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

Mahdi Miled

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

Mahdi Miled
2ème année de doctorat
UMR Science Technique Education Formation
ENS Cachan – ENS Lyon (IFE) – UniverSud
61, avenue du Président Wilson F-94235 Cachan cedex
mahdi.miled@ens-cachan.fr

Résumé

Cet article suggère d'apporter des mécanismes d'aide à la navigation dans les ressources d'apprentissage et d'aide à leur conception et à leur édition. Nous souhaitons instaurer des éléments de liaison entre ces activités de conception ou d'édition avec celles de la navigation. Dans une ressource que nous concevons comme un graphe d'unités d'apprentissage, nous proposons de guider l'apprenant sans le contraindre et d'informer le concepteur sur l'usage à travers les traces de navigation. Dans le but de faciliter la navigation dans ce graphe de ressources et son édition, nous développons un prototype intégrant un outil de visualisation interactive. Grâce aux résultats des simulations effectuées sur cet outil, nous avons pu déterminer quel type de visualisation était le plus approprié pour le type de graphe et la situation d'apprentissage envisagée. Ces techniques nous semblent essentielles pour soutenir les apprenants dans leur guidage sur leur parcours d'apprentissage car elles permettent de bien discerner les dépendances dans les grands graphes évolutifs.

Introduction

Considérant la construction d'une ressource pour l'apprentissage du langage Python en autonomie (Ortiz 2012), nous sommes confrontés à la conception d'un grand nombre de parties constituant cette ressource, à leur organisation et à leur évolution à travers l'usage qu'en feront les apprenants.

La question de la conception et de l'édition des ressources numériques d'apprentissage a été déjà abordée (Broisin et Vidal 2005). Dans le contexte participatif et collaboratif du web 2.0, les utilisateurs ne sont plus seulement consommateurs de contenus mais ils en sont également producteurs, contributeurs et annoteurs. Nous utilisons ici l'idée que le parcours des apprenants est une information utilisable par les concepteurs pour améliorer la ressource et le guidage des apprenants. Nos travaux s'inscrivent d'une part dans l'aide à la conception et la modification d'une ressource de haut niveau de granularité (cours, manuels de référence), et d'autre part dans l'aide à la navigation dans un vaste réseau de parties homogènes et interdépendantes au sein d'une telle ressource. La première partie de cette communication présente l'articulation entre les activités d'édition et de navigation. Tout en montrant notre positionnement, nous passons en revue dans une deuxième partie, les environnements permettant d'utiliser des ressources d'apprentissage, les techniques de visualisation interactive des graphes ainsi que quelques fondements sur les traces. La troisième propose un modèle pour l'articulation entre édition et parcours. La quatrième décrit la méthode de mise en œuvre d'un outil de visualisation de graphes pour aider les apprenants dans leur navigation. La cