachaneni, K., Wu, S., Do, C., & Halawa, S. (2013). MoocViz: A Large Scale, Open Access, Collaborative, Data Analytics Platform for MOOCs.
- Després, C., & Jacoboni, P. (2010). *Hop3x : Un Environnement de Suivi de TP de Programmation*. Université du Maine.
- Després, C., S., & Jacoboni, P. (2010). *Calcul d'indicateurs pour HOP3X avec UTL*. Consulté à l'adresse http://liris.cnrs.fr/rjc_eiah_2010/doku.php?id=Journ%C3%A9e%20GDR%20I3
- Dimitracopoulou, A. (2004). State of the art on Interaction and Collaboration Analysis (D26.1.1). *EU Sixth Framework programme priority, 2*.
- Djouad, T., Mille, A., Reffay, C., & Benmohamed, M. (2009). Ingénierie des indicateurs d'activités à partir de traces modélisées pour un Environnement Informatique d'Apprentissage Humain. *Revue des Sciences et Technologies de l'Information et de la Communication pour l'Education et la Formation (STICEF)*, 16.
- Djouad, T., Mille, A., Reffay, C., & Benmohammed, M. (2010). A New Approach Based on Modelled Traces to Compute Collaborative and Individual Indicators Human Interaction. In *Advanced Learning Technologies (ICALT), 2010 IEEE 10th International Conference on* (p. 53-54). IEEE.
- Djouad, T., Settouti, L. S., Prié, Y., Reffay, C., & Mille, A. (2010). Un Système à Base de Traces pour la modélisation et l'élaboration d'indicateurs d'activités éducatives individuelles et collectives. Mise à l'épreuve sur Moodle. *Mise à l'épreuve sur Moodle. Technique et Science Informatiques*.
- Dodds, P. (2001). Advanced distributed learning sharable content object reference model version 1.2. *The SCORM Content Aggregation Model*.

Références bibliographiques

- Ernst, C. J. (2008). *E-learning: conception et mise en oeuvre d'un enseignement en ligne : guide pratique pour une e-pédagogie*. Toulouse: Cepaduès.
- Falquet, G., Guyot, J., & Hurni, J.-P. (2003). Modèles et interfaces d'hyperlivres pour l'apprentissage collaboratif. In *E. Bruillard & B. de La Passardière, Ressources numériques, XML et éducation. Sciences et techniques éducatives, Hors série 2003* (p. 19–43).
- George, S. (2010). *Interactions et communications contextuelles dans les environnements informatiques pour l'apprentissage humain*. HDR. INSA de Lyon.
- Gerard, R. W. (1967). Shaping the mind: Computers in education. In *N. A. Sciences, Applied Science and Technological Progress*.
- Guéraud, V., Adam, J.-M., Pernin, J.-P., Calvary, G., & David, J.-P. (2004). L'exploitation d'Objets Pédagogiques Interactifs à distance : le projet FORMID. *Revue des Sciences et Technologies de l'Information et de la Communication pour l'Education et la Formation (STICEF)*, 11, 46 pages.
- Hadzilacos, T., Kalles, D., Karaiskakis, D., & Pouliopoulou, M. (2009). Using Graphs in Developing Educational Material. In *Proceedings of the 2nd International Workshop on Building Technology Enhanced Learning solutions for Communities of Practice* (p. 55-66). Citeseer.
- Hamel, M.-J. (2013). Analyse de l'activité de recherche d'apprenants de langue dans un prototype de dictionnaire en ligne. *Alsic. Apprentissage des Langues et Systèmes d'Information et de Communication*.
- Han, F., Veeramachaneni, K., & O'Reilly, U.-M. (s. d.). Analyzing Millions of Submissions to Help MOOC Instructors Understand Problem Solving. Consulté à l'adresse <http://groups.csail.mit.edu/EVO-DesignOpt/groupWebSite/uploads/Site/ProblemAnalytics.pdf>

- Heiwy, V., & Ducateau, C.-F. (2003). Un modèle de ressources pédagogiques pour la foaid: modélisation de ressources pédagogiques. In *Environnements Informatiques pour l'Apprentissage Humain 2003* (p. 235-246).
- Henze, N., & Nejdil, W. (2003). Logically characterizing adaptive educational hypermedia systems. In *International Workshop on Adaptive Hypermedia and Adaptive Web-based Systems (AH 2003)*.
- Heraud, J.-M., Marty, J.-C., France, L., & Carron, T. (2005). Helping the interpretation of web logs: Application to Learning Scenario Improvement. In *Proceedings of« Usage Analysis in Learning Systems » workshop* (p. 41-48). Citeseer.
- Herman, I., Melançon, G., & Marshall, M. S. (2000). Graph visualization and navigation in information visualization: A survey. *Visualization and Computer Graphics, IEEE Transactions on*, 6(1), 24-43.
- Hernandez, N., Mothe, J., Ramamonjisoa, A. B. O., Ralalason, B., & Stolf, P. (2009). Indexation multi-facettes des ressources pédagogiques pour faciliter leur ré-utilisation. *Institut de Recherche en Informatique de Toulouse, avabile on-line at ftp://ftp.irit.fr/IRIT/SIG/2008_RNTI_HMRRS. pdf, 10*. Consulté à l'adresse ftp://141.115.28.2/pub/pub/IRIT/SIG/2008_RNTI_HMRRS.pdf
- Heutte, J. (2011). *La part du collectif dans la motivation et son impact sur le bien-être comme médiateur de la réussite des étudiants: Complémentarités et contributions entre l'autodétermination, l'auto-efficacité et l'autotélisme*. Thèse de doctorat en Sciences de l'Éducation.
- Hockemeyer, C., Held, T., & Albert, D. (1998). RATH-a relational adaptive tutoring hypertext WWW-environment based on knowledge space theory. In C. Alvegard (Éd.), *Computer Aided Learning and Instruction in Science and Engineering, Proceedings of the CALISCE 98* (p. 417-423). Göteborg, Sweden: Chalmers

Références bibliographiques

- University of Technology. Consulté à l'adresse <http://css.uni-graz.at/rath/publications/calisce.pdf>
- Hodgins, H. W. (2000). Into the future a vision paper. In *Commission on Technology and Adult Learning*. Retrieved from <http://www.learnativity.com/download/MP7.PDF> on 6/13/2007. Citeseer.
- Hsiao, I.-H., Brusilovsky, P., & Sosnovsky, S. (2008). Web-based parameterized questions for object-oriented programming. In *World Conference on E-Learning in Corporate, Government, Healthcare, and Higher Education* (Vol. 2008, p. 3728-3735).
- ISKME. (2008). *Creating, Doing, and Sustaining OER: Lessons from Six Open Educational Resource Projects*. The Institute for the Study of Knowledge Management in Education. Consulté à l'adresse <http://www.iskme.org/our-work/creating-doing-and-sustaining-oer-lessons-six-open-educational-resource-projects>
- Jean-Daubias, S., Ginon, B., & Lefevre, M. (2012). Modèles et outils pour prendre en compte l'évolutivité dans les profils d'apprenants. *Revue Sticef.org*, 16-26.
- Jih, H. J. (1996). The impact of learners' pathways on learning performance in multimedia computer aided learning. *Journal of network and computer applications*, 19(4), 367-380.
- Ji, M., Michel, C., Lavoué, E., & George, S. (2013). An Architecture to Combine Activity Traces and Reporting Traces to Support Self-Regulation Processes. In *Advanced Learning Technologies (ICALT), 2013 IEEE 13th International Conference on* (p. 87-91). IEEE.
- Joab, M., Guin, D., & Trouche, L. (2003). Conception et réalisation de ressources pédagogiques vivantes : des ressources intégrant les TICE en mathématiques. In D. Desmoulins, C., Marquet, P., Bouhineau (Éd.), (p. 259-270). Strasbourg, France: INRP. Consulté à l'adresse <http://telearn.archives-ouvertes.fr/lirmm-00138490/en/>

- Klašnja-Milićević, A., Vesin, B., Ivanović, M., & Budimac, Z. (2011). Integration of recommendations and adaptive hypermedia into Java tutoring system. *Computer Science and Information Systems*, 8(1), 211-224.
- Klerkx, J., Duval, E., & Meire, M. (2004). Using information visualization for accessing learning object repositories (p. 465–470).
- Klerkx, J., Meire, M., Ternier, S., Verbert, K., & Duval, E. (2005). Information visualisation: towards an extensible framework for accessing learning object repositories (p. 4281–4287).
- Klerkx, J., Vandeputte, B., Parra, G., Santos, J. L., Van Assche, F., & Duval, E. (2010a). How to share and reuse learning resources: the ARIADNE experience. In *Sustaining TEL: From Innovation to Learning and Practice* (p. 183–196). Springer. Consulté à l'adresse http://link.springer.com/chapter/10.1007/978-3-642-16020-2_13
- Kobsa, A., Müller, D., & Nill, A. (1998). KN-AHS: An adaptive hypertext client of the user modeling system BGP-MS. *Readings in intelligent user interfaces*, 372-380.
- Koedinger, K. R., Baker, Rsj., Cunningham, K., Skogsholm, A., Leber, B., & Stamper, J. (2010). A data repository for the EDM community: The PSLC DataShop. *Handbook of educational data mining*, 43–55.
- Kölling, M., & Rosenberg, J. (1996). An object-oriented program development environment for the first programming course. In *ACM SIGCSE Bulletin* (Vol. 28, p. 83-87). ACM.
- Kop, R., & Carroll, F. (2011). Cloud computing and creativity: learning on a massive open online course. *European Journal of Open, Distance and E-learning. Special Issue on Creativity and OER*.
- Kumar, A. (2005). Usage Analysis in Tutors for C++ Programming. In *Proceedings of« Usage Analysis in Learning Systems » workshop* (p. 57-64).

Références bibliographiques

- Labat, J.-M., Pastré, P., Parage, P., Fattersack, M., Richard, J.-F., & Sander, E. (2007). Analyser les stratégies de résolution de problèmes en situation naturelle grâce à un simulateur: le cas des régleurs de plasturgie. In *Actes de la conférence EIAH 2007*.
- Littlejohn, A. (2003). *Reusing online resources: a sustainable approach to e-learning*. London ; Sterling, VA: Kogan Page.
- Livieris, I. E., Drakopoulou, K., & Pintelas, P. (2012). Predicting students' performance using artificial neural networks. In *Proceedings of the 8th Pan-Hellenic Conference "Information and Communication Technology in Education*. Consulté à l'adresse <http://hcicte2012.uth.gr/main/sites/default/files/proc/Proceedings/LivierisEtAl.pdf>
- Lonchamp, J. (2008). Un cadre conceptuel et logiciel pour la construction d'environnements d'apprentissage collaboratifs. *Sciences et Technologies de l'Information et de la Communication pour l'Éducation et la Formation (STICEF)*, volume 14, 2007.
- Louca, J., Johnson, J., Bourguine, P., Portelli, P., Tijus, C., SCIUS-BERTRAND, A., ... Collet, P. (2013). Poem Platform For Massive Personalized Education. Présenté à ECCS'13, Barcelona, Spain. Consulté à l'adresse <http://icube-publis.unistra.fr/7-LJBP13>
- Luengo, V., Vadcard, L., Tonetti, J., & Dubois, M. (2011). Diagnostic des connaissances et rétroaction épistémique adaptative en chirurgie. *Revue d'intelligence artificielle*, 25(4), 499-524.
- Maloney, J., Burd, L., Kafai, Y., Rusk, N., Silverman, B., & Resnick, M. (2004). Scratch: a sneak preview [education]. In *Creating, Connecting and Collaborating through Computing, 2004. Proceedings. Second International Conference on* (p. 104-109). IEEE.
- McGreal, R. (2008). A Typology of Learning Object Repositories. In H. H. Adelsberger, Kinshuk, J. M. Pawlowski, & D. G. Sampson (Éd.), *Handbook on Information*

Technologies for Education and Training (p. 5-28). Springer Berlin Heidelberg.

Consulté à l'adresse http://dx.doi.org/10.1007/978-3-540-74155-8_1

Merceron, A., & Yacef, K. (2004). Mining student data captured from a web-based tutoring tool: Initial exploration and results. *Journal of Interactive Learning Research*, 15(4), 319-346.

Merceron, A., & Yacef, K. (2005). Tada-ed for educational data mining. *Interactive Multimedia Electronic Journal of Computer-Enhanced Learning*, 7(1), 267-287.

Merceron, A., & Yacef, K. (2008). Interestingness Measures for Associations Rules in Educational Data. *EDM*, 8, 57-66.

Merzeau, L. (2009). Du signe à la trace: l'information sur mesure. *Hermès*, (53), 23-29.

Merzeau, L. (2010). L'intelligence de l'utilisateur. *L'utilisateur numérique*, 9-37.

Michaux, J., Blanc, X., Shapiro, M., & Sutra, P. (2011). A semantically rich approach for collaborative model edition (p. 1470-1475). ACM.

Miled, M. (2011). Les ressources complexes pour l'apprentissage : comment exploiter les traces d'utilisation et soutenir l'édition collaborative ? Mons, Belgique. Consulté à l'adresse <http://hal.archives-ouvertes.fr/hal-00620866/en/>

Miled, M. (2012). Vers une mise en relation des activités d'édition et de navigation dans les ressources d'apprentissage : cas de l'apprentissage d'un langage de programmation (p. 75-80). Amiens, France. Consulté à l'adresse <http://hal.archives-ouvertes.fr/hal-00705889>

Miled, M., Reffay, C., & Février, L. (2013). Comment identifier les stratégies de résolution dans un hypermédia épistémique sur l'apprentissage du langage Python ? In *Actes de la conférence : Environnements Informatiques pour l'Apprentissage Humain* (p. 29-40). Toulouse, France.

Références bibliographiques

- Miled, M., Reffay, C., & Laroussi, M. (2014). An early evaluation of the HiPPY tool usage: the France-IOI case study. In *Local Proceedings of the 7th International Conference on Informatics in Schools: Situation, Evolution and Perspectives*.
- Mille, A. (2006). From case-based reasoning to traces-based reasoning. *Annual Reviews in Control*, 30(2), 223-232.
- Nesbit, J. C., Belfer, K., & Leacock, T. (2003). Learning object review instrument (LORI). *E-learning research and assessment network*.
- Ng'ambi, D., & Brown, I. (2009). Intended and unintended consequences of student use of an online questioning environment. *British Journal of Educational Technology*, 40(2), 316-328.
- Nguyen, T. H. T., Bruillard, É. (2011). Partage de ressources pédagogiques numériques entre enseignants: une enquête au Vietnam. In *Actes de la conférence EIAH 2011* (p. 125–137). Consulté à l'adresse <http://halshs.archives-ouvertes.fr/hal-00605772/>
- Nodenot, T. (2005). *Contribution à l'Ingénierie dirigée par les modèles en EIAH: le cas des situations-problèmes coopératives*. Université de Pau et des Pays de l'Adour.
- Nogry, S., Jean-Daubias, S., Ollagnier-Beldame, M., & others. (2004). Évaluation des EIAH: une nécessaire diversité des méthodes (p. 265–271).
- Ortiz, P. (2012a). Hypermédia adaptatif à épistèmes pour l'apprentissage des langages de programmation. *RJC EIAH'2012*, 99.
- Ouraiba, E.-A., Choquet, C., Cottier, P., Despres, C., & Jacoboni, P. (2010). Engineering of Open Learning Scenarios-The Case of Hop3x Learning Scenarios (p. 264-268). Présenté à Advanced Learning Technologies (ICALT), 2010 IEEE 10th International Conference on, IEEE.
- Paquette, G. (2000). Construction de portails de télé-apprentissage Explor@-Une diversité de modèles pédagogiques. *science et Techniques Educatives*, 7(1), 207-226.

- Pernin, J.-P. (2005). CSE, un modèle de traitement de traces. *Rapport interne de recherche CLIPS-IMAG*.
- Prié, A. M.-Y. (2006). Une théorie de la trace informatique pour faciliter l'adaptation dans la confrontation logique d'utilisation/logique de conception.
- Reffay, C., Miled, M., Ortiz, P., & Février, L. (2013). An Epistemic Hypermedia to Learn Python as a Resource for an Introductory Course for Algorithmics in France. In *Informatics in Schools: Local Proceedings of the 6th International Conference ISSEP 2013—Selected Papers* (p. 111). Consulté à l'adresse <http://opus.kobv.de/ubp/volltexte/2013/6368/pdf/cid06.pdf#page=113>
- Rodrigo, M. M. T., Baker, R. S., McLaren, B., Jayme, A., & Dy, T. (2012). Development of a Workbench to Address the Educational Data Mining Bottleneck. (p. 152–155). Consulté à l'adresse http://users.wpi.edu/~rsbaker/edm2012_submission_8.pdf
- Romero, C., & Ventura, S. (2010a). Educational data mining: a review of the state of the art. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*. 40(6), 601–618.
- Rosenbaum, S., & Williams, J. (2004). *Learning paths: Increase profits by reducing the time it takes employees to get up-to-speed*. John Wiley & Sons.
- Sanrach, C., & Grandbastien, M. (2000). ECSAIWeb: A Web-Based Authoring System to Create Adaptive Learning Systems. In P. Brusilovsky, O. Stock, & C. Strapparava (Éd.), *Adaptive Hypermedia and Adaptive Web-Based Systems* (Vol. 1892, p. 214–226). Springer Berlin Heidelberg. Consulté à l'adresse http://dx.doi.org/10.1007/3-540-44595-1_20
- Seifi, M., Guillaume, J., Latapy, M., & Le Grand, B. (2012). Visualisation interactive multi-échelle des grands graphes. Application à un réseau de blogs. *Revue d'Intelligence Artificielle*, 26(4), 351–368.

Séminaire SDTICE 2007 — Enseigner avec le numérique — Éduscol numérique. (s. d.).

Consulté 14 mai 2014, à l'adresse

<http://eduscol.education.fr/numerique/dossier/archives/metadata/seminaire-sdtice-2007>

Shaffer, D. W., Hatfield, D., Svarovsky, G. N., Nash, P., Nulty, A., Bagley, E., ... Mislevy, R.

(2009). Epistemic Network Analysis: A Prototype for 21st-Century Assessment of Learning. *International Journal of Learning and Media*, 1(2), 33-53.

doi:10.1162/ijlm.2009.0013

Siemens, G., & Long, P. (2011). Penetrating the fog: Analytics in learning and education.

Educause Review, 46(5), 30-32.

Soller, A., Martínez, A., Jermann, P., & Muehlenbrock, M. (2005). From mirroring to

guiding: A review of state of the art technology for supporting collaborative learning.

International Journal of Artificial Intelligence in Education, 15(4), 261-290.

Sosteric, M., & Hesemeier, S. (2002). When is a Learning Object not an Object: A first step

towards a theory of learning objects. *The International Review of Research in Open and Distance Learning*, 3(2).

Tam, J., & Greenberg, S. (2006a). A framework for asynchronous change awareness in

collaborative documents and workspaces. *International Journal of Human-Computer*

Studies, 64(7), 583-598. doi:10.1016/j.ijhcs.2006.02.004

Vachet, C. (2006). Apports de l'Ingénierie Dirigée par les Modèles au Domaine des EIAH.

Présenté à RJC EIAH 2006. Consulté à l'adresse [http://events.it-](http://events.it-sudparis.eu/rjc_eiah2006/proceeding/Article8.pdf)

[sudparis.eu/rjc_eiah2006/proceeding/Article8.pdf](http://events.it-sudparis.eu/rjc_eiah2006/proceeding/Article8.pdf)

Vaucelle, A. (2009). MPEG-21: la norme des TICE du XXIème siècle? In *Conférence*

H2PTM'09, Hypertext & hypermedia, Products, Tools et Methods: Retrospective &

- Perspective 1989–2009, du 30 septembre au 2 octobre 2009* (Vol. 1). Consulté à l'adresse <http://hal.archives-ouvertes.fr/hal-00464617/>
- Veeramachaneni, K., Dernoncourt, F., Taylor, C., Pardos, Z., & O'Reilly, U.-M. (2013). Moocdb: Developing data standards for mooc data science. In *AIED 2013 Workshops Proceedings Volume* (p. 17).
- Verbert, K., & Duval, E. (2004). Towards a Global Component Architecture for Learning Objects: A Comparative Analysis of Learning Object Content Models (p. 202–208).
- Wagner, E. (2002). Steps to creating a content strategy for your organization. *Best of The*, 103.
- Warshall, S. (1962). A theorem on boolean matrices. *Journal of the ACM (JACM)*, 9(1), 11-12.
- Wiley, D. A. (2000). Connecting learning objects to instructional design theory: A definition, a metaphor, and a taxonomy. *The Instructional Use of Learning Objects: Online Version*. Consulté à l'adresse <http://reusability.org/read/chapters/wiley.doc>
- Wiley, D., & Gurrell, S. (2009). A decade of development... *Open Learning: The Journal of Open and Distance Learning*, 24(1), 11-21. doi:10.1080/02680510802627746
- Wolpers, M., Memmel, M., & Giretti, A. (2009). Metadata in architecture education—first evaluation results of the mace system. In *Learning in the synergy of multiple disciplines* (p. 112–126). Springer. Consulté à l'adresse http://link.springer.com/chapter/10.1007/978-3-642-04636-0_13
- Wu, X., & Kumar, V. (2010). *The top ten algorithms in data mining*. CRC Press.

Annexes

I.	MODÈLE DE DONNÉES POUR LE GRAPHE DES ÉPISTÈMES SOUS DHOSTON SPRINGY	194
II.	MODÈLE DE DONNÉES SOUS JSON (ADAPTÉ AU FRAMEWORK JIT).....	200
III.	FICHIER IMSMANIFEST.XML (ARCHIVE SCORM).....	218
IV.	QUELQUES FICHIERS ZEND FRAMEWORK (MODELS, VIEWS ET CONTROLLERS).....	224
1.	<i>Configuration</i>	224
2.	<i>Organisation d'une application</i>	225
3.	<i>Quelques fichiers de l'application PRECEPTOR</i>	225
a.	Le contrôleur Ressources.....	225
b.	Exemple de vue index.phtml (lié au contrôleur Ressources).....	237
c.	Fichier ForceDirectedTestGrComple.js (sans le modèle de données du graphe des épistèmes)	240
V.	MODÈLE DE TRAJECTOIRES EN RDF	255
VI.	FICHIERS PHP DU PROTOTYPE HIPPY	263
1.	<i>Extraits du Fichier Episteme.inc.php</i>	263
a.	La fonction d'individualisation	263
b.	La fonction de construction de la fermeture transitive	264
VII.	EXEMPLE DE PROCESSUS D'ANALYSE EN XML (GÉNÉRÉ PAR RAPIDMINER)	265
1.	<i>Weka Hierarchical Clustering</i>	265
VIII.	QUESTIONNAIRE	267

I. Modèle de données pour le graphe des épistèmes sous Dhoston Springy

```
var graph = new Graph();

var GenPy = graph.newNode({label: "Généralités sur Python"});
var CodePy = graph.newNode({label: "Code Python en action"});
var ProcPy = graph.newNode({label: "Le processus de programmation en python"});
var ModeEdit = graph.newNode({label: "Les deux modes d'édition"});
var SessInter = graph.newNode({label: "La session interactive"});
var ModeFic = graph.newNode({label: "Le mode fichier"});

var LetsGo = graph.newNode({label: "Let's go"});
var DecouvrPy = graph.newNode({label: "Découvrir Python"});
var BibStand = graph.newNode({label: "La bibliothèque standard"});
var PlaceComment = graph.newNode({label: "Placer des commentaires"});
var CaracASCII = graph.newNode({label: "Les caractères ASCII"});
var ChainPres = graph.newNode({label: "Les chaînes : présentation"});
var TypesBase = graph.newNode({label: "Les types de bases"});

var AccessModStand = graph.newNode({label: "Accéder à un module standard"});
var Encode = graph.newNode({label: "L'encodage"});
var LesVar = graph.newNode({label: "Les variables"});
var LesBool = graph.newNode({label: "Les booléens"});
var OpNumb = graph.newNode({label: "Les opérations sur les nombres"});
var AutrType = graph.newNode({label: "Les autres types"});

var HasardPy = graph.newNode({label: "Le hasard en Python"});
var Affich = graph.newNode({label: "Afficher"});
var NomVar = graph.newNode({label: "Les noms de variables"});
var InstrIf = graph.newNode({label: "L'instruction if"});
var TypeDonnPy = graph.newNode({label: "Les types de données en Python"});
var OpFlot = graph.newNode({label: "Les opérations sur les flottants"});
var OpEnt = graph.newNode({label: "Les opérations sur les entiers"});

var FoncRand = graph.newNode({label: "La fonction randrange"});
var InstrIfElif = graph.newNode({label: "L'instruction if/elif"});
var InstrIfElse = graph.newNode({label: "L'instruction if/else"});
var BoucleWhile = graph.newNode({label: "La boucle while"});
var OpVar = graph.newNode({label: "Les opérations avec des variables"});

var InstrIfElifElse = graph.newNode({label: "L'instruction if/elif/else"});
var EntreInput = graph.newNode({label: "Les entrées via la fonction input"});
```

```

var UtilCompt = graph.newNode({label: "L'utilisation d'un compteur"});

graph.newEdge(GenPy, ProcPy, {color: '#00A0B0'});
graph.newEdge(CodePy, ProcPy, {color: '#6A4A3C'});
graph.newEdge(ProcPy, ModeEdit, {color: '#CC333F'});
graph.newEdge(ModeEdit, SessInter, {color: '#EB6841'});
graph.newEdge(ModeEdit, ModeFic, {color: '#EDC951'});
graph.newEdge(SessInter, LetsGo, {color: '#7DBE3C'});
graph.newEdge(ModeFic, LetsGo, {color: '#000000'});

graph.newEdge(LetsGo, DecouvrPy, {color: '#00A0B0'});

graph.newEdge(DecouvrPy, BibStand, {color: '#6A4A3C'});
graph.newEdge(DecouvrPy, PlaceComment, {color: '#CC333F'});
graph.newEdge(DecouvrPy, ChainPres, {color: '#EB6841'});
graph.newEdge(DecouvrPy, TypesBase, {color: '#EDC951'});

graph.newEdge(BibStand, AccessModStand, {color: '#7DBE3C'});
graph.newEdge(PlaceComment, Encode, {color: '#000000'});
graph.newEdge(CaracASCII, Encode, {color: '#00A0B0'});

graph.newEdge(AccessModStand, HasardPy, {color: '#6A4A3C'});
graph.newEdge(HasardPy, FoncRand, {color: '#CC333F'});
graph.newEdge(ChainPres, Affich, {color: '#EB6841'});
graph.newEdge(TypesBase, LesVar, {color: '#EDC951'});

graph.newEdge(TypesBase, LesBool, {color: '#7DBE3C'});
graph.newEdge(TypesBase, OpNumb, {color: '#000000'});

graph.newEdge(TypesBase, AutrType, {color: '#00A0B0'});

graph.newEdge(TypesBase, TypeDonnPy, {color: '#6A4A3C'});
graph.newEdge(TypesBase, OpFlot, {color: '#CC333F'});

graph.newEdge(LesVar, Affich, {color: '#EB6841'});
graph.newEdge(LesVar, NomVar, {color: '#EDC951'});

graph.newEdge(LesVar, InstrIf, {color: '#7DBE3C'});
graph.newEdge(LesBool, InstrIf, {color: '#000000'});
graph.newEdge(LesBool, BoucleWhile, {color: '#00A0B0'});
graph.newEdge(OpNumb, OpVar, {color: '#6A4A3C'});
graph.newEdge(AutrType, OpEnt, {color: '#CC333F'});

graph.newEdge(Affich, BoucleWhile, {color: '#EB6841'});
graph.newEdge(Affich, EntreInput, {color: '#EDC951'});

graph.newEdge(InstrIf, InstrIfElif, {color: '#7DBE3C'});
graph.newEdge(InstrIf, InstrIfElse, {color: '#000000'});
graph.newEdge(InstrIf, UtilCompt, {color: '#00A0B0'});

```

```
graph.newEdge(TypeDonnPy, BoucleWhile, {color: '#6A4A3C'});
graph.newEdge(OpFlot, OpVar, {color: '#CC333F'});
```

```
graph.newEdge(OpEnt, OpVar, {color: '#7DBE3C'});
```

```
graph.newEdge(InstrIfElif, InstrIfElifElse, {color: '#000000'});
graph.newEdge(InstrIfElse, InstrIfElifElse, {color: '#00A0B0'});
graph.newEdge(BoucleWhile, UtilCompt, {color: '#6A4A3C'});
graph.newEdge(OpVar, EntreInput, {color: '#CC333F'});
graph.newEdge(OpVar, UtilCompt, {color: '#EB6841'});
```

```
/*graph.newEdge(, , {color: '#EDC951'});
```

```
graph.newEdge(, , {color: '#7DBE3C'});
graph.newEdge(, , {color: '#000000'});
graph.newEdge(, , {color: '#00A0B0'});
graph.newEdge(, , {color: '#6A4A3C'});
graph.newEdge(, , {color: '#CC333F'});
graph.newEdge(, , {color: '#EB6841'});
graph.newEdge(, , {color: '#EDC951'});
```

```
graph.newEdge(, , {color: '#7DBE3C'});
graph.newEdge(, , {color: '#000000'});
graph.newEdge(, , {color: '#00A0B0'});
graph.newEdge(, , {color: '#6A4A3C'});
graph.newEdge(, , {color: '#CC333F'});
graph.newEdge(, , {color: '#EB6841'});
graph.newEdge(, , {color: '#EDC951'});
```

```
*/
/* affiche les identifiants des noeuds unités
*/
//document.write(graph.nodes[0].data.label);
//document.write(Affich.id);
```

```
/*graph.removeNode(Affich);
graph.removeNode(OpVar);
graph.removeNode(OpEnt);
*/
```

```
function GetURLUnit(s) {
```

```
    Rac = "../unites&html/";
    switch (s){
```

```
        case "Généralités sur Python":
            URL = Rac + "generalites_sur_python.html";
```

```

break;

case "Code Python en action":
URL = Rac + "code_python_en_action.html";
break;

case "Le processus de programmation en python":
URL = Rac + "le_processus_de_programmation_en_python.html";
break;

case "Les deux modes d'édition":
URL = Rac + "les_deux_modes_d_edition.html";
break;

case "La session interactive":
URL = Rac + "session_interactive.html";
break;

case "Le mode fichier":
URL = Rac + "mode_fichier.html";
break;

case "Let's go":
URL = Rac + "let_s_go.html";
break;

case "Découvrir Python":
URL = Rac+"decouvrir_python.html";
break;

case "La bibliothèque standard":
URL = Rac+"la_bibliotheque_standard.html";
break;

case "Placer des commentaires":
URL = Rac+"placer_des_commentaires.html";
break;

case "Les caractères ASCII":
URL = Rac+"caracteres_ascii.html;"
break;

case "Les chaînes : présentation":
URL = Rac+"les_chaines___presentation.html";
break;

case "Les types de bases":
URL = Rac+"types_de_base.html";
break;

```

```
case "Accéder à un module standard":  
URL = Rac+"accéder_a_un_module_standard.html";  
break;
```

```
case "L'encodage":  
URL = Rac+"encodage.html";  
break;
```

```
case "Les variables":  
URL = Rac+"variables.html"  
break;
```

```
case "Les booléens":  
URL = Rac+"les_booleens.html"  
break;
```

```
case "Les opérations sur les nombres":  
URL = Rac+"opérations_sur_les_nombres.html";  
break;
```

```
case "Les autres types":  
URL = Rac+"autres_types.html";  
break;
```

```
case "Le hasard en Python":  
URL = Rac+"le_hasard_en_python.html";  
break;
```

```
case "Afficher":  
URL = Rac+"afficher.html";  
break;
```

```
case "Les noms de variables":  
URL = Rac+"noms_de_variables.html";  
break;
```

```
case "L'instruction if":  
URL = Rac+"instruction_if.html";  
break;
```

```
case "Les types de données en Python":  
URL = Rac+"types_de_donnees_en_python.html";  
break;
```

```
case "Les opérations sur les flottants":  
URL = Rac+"opérations_sur_les_flottants.html";  
break;
```

```
case "Les opérations sur les entiers":
```

```

URL = Rac+"operations_sur_les_entiers.html";
break;

case "La fonction randrange":
URL = Rac+"fonction_randrange.html";
break;

case "L'instruction if/elif":
URL = Rac+"instruction_if_elif.html";
break;

case "L'instruction if/else":
URL = Rac+"instruction_if_else.html";
break;

case "La boucle while":
URL = Rac+"boucle_while.html";
break;

case "Les opérations avec des variables":
URL = Rac+"operations_avec_des_variables.html";
break;

case "L'instruction if/elif/else":
URL = Rac+"instruction_if_elif_else.html";
break;

case "Les entrées via la fonction input":
URL = Rac+"les_entrees_via_la_fonction_input.html";
break;

case "L'utilisation d'un compteur":
URL = Rac+"utilisation_d_un_compteur.html";
break;

}
//URL = Rac+"let_s_go.html";
return URL;
}

$(document).ready(function(){
//jQuery('#springydemo').springy({ 'graph': graph });
$('#springydemo').springy({ 'graph': graph});
/*$('#OpVar').hide();
$('#Affich').hide();*/

```

```
});
```

II. Modèle de données sous JSON (adapté au framework JIT)

```
[  
  {  
    "adjacencies": [  
      {  
        "nodeTo": "graphnode2",  
        "nodeFrom": "graphnode0",  
        "data": {}  
      }  
    ],  
    "data": {  
      "$color": "#416D9C",  
      "$type": "circle"  
    },  
    "id": "graphnode0",  
    "name": "Code Python en action"  
  },  
  {  
    "adjacencies": [  
      {  
        "nodeTo": "graphnode2",  
        "nodeFrom": "graphnode1",  
        "data": {}  
      }  
    ],  
    "data": {  
      "$color": "#416D9C",  
      "$type": "circle"  
    },  
    "id": "graphnode1",  
    "name": "Généralités sur Python"  
  },  
  {  
    "adjacencies": [  
      {  
        "nodeTo": "graphnode3",  
        "nodeFrom": "graphnode2",  
        "data": {}  
      }  
    ],  
  }  
]
```

```

"data": {
  "$color": "#416D9C",
  "$type": "circle"
},
"id": "graphnode2",
"name": "Le processus de programmation en Python"
},
{
  "adjacencies": [
    {
      "nodeTo": "graphnode4",
      "nodeFrom": "graphnode3",
      "data": {}
    },
    {
      "nodeTo": "graphnode5",
      "nodeFrom": "graphnode3",
      "data": {}
    }
  ],
  "data": {
    "$color": "#416D9C",
    "$type": "circle"
  },
  "id": "graphnode3",
  "name": "Les deux modes d'édition"
},
{
  "adjacencies": [{
    "nodeTo": "graphnode6",
    "nodeFrom": "graphnode4",
    "data": {}
  },
  {
    "nodeTo": "graphnode7",
    "nodeFrom": "graphnode4",
    "data": {}
  },
  {
    "nodeTo": "graphnode8",
    "nodeFrom": "graphnode4",
    "data": {}
  }
  ],
  "data": {
    "$color": "#416D9C",
    "$type": "circle"
  },
  "id": "graphnode4",

```

```

    "name": "Mode fichier"
  },
  {
    "adjacencies": [
      {
        "nodeTo": "graphnode8",

"nodeFrom": "graphnode5",
      "data": {}
    },
    {
      "nodeTo": "graphnode7",
      "nodeFrom": "graphnode5",
      "data": {}
    },
    {
      "nodeTo": "graphnode9",
      "nodeFrom": "graphnode5",
      "data": {}
    },
    {
      "nodeTo": "graphnode10",
      "nodeFrom": "graphnode5",
      "data": {}
    }
  ],
  "data": {
    "$color": "#416D9C",
    "$type": "circle"
  },
  "id": "graphnode5",
  "name": "Session interactive"
},
{
  "adjacencies": [

],
  "data": {
    "$color": "#416D9C",
    "$type": "circle"
  },
  "id": "graphnode6",
  "name": "Exécuter en mode fichier sous Linux"
},
{
  "adjacencies": [{
    "nodeTo": "graphnode12",
    "nodeFrom": "graphnode7",
    "data": {}

```

```

    },
    {
      "nodeTo": "graphnode11",
      "nodeFrom": "graphnode7",
      "data": {}
    },
    {
      "nodeTo": "graphnode13",
      "nodeFrom": "graphnode7",
      "data": {}
    }
  ]],
  "data": {
    "$color": "#416D9C",
    "$type": "circle"
  },
  "id": "graphnode7",
  "name": "IDLE : présentation"
},
{
  "adjacencies": [{
    "nodeTo": "graphnode46",
    "nodeFrom": "graphnode8",
    "data": {}
  }],
  "data": {
    "$color": "#EBB056",
    "$type": "star"
  },
  "id": "graphnode8",
  "name": "Let's go"
},
{
  "adjacencies": [],
  "data": {
    "$color": "#416D9C",
    "$type": "circle"
  },
  "id": "graphnode9",
  "name": "Ouvrir une console sous Linux"
},
{
  "adjacencies": [
    {
      "nodeTo": "graphnode11",
      "nodeFrom": "graphnode10",
      "data": {}
    }
  ]
},
  "data": {
    "$color": "#416D9C",

```

```

    "$type": "circle"
  },
  "id": "graphnode10",
  "name": "Ouvrir une console sous Windows"
},
{
  "adjacencies": [],
  "data": {
    "$color": "#416D9C",
    "$type": "circle"
  },
  "id": "graphnode11",
  "name": "Ouvrir IDLE sous Linux"
},
{
  "adjacencies": [],
  "data": {
    "$color": "#416D9C",
    "$type": "circle"
  },
  "id": "graphnode12",
  "name": "Installer IDLE sous Linux"
},
{
  "adjacencies": [
    {
      "nodeTo": "graphnode14",
      "nodeFrom": "graphnode13",
      "data": {}
    }
  ],
  "data": {
    "$color": "#416D9C",
    "$type": "circle"
  },
  "id": "graphnode13",
  "name": "Ouvrir IDLE sous Windows"
},
{
  "adjacencies": [],
  "data": {
    "$color": "#416D9C",
    "$type": "circle"
  },
  "id": "graphnode14",
  "name": "Mode fichier dans IDLE"
},
{
  "adjacencies": [
    {

```

```

    "nodeTo": "graphnode16",
    "nodeFrom": "graphnode15",
    "data": {}
  },
  {
    "nodeTo": "graphnode17",
    "nodeFrom": "graphnode15",
    "data": {}
  }
],
"data": {
  "$color": "#416D9C",
  "$type": "circle"
},
"id": "graphnode15",
"name": "Mode interactif dans IDLE"
},
{
  "adjacencies": [
    {
      "nodeTo": "graphnode17",
      "nodeFrom": "graphnode16",
      "data": {}
    }
  ],
  "data": {
    "$color": "#416D9C",
    "$type": "circle"
  },
  "id": "graphnode16",
  "name": "Fonctionnement d'IDLE"
},
{
  "adjacencies": [],
  "data": {
    "$color": "#416D9C",
    "$type": "circle"
  },
  "id": "graphnode17",
  "name": "Limitation d'IDLE"
},
{
  "adjacencies": [
    {
      "nodeTo": "graphnode19",
      "nodeFrom": "graphnode18",
      "data": {}
    },
    {
      "nodeTo": "graphnode20",

```

```

    "nodeFrom": "graphnode18",
    "data": {}
  }
],
"data": {
  "$color": "#416D9C",
  "$type": "circle"
},
"id": "graphnode18",
"name": "Mode mixte interactif et fichier"
},{
  "adjacencies": [],
  "data": {
    "$color": "#416D9C",
    "$type": "circle"
  },
  "id": "graphnode19",
  "name": "Constantes"
},
{
  "adjacencies": [],
  "data": {
    "$color": "#416D9C",
    "$type": "circle"
  },
  "id": "graphnode20",
  "name": "Constantes entières"
},
{
  "adjacencies": [],
  "data": {
    "$color": "#416D9C",
    "$type": "circle"
  },
  "id": "graphnode21",
  "name": "Constantes flottantes"
},{
  "adjacencies": [],
  "data": {
    "$color": "#416D9C",
    "$type": "circle"
  },
  "id": "graphnode22",
  "name": "Constantes base 2"
},{
  "adjacencies": [],
  "data": {
    "$color": "#416D9C",
    "$type": "circle"
  },
  "id": "graphnode23",
  "name": "Constantes base 2"
},

```

```

    "id": "graphnode23",
    "name": "Constantes base 16"
  },{
    "adjacencies": [],
    "data": {
      "$color": "#416D9C",
      "$type": "circle"
    },
    "id": "graphnode24",
    "name": "Approximation des flottants"
  },{
    "adjacencies": [],
    "data": {
      "$color": "#416D9C",
      "$type": "circle"
    },
    "id": "graphnode25",
    "name": "Le module math"
  },{
    "adjacencies": [],
    "data": {
      "$color": "#416D9C",
      "$type": "circle"
    },
    "id": "graphnode26",
    "name": "Quotes autour d'une chaîne"
  },{
    "adjacencies": [],
    "data": {
      "$color": "#416D9C",
      "$type": "circle"
    },
    "id": "graphnode27",
    "name": "Les caractères d'échappement"
  },{
    "adjacencies": [],
    "data": {
      "$color": "#416D9C",
      "$type": "circle"
    },
    "id": "graphnode28",
    "name": "Guillemets vs double apostrophe"
  },{
    "adjacencies": [],
    "data": {
      "$color": "#416D9C",
      "$type": "circle"
    },
    "id": "graphnode29",
    "name": "Chaînes triples"
  }

```

```

},{
  "adjacencies": [],
  "data": {
    "$color": "#416D9C",

    "$type": "circle"
  },
  "id": "graphnode30",
  "name": "Edition lisible de chaînes"
},{
  "adjacencies": [],
  "data": {
    "$color": "#416D9C",
    "$type": "circle"
  },
  "id": "graphnode31",
  "name": "Caractères \' et \' "
},{
  "adjacencies": [],
  "data": {
    "$color": "#416D9C",
    "$type": "circle"
  },
  "id": "graphnode32",
  "name": "Saut de ligne \\n"
},{
  "adjacencies": [],
  "data": {
    "$color": "#416D9C",
    "$type": "circle"
  },
  "id": "graphnode33",
  "name": "Antislash \\\"
},{
  "adjacencies": [],
  "data": {
    "$color": "#416D9C",
    "$type": "circle"
  },
  "id": "graphnode34",
  "name": "Tabulation \\t"
},{
  "adjacencies": [],
  "data": {
    "$color": "#416D9C",
    "$type": "circle"
  },
  "id": "graphnode35",
  "name": "Afficher des chaînes en mode interactif"
},{

```

```

"adjacencies": [],
"data": {
  "$color": "#416D9C",
  "$type": "circle"
},
"id": "graphnode36",
"name": "Ligne et \\ENTER"
},{
"adjacencies": [],
"data": {
  "$color": "#416D9C",
  "$type": "circle"
},
"id": "graphnode37",
"name": "Chaînes brutes"
}
, {
"adjacencies": [],
"data": {
  "$color": "#416D9C",
  "$type": "circle"
},
"id": "graphnode38",
"name": "Chaîne triple vers chaîne"
}, {
"adjacencies": [],
"data": {
  "$color": "#416D9C",
  "$type": "circle"
},
"id": "graphnode39",
"name": "Séparer une chaîne suivant des sous-chaînes"
}, {
"adjacencies": [],
"data": {
  "$color": "#416D9C",
  "$type": "circle"
},
"id": "graphnode40",
"name": "Quotes dans une chaîne"
}, {
"adjacencies": [],
"data": {
  "$color": "#416D9C",
  "$type": "circle"
},
"id": "graphnode41",
"name": "Documentation d'une fonction"
}, {
"adjacencies": [],

```

```

"data": {
  "$color": "#416D9C",
  "$type": "circle"
},
"id": "graphnode42",
"name": "Espaces, sauts de ligne et tabulations"
},{
"adjacencies": [],
"data": {
  "$color": "#416D9C",
  "$type": "circle"
},
"id": "graphnode43",
"name": "Saut de ligne dans le code source"
},{
"adjacencies": [],
"data": {
  "$color": "#416D9C",
  "$type": "circle"
},
"id": "graphnode44",
"name": "Espaces, sauts de ligne et tabulations"
},{
"adjacencies": [{
  "nodeTo": "graphnode52",
  "nodeFrom": "graphnode45",
  "data": {}
}],
"data": {
  "$color": "#416D9C",
  "$type": "circle"
},
"id": "graphnode45",
"name": "Accéder à un module standard"
},{
"adjacencies": [
  {
    "nodeTo": "graphnode47",
    "nodeFrom": "graphnode46",
    "data": {}
  },{
    "nodeTo": "graphnode48",
    "nodeFrom": "graphnode46",
    "data": {}
  },{
    "nodeTo": "graphnode50",
    "nodeFrom": "graphnode46",
    "data": {}
  },{
    "nodeTo": "graphnode51",

```

```

    "nodeFrom": "graphnode46",
    "data": {
      },
    ],
    "data": {
      "$color": "#416D9C",
      "$type": "circle"
    },
    "id": "graphnode46",
    "name": "Découvrir Python"
  },{
    "adjacencies": [{
      "nodeTo": "graphnode45",
      "nodeFrom": "graphnode47",
      "data": {
      }, ],
    "data": {
      "$color": "#416D9C",
      "$type": "circle"
    },
    "id": "graphnode47",
    "name": "La bibliothèque standard"
  },{
    "adjacencies": [{
      "nodeTo": "graphnode54",
      "nodeFrom": "graphnode48",
      "data": {
      }, ],
    "data": {
      "$color": "#416D9C",
      "$type": "circle"
    },
    "id": "graphnode48",
    "name": "Placer des commentaires"
  },{
    "adjacencies": [{
      "nodeTo": "graphnode54",
      "nodeFrom": "graphnode49",
      "data": {
      }, ],
    "data": {
      "$color": "#416D9C",
      "$type": "circle"
    },
    "id": "graphnode49",
    "name": "Caractères ASCII"
  },{
    "adjacencies": [{
      "nodeTo": "graphnode35",
      "nodeFrom": "graphnode50",

```

```

    "data": {}
  }, ],
  "data": {
    "$color": "#416D9C",
    "$type": "circle"
  },
  "id": "graphnode50",
  "name": "Les chaînes : présentation"
}, {
  "adjacencies": [{
    "nodeTo": "graphnode56",
    "nodeFrom": "graphnode51",
    "data": {}
  }, {
    "nodeTo": "graphnode57",
    "nodeFrom": "graphnode51",
    "data": {}
  }, {
    "nodeTo": "graphnode58",
    "nodeFrom": "graphnode51",
    "data": {}
  }, {
    "nodeTo": "graphnode55",
    "nodeFrom": "graphnode51",
    "data": {}
  }, {
    "nodeTo": "graphnode59",
    "nodeFrom": "graphnode51",
    "data": {}
  }, {
    "nodeTo": "graphnode60",
    "nodeFrom": "graphnode51",
    "data": {}
  } ],
  "data": {
    "$color": "#416D9C",
    "$type": "circle"
  },
  "id": "graphnode51",
  "name": "Types de base"
}, {
  "adjacencies": [{
    "nodeTo": "graphnode53",
    "nodeFrom": "graphnode52",
    "data": {}
  } ],
  "data": {
    "$color": "#416D9C",
    "$type": "circle"
  },
},

```

```

    "id": "graphnode52",
    "name": "Le hasard en Python"
  },{
    "adjacencies": [],
    "data": {
      "$color": "#416D9C",
      "$type": "circle"
    },
    "id": "graphnode53",
    "name": "Fonction randrange"
  },{
    "adjacencies": [],
    "data": {
      "$color": "#416D9C",
      "$type": "circle"
    },
    "id": "graphnode54",
    "name": "Encodage"
  },{
    "adjacencies": [{
      "nodeTo": "graphnode62",
      "nodeFrom": "graphnode55",
      "data": {}
    }, {
      "nodeTo": "graphnode63",
      "nodeFrom": "graphnode55",
      "data": {}
    }, {
      "nodeTo": "graphnode64",
      "nodeFrom": "graphnode55",
      "data": {}
    } ],
    "data": {
      "$color": "#416D9C",
      "$type": "circle"
    },
    "id": "graphnode55",
    "name": "Variables"
  },{
    "adjacencies": [{
      "nodeTo": "graphnode64",
      "nodeFrom": "graphnode56",
      "data": {}
    }, {
      "nodeTo": "graphnode67",
      "nodeFrom": "graphnode56",
      "data": {}
    } ],
    "data": {
      "$color": "#416D9C",

```

```

    "$type": "circle"
  },
  "id": "graphnode56",
  "name": "Les booléens"
}, {
  "adjacencies": [{
    "nodeTo": "graphnode68",
    "nodeFrom": "graphnode57",
    "data": {}
  }, ],
  "data": {
    "$color": "#416D9C",
    "$type": "circle"
  },
  "id": "graphnode57",
  "name": "Opérations sur les nombres"
}, {
  "adjacencies": [{
    "nodeTo": "graphnode61",
    "nodeFrom": "graphnode58",
    "data": {}
  }, ],
  "data": {
    "$color": "#416D9C",
    "$type": "circle"
  },
  "id": "graphnode58",
  "name": "Autres types"
}, {
  "adjacencies": [{
    "nodeTo": "graphnode67",
    "nodeFrom": "graphnode59",
    "data": {}
  }, ],
  "data": {
    "$color": "#416D9C",
    "$type": "circle"
  },
  "id": "graphnode59",
  "name": "Types de données en Python"
}, {
  "adjacencies": [{
    "nodeTo": "graphnode68",
    "nodeFrom": "graphnode60",
    "data": {}
  }, ],
  "data": {
    "$color": "#416D9C",
    "$type": "circle"
  },
  "id": "graphnode60",
  "name": "Types de données en Python"
}, {
  "adjacencies": [{
    "nodeTo": "graphnode69",
    "nodeFrom": "graphnode61",
    "data": {}
  }, ],
  "data": {
    "$color": "#416D9C",
    "$type": "circle"
  },
  "id": "graphnode61",
  "name": "Autres types"
}, {
  "adjacencies": [{
    "nodeTo": "graphnode62",
    "nodeFrom": "graphnode62",
    "data": {}
  }, ],
  "data": {
    "$color": "#416D9C",
    "$type": "circle"
  },
  "id": "graphnode62",
  "name": "Autres types"
}, {
  "adjacencies": [{
    "nodeTo": "graphnode63",
    "nodeFrom": "graphnode63",
    "data": {}
  }, ],
  "data": {
    "$color": "#416D9C",
    "$type": "circle"
  },
  "id": "graphnode63",
  "name": "Autres types"
}, {
  "adjacencies": [{
    "nodeTo": "graphnode64",
    "nodeFrom": "graphnode64",
    "data": {}
  }, ],
  "data": {
    "$color": "#416D9C",
    "$type": "circle"
  },
  "id": "graphnode64",
  "name": "Autres types"
}, {
  "adjacencies": [{
    "nodeTo": "graphnode65",
    "nodeFrom": "graphnode65",
    "data": {}
  }, ],
  "data": {
    "$color": "#416D9C",
    "$type": "circle"
  },
  "id": "graphnode65",
  "name": "Autres types"
}, {
  "adjacencies": [{
    "nodeTo": "graphnode66",
    "nodeFrom": "graphnode66",
    "data": {}
  }, ],
  "data": {
    "$color": "#416D9C",
    "$type": "circle"
  },
  "id": "graphnode66",
  "name": "Autres types"
}, {
  "adjacencies": [{
    "nodeTo": "graphnode67",
    "nodeFrom": "graphnode67",
    "data": {}
  }, ],
  "data": {
    "$color": "#416D9C",
    "$type": "circle"
  },
  "id": "graphnode67",
  "name": "Autres types"
}, {
  "adjacencies": [{
    "nodeTo": "graphnode68",
    "nodeFrom": "graphnode68",
    "data": {}
  }, ],
  "data": {
    "$color": "#416D9C",
    "$type": "circle"
  },
  "id": "graphnode68",
  "name": "Opérations sur les nombres"
}, {
  "adjacencies": [{
    "nodeTo": "graphnode69",
    "nodeFrom": "graphnode69",
    "data": {}
  }, ],
  "data": {
    "$color": "#416D9C",
    "$type": "circle"
  },
  "id": "graphnode69",
  "name": "Opérations sur les nombres"
}
]
}

```

```

    "id": "graphnode60",
    "name": "Opérations sur les flottants"
  }, {
    "adjacencies": [{
      "nodeTo": "graphnode68",
      "nodeFrom": "graphnode61",
      "data": {}
    }, ],
    "data": {
      "$color": "#416D9C",
      "$type": "circle"
    },
    "id": "graphnode61",
    "name": "Opérations sur les entiers"
  }, {
    "adjacencies": [{
      "nodeTo": "graphnode70",
      "nodeFrom": "graphnode62",
      "data": {}
    }, {
      "nodeTo": "graphnode67",
      "nodeFrom": "graphnode62",
      "data": {}
    }, ],
    "data": {
      "$color": "#416D9C",
      "$type": "circle"
    },
    "id": "graphnode62",
    "name": "Afficher"
  }, {
    "adjacencies": [],
    "data": {
      "$color": "#416D9C",
      "$type": "circle"
    },
    "id": "graphnode63",
    "name": "Noms des variables"
  }, {
    "adjacencies": [{
      "nodeTo": "graphnode65",
      "nodeFrom": "graphnode64",
      "data": {}
    }, {
      "nodeTo": "graphnode66",
      "nodeFrom": "graphnode64",
      "data": {}
    }, {
      "nodeTo": "graphnode71",
      "nodeFrom": "graphnode64",

```

```

    "data": {}
  }, ],
  "data": {
    "$color": "#416D9C",
    "$type": "circle"
  },
  "id": "graphnode64",
  "name": "Instruction if"
}, {
  "adjacencies": [{
    "nodeTo": "graphnode69",
    "nodeFrom": "graphnode65",
    "data": {}
  }],
  "data": {
    "$color": "#416D9C",
    "$type": "circle"
  },
  "id": "graphnode65",
  "name": "Instruction if/elif"
}, {
  "adjacencies": [{
    "nodeTo": "graphnode69",
    "nodeFrom": "graphnode65",
    "data": {}
  }],
  "data": {
    "$color": "#416D9C",
    "$type": "circle"
  },
  "id": "graphnode66",
  "name": "Instruction if/else"
}, {
  "adjacencies": [{
    "nodeTo": "graphnode71",
    "nodeFrom": "graphnode67",
    "data": {}
  }],
  "data": {
    "$color": "#416D9C",
    "$type": "circle"
  },
  "id": "graphnode67",
  "name": "Boucle while"
}, {
  "adjacencies": [{
    "nodeTo": "graphnode70",
    "nodeFrom": "graphnode68",
    "data": {}
  }],
  {

```

```

    "nodeTo": "graphnode71",
    "nodeFrom": "graphnode68",
    "data": {}
  }, ],
  "data": {
    "$color": "#416D9C",
    "$type": "circle"
  },
  "id": "graphnode68",
  "name": "Opérations avec des variables"
}, {
  "adjacencies": [],
  "data": {
    "$color": "#416D9C",
    "$type": "circle"
  },
  "id": "graphnode69",
  "name": "Instruction if/elif/else"
}, {
  "adjacencies": [],
  "data": {
    "$color": "#416D9C",
    "$type": "circle"
  },
  "id": "graphnode70",
  "name": "Les entrées via la fonction input"
}, {
  "adjacencies": [],
  "data": {
    "$color": "#416D9C",
    "$type": "circle"
  },
  "id": "graphnode71",
  "name": "Utilisation d'un compteur"
}
];

```

III. Fichier imsmanifest.xml (archive SCORM)

```
<?xml version="1.0" encoding="UTF-8"?>
<manifest xmlns="http://www.imsglobal.org/xsd/imscp_v1p1"
xmlns:adlcp="http://www.adlnet.org/xsd/adlcp_v1p3"
xmlns:adlseq="http://www.adlnet.org/xsd/adlseq_v1p3"
xmlns:adlnav="http://www.adlnet.org/xsd/adlnav_v1p3"
xmlns:imsss="http://www.imsglobal.org/xsd/imsss"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" identifier="Manifest_2004-4a"
version="1.3" xsi:schemaLocation="http://www.imsglobal.org/xsd/imscp_v1p1
imscp_v1p1.xsd http://www.adlnet.org/xsd/adlcp_v1p3
adlcp_v1p3.xsd http://www.adlnet.org/xsd/adlseq_v1p3
adlseq_v1p3.xsd http://www.adlnet.org/xsd/adlnav_v1p3
adlnav_v1p3.xsd http://www.imsglobal.org/xsd/imsss
imsss_v1p0.xsd">
  <metadata>
    <schema>ADL SCORM</schema>
    <schemaversion>2004 4th Edition</schemaversion>
    <adlcp:location>courseMetadata.xml</adlcp:location> </metadata>
  <organizations default="TOC1">
    <organization identifier="TOC1">
      <title>Initiation au langage Pyhon</title>
      <item identifier="ident_d5cba7e5-3f5e-4316-a7cb-8acae82f58b3"
identifierrref="ref_d5cba7e5-3f5e-4316-a7cb-8acae82f58b3" isvisible="true">
        <title>LET'GO</title>
        <imsss:sequencing>
          <imsss:controlMode choice="true" choiceExit="true" flow="true" forwardOnly="false"
useCurrentAttemptObjectiveInfo="true" useCurrentAttemptProgressInfo="true" />
          <imsss:sequencingRules>
            <imsss:preConditionRule>
              <imsss:ruleConditions conditionCombination="all">
                <imsss:ruleCondition operator="noOp" condition="satisfied" />
              </imsss:ruleConditions>
              <imsss:ruleAction action="skip" />
            </imsss:preConditionRule>
            <imsss:exitConditionRule>
              <imsss:ruleConditions conditionCombination="all">
                <imsss:ruleCondition operator="noOp" condition="satisfied" />
                <imsss:ruleCondition operator="noOp" condition="satisfied" />
              </imsss:ruleConditions>
              <imsss:ruleAction action="exit" />
            </imsss:exitConditionRule>
          </imsss:sequencingRules>
          <imsss:objectives>
            <imsss:primaryObjective satisfiedByMeasure="true" objectiveID="">
              <imsss:minNormalizedMeasure>1</imsss:minNormalizedMeasure>
            </imsss:primaryObjective>
          </imsss:objectives>
        </item>
      </organization>
    </organizations>
  </manifest>
```

```

    </imsss:sequencing>
  </item>
  <item identifier="ident_f281a35a-89f9-4247-8f08-d1caf0c31a85"
  identifierrref="ref_f281a35a-89f9-4247-8f08-d1caf0c31a85" isvisible="true">
    <title>ACCEDER A UN MODULE STANDARD</title>
  </item>
  <item identifier="ident_103f72e9-b838-49b6-9aae-d4404cc63646"
  identifierrref="ref_103f72e9-b838-49b6-9aae-d4404cc63646" isvisible="true">
    <title>AFFICHER</title>
  </item>
  <item identifier="ident_90a3b6bb-7578-41d4-a378-466d253261e6"
  identifierrref="ref_90a3b6bb-7578-41d4-a378-466d253261e6" isvisible="true">
    <title>AUTRES TYPES</title>
  </item>
  <item identifier="ident_ca2804b4-b811-4db1-b66f-9d5a5b487081"
  identifierrref="ref_ca2804b4-b811-4db1-b66f-9d5a5b487081" isvisible="true">
    <title>BOUCLE WHILE</title>
  </item>
  <item identifier="ident_db3c5c23-90a2-4258-a031-629c3741bbc5"
  identifierrref="ref_db3c5c23-90a2-4258-a031-629c3741bbc5" isvisible="true">
    <title>CARACTERES ASCII</title>
  </item>
  <item identifier="ident_10995850-60c1-4e74-be19-cdd9826c3469"
  identifierrref="ref_10995850-60c1-4e74-be19-cdd9826c3469" isvisible="true">
    <title>CODE PYTHON EN ACTION</title>
  </item>
  <item identifier="ident_90f96cc4-61e5-4793-b661-ecelcb078d3b"
  identifierrref="ref_90f96cc4-61e5-4793-b661-ecelcb078d3b" isvisible="true">
    <title>DECOUVRIR PYTHON</title>
  </item>
  <item identifier="ident_41498bd3-2fe0-4940-9e7a-75a280bc0180"
  identifierrref="ref_41498bd3-2fe0-4940-9e7a-75a280bc0180" isvisible="true">
    <title>ENCODAGE</title>
  </item>
  <item identifier="ident_1925e73c-049c-4e6b-96d3-0f4793ce7373"
  identifierrref="ref_1925e73c-049c-4e6b-96d3-0f4793ce7373" isvisible="true">
    <title>FONCTION RANDRANGE</title>
  </item>
  <item identifier="ident_4080328a-e605-473a-af9a-9cdf1058654"
  identifierrref="ref_4080328a-e605-473a-af9a-9cdf1058654" isvisible="true">
    <title>GENERALITES SUR PYTHON</title>
  </item>
  <item identifier="ident_298e8bf9-7d4e-4f65-8bb5-bade2519e0e1"
  identifierrref="ref_298e8bf9-7d4e-4f65-8bb5-bade2519e0e1" isvisible="true">
    <title>INSTRUCTION IF ELIF</title>
  </item>
  <item identifier="ident_617a6f8c-dd2b-4e84-8a2c-383cd79d4dad"
  identifierrref="ref_617a6f8c-dd2b-4e84-8a2c-383cd79d4dad" isvisible="true">
    <title>INSTRUCTION IF</title>
  </item>

```

<item identifi er="ident_56bbcb3b-7863-4de8-a56f-7faa94691c3b"
 identifi erref="ref_56bbcb3b-7863-4de8-a56f-7faa94691c3b" isvisible="true">
 <title>INSTRUCTION IF ELIF ELSE</title>
 </item>
 <item identifi er="ident_df14d130-2b27-4c18-aaf3-321210e419b4"
 identifi erref="ref_df14d130-2b27-4c18-aaf3-321210e419b4" isvisible="true">
 <title>INSTRUCTION IF ELSE</title>
 </item>
 <item identifi er="ident_1ae36c38-ed6a-4fb3-ab11-72577729ce54"
 identifi erref="ref_1ae36c38-ed6a-4fb3-ab11-72577729ce54" isvisible="true">
 <title>LA BIBLIOTHEQUE STANDARD</title>
 </item>
 <item identifi er="ident_4135875c-04d9-43a6-bd01-57d341aa9a81"
 identifi erref="ref_4135875c-04d9-43a6-bd01-57d341aa9a81" isvisible="true">
 <title>LE HASARD EN PYTHON</title>
 </item>
 <item identifi er="ident_0ce1bd3d-13df-4ae9-9374-8ae735bf9549"
 identifi erref="ref_0ce1bd3d-13df-4ae9-9374-8ae735bf9549" isvisible="true">
 <title>LES BOOLEENS</title>
 </item>
 <item identifi er="ident_1c9fa490-d98a-4b97-887b-a8fdc9f49f0a"
 identifi erref="ref_1c9fa490-d98a-4b97-887b-a8fdc9f49f0a" isvisible="true">
 <title>LE PROCESSUS DE PROGRAMMATION EN PYTHON</title>
 </item>
 <item identifi er="ident_14b6fecf-1aa5-4682-83c4-7c394d6a77e5"
 identifi erref="ref_14b6fecf-1aa5-4682-83c4-7c394d6a77e5" isvisible="true">
 <title>LES CHAINES : PRESENTATION</title>
 </item>
 <item identifi er="ident_334310cd-91bf-4a3c-b2e7-7827e1192bb5"
 identifi erref="ref_334310cd-91bf-4a3c-b2e7-7827e1192bb5" isvisible="true">
 <title>LES DEUX MODES D'EDITION</title>
 </item>
 <item identifi er="ident_80c7aa4c-70d0-4a16-933b-fdf117063816"
 identifi erref="ref_80c7aa4c-70d0-4a16-933b-fdf117063816" isvisible="true">
 <title>LES ENTREES VIA LA FONCTION INPUT</title>
 </item>
 <item identifi er="ident_468ab3d0-f7ba-42fe-8925-f25a032f7c99"
 identifi erref="ref_468ab3d0-f7ba-42fe-8925-f25a032f7c99" isvisible="true">
 <title>NOMS DE VARIABLES</title>
 </item>
 <item identifi er="ident_ec454340-cd60-49e1-9058-69d0ef5f91e5"
 identifi erref="ref_ec454340-cd60-49e1-9058-69d0ef5f91e5" isvisible="true">
 <title>MODE FICHER DANS IDLE</title>
 </item>
 <item identifi er="ident_d6acac6c-c996-4542-ba20-4530c6b97007"
 identifi erref="ref_d6acac6c-c996-4542-ba20-4530c6b97007" isvisible="true">
 <title>OPERATIONS AVEC DES VARIABLES</title>
 </item>

```

    <item identifier="ident_8c8c9222-bde7-421b-ae9b-41b43a81a6a6"
    identifierref="ref_8c8c9222-bde7-421b-ae9b-41b43a81a6a6" isvisible="true">
      <title>OPERATIONS SUR LES ENTIERS</title>
    </item>
    <item identifier="ident_ce4bf340-5a9c-4bb6-adbc-cb5e667ce100"
    identifierref="ref_ce4bf340-5a9c-4bb6-adbc-cb5e667ce100" isvisible="true">
      <title>OPERATIONS SUR LES FLOTTANTS</title>
    </item>
    <item identifier="ident_37b5f8dc-eb2e-4e45-bc08-137c732c026d"
    identifierref="ref_37b5f8dc-eb2e-4e45-bc08-137c732c026d" isvisible="true">
      <title>OPERATIONS SUR LES NOMBRES</title>
    </item>
    <item identifier="ident_cd490968-2426-4d11-b69e-4519d693390e"
    identifierref="ref_cd490968-2426-4d11-b69e-4519d693390e" isvisible="true">
      <title>PLACER DES COMMENTAIRES</title>
    </item>
    <item identifier="ident_16f582a9-d67d-493b-ba20-ccb081add474"
    identifierref="ref_16f582a9-d67d-493b-ba20-ccb081add474" isvisible="true">
      <title>SESSION INTERACTIVE</title>
    </item>
    <item identifier="ident_708c980a-1592-4ee7-b536-f590e57aae46"
    identifierref="ref_708c980a-1592-4ee7-b536-f590e57aae46" isvisible="true">
      <title>TYPES DE BASE</title>
    </item>
    <item identifier="ident_7dfd59ef-f140-4f70-91b0-cb128a5a9cae"
    identifierref="ref_7dfd59ef-f140-4f70-91b0-cb128a5a9cae" isvisible="true">
      <title>TYPES DE DONNEES EN PYTHON</title>
    </item>
    <item identifier="ident_fb93a746-e5e1-4d48-b8a3-72e93c4cde57"
    identifierref="ref_fb93a746-e5e1-4d48-b8a3-72e93c4cde57" isvisible="true">
      <title>UTILISATION D'UN COMPTEUR</title>
    </item>
    <item identifier="ident_c4fca6e2-9b2b-4418-b77b-a9243c5c4999"
    identifierref="ref_c4fca6e2-9b2b-4418-b77b-a9243c5c4999" isvisible="true">
      <title>VARIABLES</title>
    </item>
  </organization>
</organizations>
<resources>
  <resource identifier="ref_d5cba7e5-3f5e-4316-a7cb-8acae82f58b3" type="webcontent"
  adlcp:scormType="sco" href="let_s_go.html">
    <file href="let_s_go.html" />
  </resource>
  <resource identifier="ref_f281a35a-89f9-4247-8f08-d1caf0c31a85" type="webcontent"
  adlcp:scormType="sco" href="accéder_a_un_module_standard.html">
    <file href="accéder_a_un_module_standard.html" />
  </resource>
  <resource identifier="ref_103f72e9-b838-49b6-9aae-d4404cc63646" type="webcontent"
  adlcp:scormType="sco" href="afficher.html">
    <file href="afficher.html" />
  </resource>

```

```

</resource>
<resource identifier="ref_90a3b6bb-7578-41d4-a378-466d253261e6" type="webcontent"
adlcp:scormType="sco" href="autres_types.html">
  <file href="autres_types.html" />
</resource>
<resource identifier="ref_ca2804b4-b811-4db1-b66f-9d5a5b487081" type="webcontent"
adlcp:scormType="sco" href="boucle_while.html">
  <file href="boucle_while.html" />
</resource>
<resource identifier="ref_db3c5c23-90a2-4258-a031-629c3741bbc5" type="webcontent"
adlcp:scormType="sco" href="caracteres_ascii.html">
  <file href="caracteres_ascii.html" />
</resource>
<resource identifier="ref_10995850-60c1-4e74-be19-cdd9826c3469" type="webcontent"
adlcp:scormType="sco" href="code_python_en_action.html">
  <file href="code_python_en_action.html" />
</resource>
<resource identifier="ref_90f96cc4-61e5-4793-b661-ece1cb078d3b" type="webcontent"
adlcp:scormType="sco" href="decouvrir_python.html">
  <file href="decouvrir_python.html" />
</resource>
<resource identifier="ref_41498bd3-2fe0-4940-9e7a-75a280bc0180" type="webcontent"
adlcp:scormType="sco" href="encodage.html">
  <file href="encodage.html" />
</resource>
<resource identifier="ref_1925e73c-049c-4e6b-96d3-0f4793ce7373" type="webcontent"
adlcp:scormType="sco" href="fonction_randrange.html">
  <file href="fonction_randrange.html" />
</resource>
<resource identifier="ref_4080328a-e605-473a-af9a-9cdf1058654" type="webcontent"
adlcp:scormType="sco" href="generalites_sur_python.html">
  <file href="generalites_sur_python.html" />
</resource>
<resource identifier="ref_298e8bf9-7d4e-4f65-8bb5-bade2519e0e1" type="webcontent"
adlcp:scormType="sco" href="instruction_if_elif.html">
  <file href="instruction_if_elif.html" />
</resource>
<resource identifier="ref_617a6f8c-dd2b-4e84-8a2c-383cd79d4dad" type="webcontent"
adlcp:scormType="sco" href="instruction_if.html">
  <file href="instruction_if.html" />
</resource>
<resource identifier="ref_56bbcb3b-7863-4de8-a56f-7faa94691c3b" type="webcontent"
adlcp:scormType="sco" href="instruction_if_elif_else.html">
  <file href="instruction_if_elif_else.html" />
</resource>
<resource identifier="ref_df14d130-2b27-4c18-aaf3-321210e419b4" type="webcontent"
adlcp:scormType="sco" href="instruction_if_else.html">
  <file href="instruction_if_else.html" />
</resource>

```

```

<resource identifier="ref_1ae36c38-ed6a-4fb3-ab11-72577729ce54" type="webcontent"
adlcp:scormType="sco" href="la_bibliotheque_standard.html">
  <file href="la_bibliotheque_standard.html" />
</resource>
<resource identifier="ref_4135875c-04d9-43a6-bd01-57d341aa9a81" type="webcontent"
adlcp:scormType="sco" href="le_hasard_en_python.html">
  <file href="le_hasard_en_python.html" />
</resource>
<resource identifier="ref_0ce1bd3d-13df-4ae9-9374-8ae735bf9549" type="webcontent"
adlcp:scormType="sco" href="les_booleens.html">
  <file href="les_booleens.html" />
</resource>
<resource identifier="ref_1c9fa490-d98a-4b97-887b-a8fdc9f49f0a" type="webcontent"
adlcp:scormType="sco" href="le_processus_de_programmation_en_python.html">
  <file href="le_processus_de_programmation_en_python.html" />
</resource>
<resource identifier="ref_14b6fecf-1aa5-4682-83c4-7c394d6a77e5" type="webcontent"
adlcp:scormType="sco" href="les_chaines___presentation.html">
  <file href="les_chaines___presentation.html" />
</resource>
<resource identifier="ref_334310cd-91bf-4a3c-b2e7-7827e1192bb5" type="webcontent"
adlcp:scormType="sco" href="les_deux_modes_d_edition.html">
  <file href="les_deux_modes_d_edition.html" />
</resource>
<resource identifier="ref_80c7aa4c-70d0-4a16-933b-fdf117063816" type="webcontent"
adlcp:scormType="sco" href="les_entrees_via_la_fonction_input.html">
  <file href="les_entrees_via_la_fonction_input.html" />
</resource>
<resource identifier="ref_468ab3d0-f7ba-42fe-8925-f25a032f7c99" type="webcontent"
adlcp:scormType="sco" href="noms_de_variables.html">
  <file href="noms_de_variables.html" />
</resource>
<resource identifier="ref_ec454340-cd60-49e1-9058-69d0ef5f91e5" type="webcontent"
adlcp:scormType="sco" href="mode_fichier_dans_IDLE.html">
  <file href="mode_fichier_dans_IDLE.html" />
</resource>
<resource identifier="ref_d6acac6c-c996-4542-ba20-4530c6b97007" type="webcontent"
adlcp:scormType="sco" href="operations_avec_des_variables.html">
  <file href="operations_avec_des_variables.html" />
</resource>
<resource identifier="ref_8c8c9222-bde7-421b-ae9b-41b43a81a6a6" type="webcontent"
adlcp:scormType="sco" href="operations_sur_les_entiers.html">
  <file href="operations_sur_les_entiers.html" />
</resource>
<resource identifier="ref_ce4bf340-5a9c-4bb6-adbc-cb5e667ce100" type="webcontent"
adlcp:scormType="sco" href="operations_sur_les_flottants.html">
  <file href="operations_sur_les_flottants.html" />
</resource>
<resource identifier="ref_37b5f8dc-eb2e-4e45-bc08-137c732c026d" type="webcontent"
adlcp:scormType="sco" href="operations_sur_les_nombres.html">

```

```

    <file href="operations_sur_les_nombres.html" />
  </resource>
  <resource identifiant="ref_cd490968-2426-4d11-b69e-4519d693390e" type="webcontent"
adlcp:scormType="sco" href="placer_des_commentaires.html">
    <file href="placer_des_commentaires.html" />
  </resource>
  <resource identifiant="ref_16f582a9-d67d-493b-ba20-ccb081add474" type="webcontent"
adlcp:scormType="sco" href="session_interactive.html">
    <file href="session_interactive.html" />
  </resource>
  <resource identifiant="ref_708c980a-1592-4ee7-b536-f590e57aae46" type="webcontent"
adlcp:scormType="sco" href="types_de_base.html">
    <file href="types_de_base.html" />
  </resource>
  <resource identifiant="ref_7dfd59ef-f140-4f70-91b0-cb128a5a9cae" type="webcontent"
adlcp:scormType="sco" href="types_de_donnees_en_python.html">
    <file href="types_de_donnees_en_python.html" />
  </resource>
  <resource identifiant="ref_fb93a746-e5e1-4d48-b8a3-72e93c4cde57" type="webcontent"
adlcp:scormType="sco" href="utilisation_d_un_compteur_.html">
    <file href="utilisation_d_un_compteur_.html" />
  </resource>
  <resource identifiant="ref_c4fca6e2-9b2b-4418-b77b-a9243c5c4999" type="webcontent"
adlcp:scormType="sco" href="Variables.html">
    <file href="Variables.html" />
  </resource>
</resources>
</manifest>

```

IV. Quelques fichiers Zend Framework (models, views et controllers)

1. Configuration

Le framework peut être téléchargé sur <http://framework.zend.com/download/latest> .

La version 1.11.3 nécessite un serveur PHP5.2.4 (ou plus) et un serveur web supportant le mode de réécriture (mod_rewrite). L'extension mod_rewrite doit être installée. Le serveur apache doit supporter les fichiers .htaccess notamment en changeant la directive "AllowOverride None" à "AllowOverride All" (dans httpd.conf). La création d'un hôte virtuel passe par les instructions suivantes:

```

<VirtualHost *:80>
    ServerName zf-tutorial.localhost
    DocumentRoot /var/www/html/zf-tutorial/public

```

```
<Directory "/var/www/html/zf-tutorial/public">
```

```
    AllowOverride All
```

```
</Directory>
```

```
</VirtualHost>
```

2. Organisation d'une application

Une architecture typique de ZF serait de la forme suivante :

Le dossier library doit contenir le dossier library/Zend de l'archive Zend Framework.

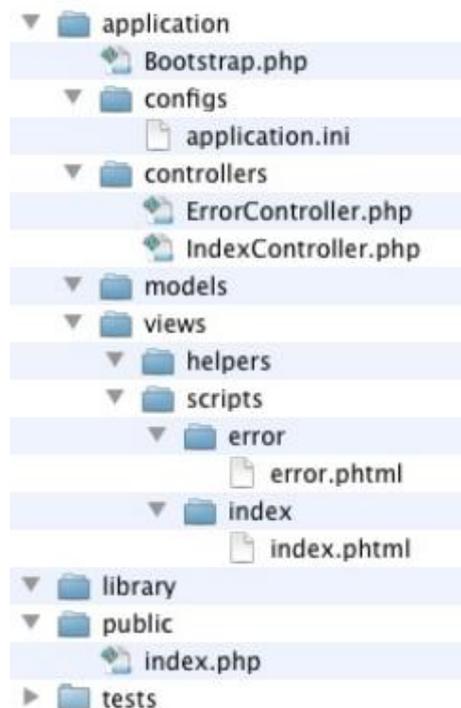


Figure 97 : organisation d'une application Zend Framework

Chaque contrôleur contient une série d'actions. Les fichiers .phtml sont des templates de vue. La démarche utilisée dans ZF est d'utiliser un contrôleur frontal (Front Controller) qui dirige toutes les requêtes à travers un fichier index.php (ou encore bootstrap.php). Ceci aura pour tâche de créer une instance Zend_Application. Le fichier application.ini contient la configuration en termes de variable de chemins de bibliothèques ou de base de données et ce dans différents cycles (développement, tests, production).

3. Quelques fichiers de l'application PRECEPTOR

a. Le contrôleur Ressources

```
<?php
```

```
class RessourcesController extends Zend_Controller_Action
```

```

{

    protected $auth,$username,$author;

    public function init()
    {
        /* Initialize action controller here */

        $auth = Zend_Auth::getInstance();
        $username = $auth->getIdentity()->username;
        $author = "Pascal ortiz";
        $this->view->setEscape('utf8_encode');
    }

    public function changervueAction(){

        echo '<br/>';
        $Url = $this->view->url(array('controller' => 'ressources', 'action' => 'index'), 'default',
true);
        echo '<a href="'. $Url. ">Changer de vue</a>';
        echo '<br/>';

    }

    public function indexAction()
    {
        //$var = $_POST['name'];
        echo '<br/>';
        $Url = $this->view->url(array('controller' => 'ressources', 'action' => 'changervue'),
'default', true);
        echo '<a href="'. $Url. ">Changer de vue</a>';
        echo '<br/>';

    }
}

```

```

public function listevalAction(){

    // action body
    //création d'un d'une instance Default_Model_Users
    $users = new Application_Model_UnitsEvaluation();

    //$this->view permet d'accéder à la vue qui sera utilisée par l'action
    //on initialise la valeur usersArray de la vue
    //(cf. application/views/scripts/users/index.phtml)
    //la valeur correspond à un tableau d'objets de type Default_Model_Users récupérés
    par la méthode fetchAll()
    //création de notre objet Paginator avec comme paramètre la méthode
    //récupérant toutes les entrées dans notre base de données
    $paginator = Zend_Paginator::factory($users->fetchAll());
    //indique le nombre d'éléments à afficher par page
    $paginator->setItemCountPerPage(40);
    //récupère le numéro de la page à afficher
    $paginator->setCurrentPageNumber($this->getRequest()->getParam('page'));

    //$this->view permet d'accéder à la vue qui sera utilisée par l'action
    //on initialise la valeur usersArray de la vue
    //(cf. application/views/scripts/users/index.phtml)
    $this->view->unitsArray = $paginator;
    // $this->view->usersArray = $users->fetchAll();

}

public function deleteevalAction(){
    //récupère les paramètres de la requête
    $params = $this->getRequest()->getParams();

    //vérifie que le paramètre id existe
    if(isset($params['id']))
    {
        $id = $params['id'];
    }
}

```

```

//création du modèle pour la suppression
$evals = new Application_Model_UnitsEvaluation ();
//appel de la fonction de suppression avec en argument,
//la clause where qui sera appliquée
$result = $evals->delete("id=$id");

//redirection
$this->_helper->redirector('listeval');
}
else
{
    $this->view->form = 'Impossible delete: id missing !';
}
}

public function editevalAction(){
    //création du formulaire

    $auth = Zend_Auth::getInstance();
    $username = $auth->getIdentity()->username;
    $form = new Application_Form_UnitsEvaluation();
    //indique l'action qui va traiter le formulaire
    $form->setAction($this->view->url(array('controller' => 'ressources', 'action' => 'editeval'),
'default', true));
    $form->submit->setLabel('Update');

    //assigne le formulaire à la vue
    $this->view->form = $form;

    //si la page est POSTée = formulaire envoyé
    if($this->getRequest()->isPost())
    {
        //récupération des données envoyées par le formulaire
        $data = $this->getRequest()->getPost();
    }
}

```

```

//vérifie que les données répondent aux conditions des validateurs
if($form->isValid($data))
{
    //création et initialisation d'un objet Default_Model_Users
    //qui sera enregistré dans la base de données
    $evals = new Application_Model_UnitsEvaluation();
    $evals->setId((int)$form->getValue('id'));
    // $users = new Application_Model_Units();
    $evals->setFreetext($form->getValue('freetext'));
    $evals->setPresentation($form->getValue('presentation'));
    $evals->setFun($form->getValue('fun'));
    $evals->setPedagogicalInterest($form->getValue('pedagogic'));
    $evals->setUnitName($form->getValue('unitname'));
    $evals->setOriginality($form->getValue('originality'));
    //if($username == $eval->getEvaluator())
    $evals->setEvaluator($username);
    // $users->setRole($form->getElement('role')->getMultiOptions());

    $evals->save();

    //redirection
    $this->_helper->redirector('listeval');
}
else
{
    //si erreur rencontrée, le formulaire est rempli avec les données
    //envoyées précédemment
    $form->populate($data);
}
}
else
{
    //récupération de l'id passé en paramètre
    $id = $this->_getParam('id', 0);
}

```

```

if($id > 0)
{
    //récupération de l'entrée
    $evals = new Application_Model_UnitsEvaluation();
    $eval = $evals->find($id);

    //assignation des valeurs de l'entrée dans un tableau
    //tableau utilisé pour la méthode populate() qui va remplir le champs du formulaire
    //avec les valeurs du tableau
    $data[] = array();
    $data['id'] = $eval->getId();
    $data['presentation'] = $eval->getPresentation();
    $data['freetext'] = $eval->getFreetext();
    $data['originality'] = $eval->getOriginality();
    $data['fun'] = $eval->getFun();
    $data['pedagogic'] = $eval->getPedagogicalInterest();
    $data['unitname'] = $eval->getUnitName();

    //if($username == $eval->getEvaluator())
    $data['evaluator'] = $eval->getEvaluator();

    $form->populate($data);
}
}

public function evalAction(){

    $auth = Zend_Auth::getInstance();
    $username = $auth->getIdentity()->username;
    $author = "Pascal ortiz";

    $form = new Application_Form_UnitsEvaluation();

```

```

//indique l'action qui va traiter le formulaire
$form->setAction($this->view->url(array('controller' => 'ressources', 'action' => 'eval'),
'default', true));
$form->submit->setLabel('Evaluer');

//assigne le formulaire à la vue
$this->view->form = $form;
    $this->view->user = $username;
    $this->view->author = $author;

//si la page est POSTée = formulaire envoyé
if($this->_request->isPost())
{
    //récupération des données envoyées par le formulaire
    $data = $this->_request->getPost();

    //vérifie que les données répondent aux conditions des validateurs
    if($form->isValid($data))
    {
        //création et initialisation d'un objet Default_Model_Users
        //qui sera enregistré dans la base de données
        $evals = new Application_Model_UnitsEvaluation();
        $evals->setUnitId($form->getValue('unitname'));

        // $evals->setEvaluator($form->getValue('evaluator'));
        $evals->setUnitName($form->getValue('unitname'));
        $evals->setEvaluator($username);
        $evals->setFreetext($form->getValue('freetext'));
        $evals->setAuthor($form->getValue('author'));

        $evals->setPedagogicalInterest($form->getValue('pedagogic'));
        $evals->setFun($form->getValue('fun'));
        $evals->setOriginality($form->getValue('originality'));
        $evals->setPresentation($form->getValue('presentation'));
    }
}

```

```

// $users->setRole($form->getElement('role')->getMultiOptions());

$vals->save();
//redirection
$this->_helper->redirector('listeval');
}
else
{
    //si erreur rencontrée, le formulaire est rempli avec les données
    //envoyées précédemment
    $form->populate($data);
}
}
}

/* public function unitAction(){

    $var = $this->getRequest()->getPost('nom_page');
    echo "ok";
}*/

public function historyAction(){

    $auth = Zend_Auth::getInstance();
    $username = $auth->getIdentity()->username;

    echo "dernières unités consultées";
}

public function logAction(){

    $self = $PHP_SELF;
    $chemin = dirname($self);
    echo $chemin;
    echo "test";
}

```

```

}

public function editAction(){
//$this->view->headScript()->appendFile('tinymce/jscripts/tiny_mce/tiny_mce.js');

}

public function editunitAction(){

$this->view->headScript()->appendFile('/tinymce/jscripts/tiny_mce/tiny_mce.js');
    $form = new Application_Form_Units();
//indique l'action qui va traiter le formulaire
    $form->setAction($this->view->url(array('controller' => 'ressources', 'action' => 'editunit'),
'default', true));
    $form->submit->setLabel('Evaluer');

//assigne le formulaire à la vue
$this->view->form = $form;
    $this->view->user = $username;
    $this->view->author = $author;

//si la page est POSTée = formulaire envoyé
if($this->_request->isPost())
{
    //récupération des données envoyées par le formulaire
    $data = $this->_request->getPost();

//vérifie que les données répondent aux conditions des validateurs
if($form->isValid($data))
{
    //création et initialisation d'un objet Default_Model_Users
    //qui sera enregistré dans la base de données
    $evals = new Application_Model_UnitsEvaluation();
    $evals->setUnitId($form->getValue('unitname'));
}
}
}

```

```

//$evals->setEvaluator($form->getValue('evaluator'));
$evals->setUnitName($form->getValue('unitname'));
    $evals->setEvaluator($username);
$evals->setFreetext($form->getValue('freetext'));
$evals->setAuthor($form->getValue('author'));

$evals->setPedagogicalInterest($form->getValue('pedagogic'));
$evals->setFun($form->getValue('fun'));
$evals->setOriginality($form->getValue('originality'));
$evals->setPresentation($form->getValue('presentation'));
// $users->setRole($form->getElement('role')->getMultiOptions());

$evals->save();

//redirection
$this->_helper->redirector('listeval');
}
else
{
    //si erreur rencontrée, le formulaire est rempli avec les données
    //envoyées précédemment
    $form->populate($data);
}
}

}

public function archiveAction(){

    $file = '/public/units_py.zip';

    $zip = new ZipArchive();
    $filename = "python_scorm.zip";

```

```

if ($zip->open($filename, ZIPARCHIVE::CREATE)!==TRUE) {
    exit("Impossible d'ouvrir <$filename>\n");
}

$zip->addFromString("/preceptor/units/afficher.html" . time(), "#1 Ceci est une chaîne
texte, ajoutée comme testfilephp.txt.\n");
$zip->addFromString("/preceptor/units/encodage.html" . time(), "#2 Ceci est une
chaîne texte, ajoutée comme testfilephp2.txt.\n");
$zip->addFile("/preceptor/units/let_s_go.html", "/preceptor/units/boucle_while.html");
echo "Nombre de fichiers : " . $zip->numFiles . "\n";
echo "statut : " . $zip->status . "\n";
$zip->close();

//file_get_contents("units_py.zip");
/*$this->getResponse()
    ->setBody($file)
    ->setHeader('Content-Type', 'application/zip')
    ->setHeader('Content-Disposition', 'attachment; filename="units_py.zip"')
    ->setHeader('Content-Length', strlen($file));
*/

/*$filter = new Zend_Filter_Compress(array(
    'adapter' => 'Zip',
    'options' => array(
        'archive' => '/preceptor/public/python_scorn.zip'
    ),
));

$compressed = $filter->filter('/preceptor/public/units/afficher.html');
$compressed = $filter->filter('/preceptor/public/units/encodage.html');
$compressed = $filter->filter('/preceptor/public/units/boucle_while.html');*/

/*$file = file_get_contents('python_scorn.zip');
$this->getResponse()
    ->setBody($file)
    ->setHeader('Content-Type', 'application/zip')

```

```

        ->setHeader('Content-Disposition', 'attachment;
filename="python_scorm.zip")
        ->setHeader('Content-Length', strlen($file));*/
// Retourne true en cas de succès et crée le fichier d'archive

        /*$zip = new ZipArchive();
        $zip->addFromString ('/preceptor/public/units/afficher.html',
'/preceptor/public/units/afficher.html');
        $zip->addFromString ('/preceptor/public/units/boucle_while.html',
'/preceptor/public/units/boucle_while.html');
        $zip->close();

        //$file = file_get_contents('python_scorm.zip');

        $file = file_get_contents($zip);

        $this->getResponse()
            ->setBody($file)
            ->setHeader('Content-Type', 'application/zip')
            ->setHeader('Content-Disposition', 'attachment;
filename="python_scorm.zip")
            ->setHeader('Content-Length', strlen($file));

        //If using Zend_Layout, we need to disable it:
        $this->_helper->layout->disableLayout();

        //Disable ViewRenderer:
        $this->_helper->viewRenderer->setNoRender(true);*/
    }

    public function trackAction(){

        //    $var = $_POST['name'];
        //    $var = $_POST['id'];
        //    $var = $this->getRequest()->getPost('name');

```

```

        $auth = Zend_Auth::getInstance();
        $username = $auth->getIdentity()->username;
        //$var = $this->getParam('name');
        $users = new Application_Model_Users();

        $users->setUsername($username);
        $users->setPassword("");
        $users->setRole("admin");

        $users->save();
    }

```

b. Exemple de vue index.phtml (lié au contrôleur Ressources)

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title>ForceDirected - Graph Operations</title>

<!-- CSS Files -->
<!-- <link type="text/css" href="<?php echo $this->baseUrl();?>/css/baseBis.css"
rel="stylesheet" /> -->
<!-- <link type="text/css" href="<?php echo $this->baseUrl();?>/css/ForceDirected.css"
rel="stylesheet" /> -->
<!-- <link type="text/css" href="<?php echo $this->baseUrl();?>/css/main.css"
rel="stylesheet" /> -->
<!--[if IE]><script type="text/javascript" src="<?php echo $this-
>baseUrl();?>/js/excanvas.js"></script><![endif]>-->

<!-- JIT Library File -->
<script type="text/javascript" src="<?php echo $this->baseUrl();?>/js/jit.js"></script>
<script type="text/javascript" src="<?php echo $this-
>baseUrl();?>/js/ForceDirectedTestGrComple.js">
</script>

```

```
<script type="text/javascript" src="<?php echo $this->baseUrl();?>/js/jquery.1.4.4.min.js"></script>
```

```
<!-- Example File -->
```

```
<script type="text/javascript">
```

```
$.noConflict();
```

```
$("#sauve").click(function(){
```

```
});
```

```
$(document).ready(function() {
```

```
$("#sauve").click(function() {
```

```
//
```

```
    //window.open("graphUnits.json","enregistrer sous");
```

```
    $.ajax({
```

```
        type: "POST",
```

```
        url: "ressources/changervue",
```

```
        data: "name="+node.name,
```

```
        // datatype: "String",
```

```
        success: function(msg){
```

```
            alert( "Data Saved: " + msg );
```

```
            $('#right-container').html(str);
```

```
        }
```

```
    });
```

```
    alert("ok");
```

```
});
```

```
});
```

```
</script>
```

```
</head>
```

```
<body onload="init();">
```

```
<div id="pageWrapper">
```

```
<div id="left-container">
```

```
  <div class="text">
```

```
    <h4>
```

```
    Graphe des unités pour l'apprentissage du langage Python
```

```
    </h4><br/>
```

```
      <b>Cliquez sur l'intitulé d'un noeud </b> pour sélectionner un noeud et ses liens.<br /><br />
```

```
      <b>Cliquez sur 'x' </b> pour cacher les liens d'un noeud.<br /><br />
```

```
      <b>Cliquez sur '+' </b> pour faire apparaître les liens d'un noeud.<br /><br />
```

```
      Naviguez en <b>traînant les noeuds</b> et <b>zooomez </b> à l'aide de la molette de la souris <b></b><br/><br/>
```

```
      Accédez au contenu d'une unité à l'aide du <b>double clic</b>
```

```
    </div>
```

```
  <div id="id-list"></div>
```

```
</div>
```

```
<div id="infovis">
```

```
</div>
```

```
<div id="right-container">
```

```
<div id="inner-details"></div>
```

```

</div>
<div id="log"></div>
<p>Instructions diverses</p>
<button id="sauve" >sauvegarder le graphe</button>
<button id="colorchm" >colorier chemin</button>

<!--<a type="application/octet-stream" href="graphUnits.json">test</a>
-->

</div>
<iframe src="http://codepad.org" width="810" height="500">
  <p>Your browser does not support iframes.</p>
</iframe>

```

c. Fichier ForceDirectedTestGrComplet.js (sans le modèle de données du graphe des épistèmes)

```

Debut = new Date();
var labelType, useGradients, nativeTextSupport, animate;

(function() {
  var ua = navigator.userAgent,
      iStuff = ua.match(/iPhone/i) || ua.match(/iPad/i),
      typeOfCanvas = typeof HTMLCanvasElement,
      nativeCanvasSupport = (typeOfCanvas === 'object' || typeOfCanvas === 'function'),
      textSupport = nativeCanvasSupport
        && (typeof document.createElement('canvas').getContext('2d').fillText === 'function');
  // I'm setting this based on the fact that ExCanvas provides text support
  // for IE
  // and that as of today iPhone/iPad current text support is lame
  labelType = (!nativeCanvasSupport || (textSupport && !iStuff)) ? 'Native' : 'HTML';
  nativeTextSupport = labelType === 'Native';
  useGradients = nativeCanvasSupport;
  animate = !(iStuff || !nativeCanvasSupport);

```

```
})();
```

```
var Log = {  
  elem: false,  
  write: function(text){  
    if (!this.elem)  
      this.elem = document.getElementById('log');  
    this.elem.innerHTML = text;  
    this.elem.style.left = (500 - this.elem.offsetWidth / 2) + 'px';  
  }  
};
```

```
/** fonction de mapping nodename/nodepage ou nodeaperçu */
```

```
function convert(n){  
  
  var nom_page = "";  
  
  switch(n){  
  
    case "Afficher":  
  
      nom_page="afficher";  
      break;  
  
    case "Généralités sur Python" :  
      nom_page = "generalites_sur_python";  
      break;  
  
    case "Encodage":  
  
      nom_page="encodage";  
      break;  
  
    case "Instruction if":  
      nom_page = "instruction_if";
```

```
        break;

case "Boucle while":
    nom_page = "boucle_while";
    break;

case "Découvrir Python":
    nom_page = "decouvrir_python";
    break;

case "Le processus de programmation en Python":
    nom_page = "le_processus_de_programmation_en_python";
    break;

case "Les booléens":
    nom_page = "les_booleens";
    break;

case "Les deux modes d'édition":
    nom_page = "les_deux_modes_d_edition";
    break;

case "Let's go":
    nom_page = "let_s_go";
    break;

case "Session interactive":
    nom_page = "session_interactive";
    break;

case "Variables":
    nom_page = "Variables";
    break;

case "Types de base":
```

```

    nom_page = "types_de_base";
    break;

case "Accéder à un module standard":
    nom_page = "accéder_a_un_module_standard";
    break;

case "Autres types":
    nom_page = "autres_types";
    break;

/** test à commencer ici **/

case "Caractères ASCII":
    nom_page = "caracteres_ascii";
    break;

case "Fonction randrange":
    nom_page = "fonction_randrange";
    break;

case "Instruction if/elif/else":
    nom_page = "instruction_if_elif_else";
    break;

case "Instruction if/elif":
    nom_page = "instruction_if_elif";
    break;

case "Instruction if/else":
    nom_page = "instruction_if_else";
    break;

case "La bibliothèque standard":
    nom_page = "la_bibliotheque_standard";
    break;

case "Le hasard en Python":

```

```
    nom_page = "le_hasard_en_python";
    break;

case "Les chaînes : présentation":
    nom_page = "les_chaines___presentation";
    break;

case "Les entrées via la fonction input":
    nom_page = "les_entrees_via_la_fonction_input";
    break;

case "Mode fichier":
    nom_page = "mode_fichier_dans_IDLE";
    break;

case "Noms des variables":
    nom_page = "noms_de_variables";
    break;

case "Opérations avec des variables":
    nom_page = "operations_avec_des_variables";
    break;

case "Opérations sur les entiers":
    nom_page = "operations_sur_les_entiers";
    break;

case "Opérations sur les flottants":
    nom_page = "operations_sur_les_flottants";
    break;

case "Opérations sur les nombres":
    nom_page = "operations_sur_les_nombres";
    break;
```

```

    case "Placer des commentaires":
        nom_page = "placer_des_commentaires";
        break;

    case "Types de données en Python":
        nom_page = "types_de_donnees_en_python";
        break;

    case "Utilisation d'un compteur":
        nom_page = "utilisation_d_un_compteur_";
        break;

    }
    return nom_page;
}

// end
// init ForceDirected

function init(){
    // init data
    var fd = new $jit.ForceDirected({
        // id of the visualization container
        injectInto: 'infovis',
        // Enable zooming and panning
        // with scrolling and DnD
        Navigation: {
            enable: true,
            // Enable panning events only if we're dragging the empty
            // canvas (and not a node).
            panning: 'avoid nodes',
            zooming: 50 // zoom speed. 10 higher is more sensible
        },
        // Change node and edge styles such as

```

```

// color and width.
// These properties are also set per node
// with dollar prefixed data-properties in the
// JSON structure.
Node: {
  overridable: true,
  dim: 7
},
Edge: {
  type: 'arrow',
  overridable: true,
  color: '#23A4FF',
  lineWidth: 0.4
},
  // Add Tips
Tips: {
  enable: true,
  enableForEdges: true,
  onShow: function(tip, node) {
    // count connections
    var count = 0;
    var nom_png = convert(node.name);

    node.eachAdjacency(function() { count++; });
    // display node info in tooltip
    tip.innerHTML = "<div class=\"tip-title\">" + node.name + "</div>"
      + "<div class=\"tip-text\"><b>connections:</b>" + count + "</div>"
      + "<img src='../units/png/' + nom_png + ".png' width='400' height='400'/>" ;
  }
},
// Add node events
Events: {
  enable: true,
  enableForEdges: true,
  type: 'Native',
  // Change cursor style when hovering a node

```

```

onMouseEnter: function() {
  fd.canvas.getElement().style.cursor = 'move';

},
  onMouseWheel: function(delta,e){
    //if(delta==1) alert("positif");
  },
onMouseLeave: function() {
  fd.canvas.getElement().style.cursor = "";
},
// Update node positions when dragged
onDragMove: function(node, eventInfo, e) {
  var pos = eventInfo.getPos();
  node.pos.setc(pos.x, pos.y);
  // fd.animate();
  fd.plot();
},
// Implement the same handler for touchscreens
onTouchMove: function(node, eventInfo, e) {
  $jit.util.event.stop(e); // stop default touchmove event
  this.onDragMove(node, eventInfo, e);
}
},
// Number of iterations for the FD algorithm
iterations: 200,
// Edge length
levelDistance: 70,
// This method is only triggered
// on label creation and only for DOM labels (not native canvas ones).
onCreateLabel: function(domElement, node){
  // Create a 'name' and 'close' buttons and add them
  // to the main node label
  fd.graph.eachNode(function(n){
    n.eachAdjacency(function(adj) {
      adj.setData('alpha', 1, 'end');

```

```
});});
```

```
var nameContainer = document.createElement('span'),
    closeButton = document.createElement('span'),
    plusButton = document.createElement('span'),
    style = nameContainer.style;
nameContainer.className = 'name';
nameContainer.innerHTML = node.name;
closeButton.className = 'close';
closeButton.innerHTML = 'x';
plusButton.className = 'plus';
plusButton.innerHTML = '+';
```

```
domElement.appendChild(nameContainer);
domElement.appendChild(closeButton);
domElement.appendChild(plusButton);
style.fontSize = "0.8em";
style.visibility = "hidden";
style.color = "#ddd";
//style.color = "black";
// Fade the node and its connections when
// clicking the close button
```

```
closeButton.onclick = function() {
    //node.setData('alpha', 0, 'end');
    style.visibility = "visible";
    node.eachAdjacency(function(adj) {
        adj.setData('alpha', 0, 'end');
    });
    fd.fx.animate({
        modes: ['node-property:alpha',
            'edge-property:alpha'],
        duration: 500
```

```

});
};

plusButton.onclick = function() {
    style.visibility = "visible";
    //node.setData('alpha', 1, 'end');
    node.eachAdjacency(function(adj) {
        adj.setData('alpha', 1, 'end');
    });
    fd.fx.animate({
        modes: ['node-property:alpha',
            'edge-property:alpha'],
        duration: 500
    });
    /*if (cmj.jit[node.id][0].adjacencies.length > 0) {
        fd.op.sum(cmj.jit[node.id], {
            type: "fade:con",
            //      hideLabels: false,
            duration: 1000
        });
    }*/
}

```

```

// Toggle a node selection when clicking
// its name. This is done by animating some
// node styles like its dimension and the color
// and lineWidth of its adjacencies.

```

```

    domElement.ondblclick = function(){
        fd.graph.eachNode(function(n){
            /*var nom_page = "";

            switch(node.name){

                case "Afficher":

```

```

        nom_page="afficher";
        break;

    case "Généralités sur Python" :
        nom_page = "generalites_sur_python"
        break;

    case "Encodage":

        nom_page="encodage";
        break;

    case "Instruction if":
        nom_page = "instruction_if"
        break;

}*/
var nom_page = convert(node.name);

```

```

window.open("../units/"+nom_page+".html","titre","width=600,height=480");
        var v = fd.toJSON();
        var str = JSON.stringify(v);

        // $.noConflict();
        // if(n.seletced && node.selected

        // n.selected){

        if(n.selected){
        // alert(str);
        $.ajax({
            type: "POST",
            url: "ressources/track",

```

==

```

);
container').html(str);

data: "name="+node.name,
// datatype: "String",

success: function(msg){
alert( "Data Saved: " + msg

//$('#right-
}

});

}

});};
nameContainer.onclick = function() {
// set final styles
fd.graph.eachNode(function(n) {
if(n.id != node.id) delete n.selected;
n.setData('dim', 7, 'end');
n.eachAdjacency(function(adj) {

adj.setDataset('end', {
lineWidth: 0.4,
color: '#23a4ff'
});
});
});
if(!node.selected) {
node.selected = true;
node.setData('dim', 17, 'end');
node.eachAdjacency(function(adj) {
adj.setDataset('end', {
lineWidth: 3,
color: '#36acfb'

```

```

    });
  });
} else {
  delete node.selected;
}
// trigger animation to final styles
fd.fx.animate({
  modes: ['node-property:dim',
    'edge-property:lineWidth:color'],
  duration: 500
});
// Build the right column relations list.
// This is done by traversing the clicked node connections.
var html = "<h4>" + node.name + "</h4><b> Dépendances:</b><ul><li>",
  list = [];
node.eachAdjacency(function(adj){
  if(adj.getData('alpha')) list.push(adj.nodeTo.name);
});
// append connections information
$jjit.id('inner-details').innerHTML = html + list.join("</li><li>") + "</li></ul>";
};

},
// Change node styles when DOM labels are placed
// or moved.
onPlaceLabel: function(domElement, node){
  var style = domElement.style;
  var left = parseInt(style.left);
  var top = parseInt(style.top);
  var w = domElement.offsetWidth;
  style.left = (left - w / 2) + 'px';
  style.top = (top + 10) + 'px';
  style.display = "";

```

```

    }
  });

  // load JSON data.
  fd.loadJSON(json);
  // sauvegarde du graphe
  /*
    * var sauv = fd.toJSON(); alert(JSON.stringify(sauv));
    */

  // compute positions incrementally and animate.
  fd.computeIncremental({
    iter: 40,
    property: 'end',
    onStep: function(perc){
      Log.write('Chargement en cours : '+perc + '% ...');
    },

    onComplete: function(){
      Log.write('terminé');
      fd.animate({
        modes: ['linear'],
        transition: $jit.Trans.Elastic.easeOut,
        duration: 2500
      });
    }
  });
  // Fin = new Date();
  //document.write((Fin-Debut)+ " ms d'execution javascript");
  $.ajax({
    type: "POST",
    url: "ressources/history",
    data: "name="+node.name,
    // datatype: "String",

    success: function(msg){
      alert( "Data Saved: " + msg );
      $('#right-container').html(str);
    }
  });

```

```

    }

    });
}

});
/*
    * fd.computeIncremental({ iter: 20, property: ['end', 'start', 'current'],
    * onStep: function(perc) { Log.write("loading " + perc + "%"); },
    * onComplete: function() { Log.write("done");
    *
    * //fd.animate(); fd.plot(); } });
    */
// end
/*
* var c = document.getElementById('infovis'); var data = c.getDataURL() ;
*/
$("#colorchm").click(function(n){

// adj.data.$color = "#ccc";
/*node.eachAdjacency(function(adj) {
    adj.data.$color = "red";//#D5A5FF";
    adj.data.$lineWidth = 3;

});*/
n.eachAdjacency(function(adj) {

    adj.setDataset('end', {
    lineWidth: 0.4,
    color: '#a015f'
    });
});
Log.write("oups");
});

```



```

</owl:Class>
<owl:Class rdf:ID="Chapter">
  <owl:equivalentClass>
    <owl:Restriction>
      <owl:onProperty>
        <owl:ObjectProperty rdf:about="#contains"/>
      </owl:onProperty>
      <owl:someValuesFrom>
        <owl:Class rdf:about="#Exercise"/>
      </owl:someValuesFrom>
    </owl:Restriction>
  </owl:equivalentClass>
</owl:Class>
<owl:Class rdf:about="#Exercise">
  <rdfs:subClassOf rdf:resource="#Macro-trajectory"/>
</owl:Class>
<owl:Class rdf:about="#Epistemic_trajectory">
  <owl:equivalentClass>
    <owl:Class>
      <owl:intersectionOf rdf:parseType="Collection">
        <owl:Restriction>
          <owl:someValuesFrom>
            <owl:Class rdf:ID="Episteme"/>
          </owl:someValuesFrom>
          <owl:onProperty>
            <owl:ObjectProperty rdf:about="#contains"/>
          </owl:onProperty>
        </owl:Restriction>
        <owl:Restriction>
          <owl:someValuesFrom rdf:resource="#Episteme_interaction"/>
          <owl:onProperty>
            <owl:ObjectProperty rdf:about="#contains"/>
          </owl:onProperty>
        </owl:Restriction>
      </owl:intersectionOf>
    </owl:Class>
  </owl:equivalentClass>
  <rdfs:subClassOf rdf:resource="#Trajectory"/>
</owl:Class>
<owl:ObjectProperty rdf:ID="is_prerequisite"/>
<owl:ObjectProperty rdf:ID="has_prerequisite">
  <rdfs:range rdf:resource="#Episteme"/>
  <owl:inverseOf rdf:resource="#is_prerequisite"/>
  <rdfs:domain>
    <owl:Class>
      <owl:unionOf rdf:parseType="Collection">
        <owl:Restriction>
          <owl:onProperty rdf:resource="#has_prerequisite"/>
          <owl:someValuesFrom rdf:resource="#Episteme"/>
        </owl:Restriction>
      </owl:unionOf>
    </owl:Class>
  </rdfs:domain>

```

```

    <owl:Class rdf:about="#Exercise"/>
  </owl:unionOf>
</owl:Class>
</rdfs:domain>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="validated_by">
  <owl:inverseOf>
    <owl:ObjectProperty rdf:ID="validates"/>
  </owl:inverseOf>
  <rdfs:subPropertyOf>
    <rdf:Property rdf:about="http://www.w3.org/2002/07/owl#topObjectProperty"/>
  </rdfs:subPropertyOf>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:about="#validates">
  <rdfs:range>
    <owl:Restriction>
      <owl:onProperty rdf:resource="#validates"/>
      <owl:someValuesFrom rdf:resource="#Episteme"/>
    </owl:Restriction>
  </rdfs:range>
  <rdfs:domain rdf:parseType="Resource">
    <owl:onClass rdf:resource="#Student"/>
    <owl:onProperty rdf:resource="#has_prerequisite"/>
    <owl:qualifiedCardinality
rdf:datatype="http://www.w3.org/2001/XMLSchema#nonNegativeInteger"
      >1</owl:qualifiedCardinality>
  </rdfs:domain>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="hasAsTrajectory">
  <rdfs:domain rdf:resource="#Student"/>
  <rdfs:range rdf:resource="#Trajectory"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="hasAsTabs">
  <rdfs:domain rdf:resource="#Micro-trajectory"/>
  <rdfs:range rdf:resource="#Tab_interaction"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:about="#contains">
  <rdfs:subPropertyOf rdf:resource="http://www.w3.org/2002/07/owl#topObjectProperty"/>
</owl:ObjectProperty>
<owl:ObjectProperty rdf:ID="hasAsValidatedExercises">
  <rdfs:range rdf:resource="#Exercise"/>
  <rdfs:domain rdf:resource="#Macro-trajectory"/>
</owl:ObjectProperty>
<owl:NamedIndividual rdf:ID="modifier_une_liste">
  <rdf:type rdf:resource="#Episteme"/>
</owl:NamedIndividual>
<owl:NamedIndividual rdf:ID="constantes_flottantes">
  <rdf:type rdf:resource="#Episteme"/>
</owl:NamedIndividual>
<owl:NamedIndividual rdf:ID="tabTask">

```

```

    <rdf:type rdf:resource="#Tab_interaction"/>
</owl:NamedIndividual>
<owl:NamedIndividual rdf:ID="utilisation_d_un_compteur">
  <rdf:type rdf:resource="#Episteme"/>
</owl:NamedIndividual>
<owl:NamedIndividual rdf:ID="variables">
  <rdf:type rdf:resource="#Episteme"/>
</owl:NamedIndividual>
<owl:NamedIndividual rdf:ID="autres_types">
  <rdf:type rdf:resource="#Episteme"/>
</owl:NamedIndividual>
<owl:NamedIndividual rdf:ID="division_entiere">
  <rdf:type rdf:resource="#Episteme"/>
</owl:NamedIndividual>
<owl:NamedIndividual rdf:ID="submitted">
  <rdf:type rdf:resource="#Tab_interaction"/>
</owl:NamedIndividual>
<owl:NamedIndividual rdf:ID="methodes_d_un_objet">
  <rdf:type rdf:resource="#Episteme"/>
</owl:NamedIndividual>
<owl:NamedIndividual rdf:ID="les_fonctions_built_in">
  <rdf:type rdf:resource="#Episteme"/>
</owl:NamedIndividual>
<owl:NamedIndividual rdf:ID="tabActivity">
  <rdf:type rdf:resource="#Tab_interaction"/>
</owl:NamedIndividual>
<owl:NamedIndividual rdf:ID="importation_facile_et_complete">
  <rdf:type rdf:resource="#Episteme"/>
</owl:NamedIndividual>
<owl:NamedIndividual rdf:ID="operations_sur_les_nombres">
  <rdf:type rdf:resource="#Episteme"/>
</owl:NamedIndividual>
<owl:NamedIndividual rdf:ID="instruction_if">
  <rdf:type rdf:resource="#Episteme"/>
</owl:NamedIndividual>
<owl:NamedIndividual rdf:ID="indentation_d_une_instruction_composee_avec_clause">
  <rdf:type rdf:resource="#Episteme"/>
</owl:NamedIndividual>
<owl:NamedIndividual rdf:ID="operateurs_d_egalite">
  <rdf:type rdf:resource="#Episteme"/>
</owl:NamedIndividual>
<owl:NamedIndividual rdf:ID="boucle_for_parcours_d_une_liste_par_indices">
  <rdf:type rdf:resource="#Episteme"/>
</owl:NamedIndividual>
<owl:NamedIndividual rdf:ID="constantes_entieres">
  <rdf:type rdf:resource="#Episteme"/>
</owl:NamedIndividual>
<owl:NamedIndividual rdf:ID="indentation_et_instruction_if">
  <rdf:type rdf:resource="#Episteme"/>
</owl:NamedIndividual>

```

```

<owl:NamedIndividual rdf:ID="affichage_multiple">
  <rdf:type rdf:resource="#Episteme"/>
</owl:NamedIndividual>
<owl:NamedIndividual rdf:ID="python_dans_un_navigateur">
  <rdf:type rdf:resource="#Episteme"/>
</owl:NamedIndividual>
<owl:NamedIndividual rdf:ID="repetier_une_action_n_fois">
  <rdf:type rdf:resource="#Episteme"/>
</owl:NamedIndividual>
<owl:NamedIndividual rdf:ID="le_caractere_espace">
  <rdf:type rdf:resource="#Episteme"/>
</owl:NamedIndividual>
<owl:NamedIndividual rdf:ID="types_de_base">
  <rdf:type rdf:resource="#Episteme"/>
</owl:NamedIndividual>
<owl:NamedIndividual rdf:ID="generalites_sur_python">
  <rdf:type rdf:resource="#Episteme"/>
</owl:NamedIndividual>
<owl:NamedIndividual rdf:ID="afficher">
  <rdf:type rdf:resource="#Episteme"/>
</owl:NamedIndividual>
<owl:NamedIndividual rdf:ID="les_entrees_via_la_fonction_input">
  <rdf:type rdf:resource="#Episteme"/>
</owl:NamedIndividual>
<owl:NamedIndividual rdf:ID="operateurs_and__or_et_not">
  <rdf:type rdf:resource="#Episteme"/>
</owl:NamedIndividual>
<owl:NamedIndividual rdf:ID="opened">
  <rdf:type rdf:resource="#Tab_interaction"/>
</owl:NamedIndividual>
<owl:NamedIndividual rdf:ID="print_et_saut_de_ligne">
  <rdf:type rdf:resource="#Episteme"/>
</owl:NamedIndividual>
<owl:NamedIndividual rdf:ID="affichage_et_variables">
  <rdf:type rdf:resource="#Episteme"/>
</owl:NamedIndividual>
<owl:NamedIndividual rdf:ID="operations_avec_des_variables">
  <rdf:type rdf:resource="#Episteme"/>
</owl:NamedIndividual>
<owl:NamedIndividual rdf:ID="operateurs_logiques_et__ou_et_non">
  <rdf:type rdf:resource="#Episteme"/>
</owl:NamedIndividual>
<owl:NamedIndividual rdf:ID="creation_d_une_liste_de_longueur_arbitraire">
  <rdf:type rdf:resource="#Episteme"/>
</owl:NamedIndividual>
<owl:NamedIndividual rdf:ID="langage_et_bibliotheques">
  <rdf:type rdf:resource="#Episteme"/>
</owl:NamedIndividual>
<owl:NamedIndividual rdf:ID="comparaison_de_nombres">
  <rdf:type rdf:resource="#Episteme"/>

```

```

</owl:NamedIndividual>
<owl:NamedIndividual rdf:ID="instructions_if_imbriquee">
  <rdf:type rdf:resource="#Episteme"/>
</owl:NamedIndividual>
<owl:NamedIndividual rdf:ID="constantes">
  <rdf:type rdf:resource="#Episteme"/>
</owl:NamedIndividual>
<owl:NamedIndividual rdf:ID="epiSeen">
  <rdf:type rdf:resource="#Episteme_interaction"/>
</owl:NamedIndividual>
<owl:NamedIndividual rdf:ID="le_processus_de_programmation_en_python">
  <rdf:type rdf:resource="#Episteme"/>
</owl:NamedIndividual>
<owl:NamedIndividual rdf:ID="repete_le_contenu_d_une_liste">
  <rdf:type rdf:resource="#Episteme"/>
</owl:NamedIndividual>
<owl:NamedIndividual rdf:ID="instructions_simple_et_composee">
  <rdf:type rdf:resource="#Episteme"/>
</owl:NamedIndividual>
<Student rdf:ID="Student_1">
  <hasAsTrajectory>
    <owl:NamedIndividual rdf:ID="epiGraphDbClick">
      <rdf:type rdf:resource="#Episteme_interaction"/>
    </owl:NamedIndividual>
  </hasAsTrajectory>
  <hasAsTrajectory>
    <owl:NamedIndividual rdf:ID="tabCorrection">
      <rdf:type rdf:resource="#Tab_interaction"/>
    </owl:NamedIndividual>
  </hasAsTrajectory>
  <hasAsTrajectory>
    <owl:NamedIndividual rdf:ID="tabEpistemes">
      <rdf:type rdf:resource="#Tab_interaction"/>
    </owl:NamedIndividual>
  </hasAsTrajectory>
</Student>
<owl:NamedIndividual rdf:ID="conversion_d_une_chaine_en_nombre">
  <rdf:type rdf:resource="#Episteme"/>
</owl:NamedIndividual>
<owl:NamedIndividual rdf:ID="listes">
  <rdf:type rdf:resource="#Episteme"/>
</owl:NamedIndividual>
<owl:NamedIndividual rdf:ID="trier_une_liste_avec_la_methode_sort">
  <rdf:type rdf:resource="#Episteme"/>
</owl:NamedIndividual>
<owl:NamedIndividual rdf:ID="placer_des_commentaires">
  <rdf:type rdf:resource="#Episteme"/>
</owl:NamedIndividual>
<owl:NamedIndividual rdf:ID="boucle_for_imbriquee">
  <rdf:type rdf:resource="#Episteme"/>

```

```

</owl:NamedIndividual>
<owl:NamedIndividual rdf:ID="expressions_booleennes">
  <rdf:type rdf:resource="#Episteme"/>
</owl:NamedIndividual>
<owl:NamedIndividual rdf:ID="les_types_built_in">
  <rdf:type rdf:resource="#Episteme"/>
</owl:NamedIndividual>
<owl:NamedIndividual rdf:ID="epiGraphClick">
  <rdf:type rdf:resource="#Episteme_interaction"/>
</owl:NamedIndividual>
<owl:NamedIndividual rdf:ID="la_bibliotheque_standard">
  <rdf:type rdf:resource="#Episteme"/>
</owl:NamedIndividual>
<owl:NamedIndividual rdf:ID="helpRequested">
  <rdf:type rdf:resource="#Tab_interaction"/>
</owl:NamedIndividual>
<owl:NamedIndividual rdf:ID="read">
  <rdf:type rdf:resource="#Tab_interaction"/>
</owl:NamedIndividual>
<owl:NamedIndividual rdf:ID="methode_split">
  <rdf:type rdf:resource="#Episteme"/>
</owl:NamedIndividual>
<owl:NamedIndividual rdf:ID="adviceRequested">
  <rdf:type rdf:resource="#Tab_interaction"/>
</owl:NamedIndividual>
<owl:NamedIndividual rdf:ID="le_type_float">
  <rdf:type rdf:resource="#Episteme"/>
</owl:NamedIndividual>
<owl:NamedIndividual rdf:ID="multiples__division_entiere">
  <rdf:type rdf:resource="#Episteme"/>
</owl:NamedIndividual>
<owl:NamedIndividual rdf:ID="boucle_while">
  <rdf:type rdf:resource="#Episteme"/>
</owl:NamedIndividual>
<owl:NamedIndividual rdf:ID="reaffectation_de_variable">
  <rdf:type rdf:resource="#Episteme"/>
</owl:NamedIndividual>
<owl:NamedIndividual rdf:ID="affectation_multiple">
  <rdf:type rdf:resource="#Episteme"/>
</owl:NamedIndividual>
<owl:NamedIndividual rdf:ID="succession_d_entrees">
  <rdf:type rdf:resource="#Episteme"/>
</owl:NamedIndividual>
<owl:NamedIndividual rdf:ID="epiClick">
  <rdf:type rdf:resource="#Episteme_interaction"/>
</owl:NamedIndividual>
<owl:NamedIndividual rdf:ID="boucle_for_et_accumulateur">
  <rdf:type rdf:resource="#Episteme"/>
</owl:NamedIndividual>
<owl:NamedIndividual rdf:ID="boucle_for_et_succession_d_entiers">

```

```

    <rdf:type rdf:resource="#Episteme"/>
</owl:NamedIndividual>
<owl:NamedIndividual rdf:ID="affichage_ameliore">
    <rdf:type rdf:resource="#Episteme"/>
</owl:NamedIndividual>
<owl:NamedIndividual rdf:ID="instruction_if_else">
    <rdf:type rdf:resource="#Episteme"/>
</owl:NamedIndividual>
<owl:NamedIndividual rdf:ID="tabHelp">
    <rdf:type rdf:resource="#Tab_interaction"/>
</owl:NamedIndividual>
<owl:NamedIndividual rdf:ID="instruction_if_elif">
    <rdf:type rdf:resource="#Episteme"/>
</owl:NamedIndividual>
<owl:NamedIndividual rdf:ID="les_booleens">
    <rdf:type rdf:resource="#Episteme"/>
</owl:NamedIndividual>
<owl:NamedIndividual rdf:ID="affectation_multiple_et_echange_de_variables">
    <rdf:type rdf:resource="#Episteme"/>
</owl:NamedIndividual>
<owl:NamedIndividual rdf:ID="tabEditor">
    <rdf:type rdf:resource="#Tab_interaction"/>
</owl:NamedIndividual>
<owl:NamedIndividual rdf:ID="instruction_if_dans_une_boucle_for">
    <rdf:type rdf:resource="#Episteme"/>
</owl:NamedIndividual>
<owl:NamedIndividual rdf:ID="afficher_sur_une_seule_ligne">
    <rdf:type rdf:resource="#Episteme"/>
</owl:NamedIndividual>
<owl:NamedIndividual rdf:ID="boucle_for___modification_d_une_liste">
    <rdf:type rdf:resource="#Episteme"/>
</owl:NamedIndividual>
<owl:NamedIndividual rdf:ID="variables_et_typage">
    <rdf:type rdf:resource="#Episteme"/>
</owl:NamedIndividual>
<owl:NamedIndividual rdf:ID="affichage_et_passage_a_la_ligne">
    <rdf:type rdf:resource="#Episteme"/>
</owl:NamedIndividual>
<owl:NamedIndividual rdf:ID="les_chaines___presentation">
    <rdf:type rdf:resource="#Episteme"/>
</owl:NamedIndividual>
<owl:NamedIndividual rdf:ID="quelques_objets_du_module_math">
    <rdf:type rdf:resource="#Episteme"/>
</owl:NamedIndividual>
<owl:NamedIndividual rdf:ID="instruction_if_elif_else">
    <rdf:type rdf:resource="#Episteme"/>
</owl:NamedIndividual>
<owl:NamedIndividual rdf:ID="notion_de_module">
    <rdf:type rdf:resource="#Episteme"/>
</owl:NamedIndividual>

```

```

<owl:NamedIndividual rdf:ID="afficher_un_message">
  <rdf:type rdf:resource="#Episteme"/>
</owl:NamedIndividual>
</rdf:RDF>

```

<!-- Created with Protege (with OWL Plugin 3.4, Build 533) <http://protege.stanford.edu> -->

VI. Fichiers PHP du prototype HiPPY

1. Extraits du Fichier Episteme.inc.php

a. La fonction d'individualisation

```

public static function getJitDataFromEpis($aAll, $sRootName = "")
{
  $aAllEpis = array();

  if ($sRootName != "") {
    $aAllEpis[0] = array(
      'id' => -1,
      'name' => '<span title="Il ne s\'agit pas d\'une notion">'. $sRootName. '</span>',
      'data' => array(
        'color' => 'lightblue',
        'type' => 'circle',
      ),
      'adjacencies' => array()
    );
  }
  $aAllSucc = array();
  foreach ($aAll as $sepi)
  {
    $sepiData = array(
      'id' => $sepi->id,
      'name' => ($sepi->bValidated?"<br/>:". $sepi->sTitle. "</span>",
      'data' => array(
        'color' => $sepi->bValidated?'lightgreen':'white',
        'type' => 'circle',
      ),
      'adjacencies' => array()
    );
    foreach ($sepi->aPrecDirect as $succ)
    {
      $aAllSucc[] = $succ->id;
      $sepiData['adjacencies'][] = array(
        'nodeFrom' => $sepi->id,
        'nodeTo' => $succ->id,
        'data' => array(),
      );
    }
    if (isset($sepi->bDirect))

```

```

    {
      if ($epi->bDirect)
        $aAllEpi[0]['adjacencies'][] = array(
          'nodeFrom' => -1,
          'nodeTo' => $epi->id,
          'data' => array(),
        );
    }
    $aAllEpi[] = $epiData;
  }
  foreach ($aAll as $epi) {
    if (!in_array($epi->id, $aAllSucc)) {
      $aAllEpi[0]['adjacencies'][] = array(
        'nodeFrom' => -1,
        'nodeTo' => $epi->id,
        'data' => array(),
      );
    }
  }
}

return $aAllEpi;
}

```

b. La fonction de construction de la fermeture transitive

public static function setClosureDAG()

```

{
  // Src => Dst : src is a pre-requisite for dst
  sql::query("DELETE FROM epistememes_relations WHERE
sType=''.self::LINK_DISTANT.'");
  $query = "
  SELECT idEpiSrc, idEpiDst
  FROM epistememes_relations
  WHERE sType = ''.Episteme::LINK_DIRECT.' ";
  $aDirect = array();
  foreach(sql::queryFetchArrayFree($query) as $rel)
  {
    $aDirect[$rel->idEpiSrc][] = $rel->idEpiDst;
  }
  $query = "SELECT MAX(id) AS idMax FROM epistememes";
  $res = sql::queryFetchArrayFree($query);
  $iMax = $res[0]->idMax;

  for ($i = 1; $i <= $iMax; $i++)
  {
    if (!array_key_exists($i, $aDirect))
      $aDirect[$i] = array();
  }

  $aDistant = $aDirect;
  for ($n2 = 1; $n2 <= $iMax; $n2++)

```

```

    for ($n1 = 1; $n1 <= $iMax; $n1++)
        for ($n3 = 1; $n3 <= $iMax; $n3++)
            if (in_array($n2, $aDistant[$n1]) && in_array($n3, $aDistant[$n2]) &&
!in_array($n3, $aDistant[$n1]))
                $aDistant[$n1][] = $n3;

    foreach ($aDistant as $src => $aDst)
        foreach ($aDst as $dst)
            if (!in_array($dst, $aDirect[$src]))
                self::addEpiRelation($src, $dst, self::LINK_DISTANT);
}

```

VII. Exemple de processus d'analyse en XML (génééré par RapidMiner)

1. Weka Hierarchical Clustering

```

<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<process version="5.3.013">
  <context>
    <input/>
    <output/>
    <macros/>
  </context>
  <operator activated="true" class="process" compatibility="5.3.013" expanded="true"
name="Process">
    <parameter key="logverbosity" value="init"/>
    <parameter key="random_seed" value="2001"/>
    <parameter key="send_mail" value="never"/>
    <parameter key="notification_email" value=""/>
    <parameter key="process_duration_for_mail" value="30"/>
    <parameter key="encoding" value="SYSTEM"/>
    <process expanded="true">
      <operator activated="true" class="read_csv" compatibility="5.3.013" expanded="true"
height="60" name="Read CSV" width="90" x="45" y="75">
        <parameter key="csv_file"
value="C:\Users\stef_2\Downloads\logs_user_epi_290114.csv"/>
        <parameter key="column_separators" value=","/>
        <parameter key="trim_lines" value="false"/>
        <parameter key="use_quotes" value="true"/>
        <parameter key="quotes_character" value="&quot;"/>
        <parameter key="escape_character" value="\"/>
        <parameter key="skip_comments" value="false"/>
        <parameter key="comment_characters" value="#/>
        <parameter key="parse_numbers" value="true"/>
        <parameter key="decimal_character" value="."/>
        <parameter key="grouped_digits" value="false"/>
        <parameter key="grouping_character" value=""/>
        <parameter key="date_format" value=""/>

```

```

<parameter key="first_row_as_names" value="false"/>
<list key="annotations">
  <parameter key="0" value="Name"/>
</list>
<parameter key="time_zone" value="SYSTEM"/>
<parameter key="locale" value="English (United States)"/>
<parameter key="encoding" value="windows-1252"/>
<list key="data_set_meta_data_information">
  <parameter key="0" value="idUser.true.integer.attribute"/>
  <parameter key="1" value="NbEpi.true.integer.attribute"/>
  <parameter key="2" value="nbSolved.true.integer.attribute"/>
  <parameter key="3" value="sLastLoginDate.true.polynomial.attribute"/>
  <parameter key="4" value="sRegistrationDate.true.polynomial.attribute"/>
  <parameter key="5" value="sType.true.polynomial.attribute"/>
  <parameter key="6" value="sContextType.true.binominal.attribute"/>
  <parameter key="7" value="idContext.true.integer.attribute"/>
</list>
<parameter key="read_not_matching_values_as_missings" value="true"/>
<parameter key="datamanagement" value="double_array"/>
</operator>
<operator activated="true" class="select_attributes" compatibility="5.3.013"
expanded="true" height="76" name="Select Attributes" width="90" x="112" y="255">
  <parameter key="attribute_filter_type" value="subset"/>
  <parameter key="attribute" value=""/>
  <parameter key="attributes" value="|nbSolved|NbEpi"/>
  <parameter key="use_except_expression" value="false"/>
  <parameter key="value_type" value="attribute_value"/>
  <parameter key="use_value_type_exception" value="false"/>
  <parameter key="except_value_type" value="time"/>
  <parameter key="block_type" value="attribute_block"/>
  <parameter key="use_block_type_exception" value="false"/>
  <parameter key="except_block_type" value="value_matrix_row_start"/>
  <parameter key="invert_selection" value="false"/>
  <parameter key="include_special_attributes" value="false"/>
</operator>
<operator activated="true" class="weka:W-HierarchicalClusterer"
compatibility="5.3.001" expanded="true" height="76" name="W-HierarchicalClusterer"
width="90" x="411" y="133">
  <parameter key="D" value="false"/>
  <parameter key="B" value="false"/>
  <parameter key="N" value="4.0"/>
  <parameter key="P" value="false"/>
  <parameter key="L" value="SINGLE"/>
  <parameter key="A" value="weka.core.EuclideanDistance -R first-last"/>
  <parameter key="add_as_label" value="false"/>
  <parameter key="remove_unlabeled" value="false"/>
</operator>
<connect from_op="Read CSV" from_port="output" to_op="Select Attributes"
to_port="example set input"/>

```

```

    <connect from_op="Select Attributes" from_port="example set output" to_op="W-
HierarchicalClusterer" to_port="example set"/>
    <connect from_op="W-HierarchicalClusterer" from_port="cluster model" to_port="result
2"/>
    <connect from_op="W-HierarchicalClusterer" from_port="clustered set" to_port="result
1"/>
    <portSpacing port="source_input 1" spacing="0"/>
    <portSpacing port="sink_result 1" spacing="0"/>
    <portSpacing port="sink_result 2" spacing="0"/>
  </process>
</operator>
</process>

```

VIII. Questionnaire

Le questionnaire est toujours visible sur

https://docs.google.com/forms/d/1gepFwYHdSIItLfnvt6s-fKjvn1Jea76bQu59UuH-T400/viewform?pli=1&edit_requested=true

Questionnaire France-IOI (apprenant)

Ce questionnaire a pour but de mieux connaître les utilisateurs du graphe des notions pour un chapitre ou pour un exercice (parcours lycée) afin de trouver les meilleures voies d'adaptation et de personnalisation. Ce questionnaire sera utilisé par le laboratoire Sciences Techniques Education Formation de l'ENS-Cachan pour évaluer en particulier l'aide qu'apporte l'onglet "Notions associées" en programmation Python et le mettre en correspondance avec votre manière de résoudre les problèmes sur la plateforme France-IOI. Merci d'avance de prendre le temps de répondre le plus précisément possible.

*Obligatoire

Login

Age

Sexe *

- Féminin
- Masculin

Dernière classe fréquentée avant le baccalauréat

(Précisez où vous en êtes ou bien quel fut la filière de sortie : 3^e, CAP, BEP, Terminale...)

Avez-vous obtenu le baccalauréat ? *

- Oui
- Non

Intitulé du dernier diplôme obtenu *

(Certificat d'étude, Brevet des collèges, CAP, BEP, Baccalauréat ou diplôme supérieur)

En quelle année avez-vous obtenu ce dernier diplôme ? *

(Exemple : 2010)

Intitulé du prochain diplôme envisagé

(Brevet des collèges, CAP, BEP, Baccalauréat ou diplôme supérieur)

Votre niveau de maîtrise de la programmation avant d'étudier python sur France-IOI *

Combien de lignes en Python aviez-vous déjà écrit avant de commencer ici ? *

Si vous avez déjà écrit plus de 1000 lignes de code, depuis combien de temps avez-vous arrêté de programmer *

(de façon régulière ou intense)

Sur les deux dernières années, depuis combien de temps vous programmez en Python ? *

- Moins d'un mois
- Entre 1 et 6 mois
- Entre 6 mois et 1 an
- Plus d'un an

En quelle année, avez-vous commencé à programmer ? *

Faites-vous partie d'un groupe ISN en terminale S depuis septembre 2013 ? *

(ISN = option Initiation aux Sciences du Numérique de Terminale S)

- Oui
- Non

Dans un exercice, la rubrique "Notions associées" vous semble *

- Inutile
- Parfois utile
- Utile
- Très utile

Dites pourquoi ? *

Le système de navigation (graphe dynamique) dans les "notions associées" vous semble *

- Non satisfaisant
- Peu satisfaisant

- Satisfaisant
- Très satisfaisant

Dites pourquoi ? *

Quelles améliorations vous souhaitez voir ? *

Le système de navigation dans les "Notions associées" vous permet de voir les notions maîtrisées d'une manière *

- Non satisfaisante
- Peu satisfaisante
- Satisfaisante
- Très satisfaisante

Quel élément vous appréciez le plus dans les "Notions associées" ? *

- L'animation et le centrage de notions
- L'affichage du bilan des notions maîtrisées
- Les rappels sur une notion
- L'affichage des notions prérequis pour un exercice

Pourquoi consultez-vous les "Notions associées" d'un exercice ? *

- Pour vérifier si telle notion est liée à telle autre
- Pour voir les notions maîtrisées
- Pour voir les prérequis d'un exercice
- Pour voir des rappels sur une notion
- Autre :

Résumé

Les travaux de recherche de cette thèse concernent principalement l'aide à la navigation individualisée dans un hypermédia épistémique. Nous disposons d'un certain nombre de ressources qui peut se formaliser à l'aide d'un graphe acyclique orienté (DAG) : le graphe des épistèmes. Après avoir cerné les environnements de ressources et de parcours, les modalités de visualisation et de navigation, de traçage, d'adaptation et de fouille de données, nous avons présenté une approche consistant à corréliser les activités de conception ou d'édition à celles dédiées à l'utilisation et la navigation dans les ressources. Cette approche a pour objectif de fournir des mécanismes d'individualisation de la navigation dans un environnement qui se veut évolutif. Nous avons alors construit des prototypes appropriés pour mettre à l'épreuve le graphe des épistèmes. L'un de ces prototypes a été intégré à une plateforme existante. Cet hypermédia épistémique baptisé HiPPY propose des ressources et des parcours portant sur l'apprentissage du langage Python. Il s'appuie sur un graphe des épistèmes, une navigation dynamique et un bilan de connaissances personnalisé. Ce prototype a fait l'objet d'une expérimentation qui nous a donné la possibilité d'évaluer les principes introduits et d'analyser certains usages.

Mots-clefs : Hypermédia adaptatif, navigation, visualisation, ressources, parcours, traces, fouille de données en éducation, Python.

Abstract

This research work mainly concerns means of assistance in individualized navigation through an epistemic hypermedia. We have a number of resources that can be formalized by a directed acyclic graph (DAG) called the graph of epistemes. After identifying resources and pathways environments, methods of visualization and navigation, tracking, adaptation and data mining, we presented an approach correlating activities of design or editing with those dedicated to resources' use and navigation. This provides ways of navigation's individualization in an environment which aims to be evolutive. Then, we built prototypes to test the graph of epistemes. One of these prototypes was integrated into an existing platform. This epistemic hypermedia called HiPPY provides resources and pathways on Python language. It is based on a graph of epistemes, a dynamic navigation and a personalized knowledge diagnosis. This prototype, which was experimented, gave us the opportunity to evaluate the introduced principles and analyze certain uses.

Keywords : Adaptive hypermedia, navigation, visualization, resources, pathways, traces, educational data mining, Python.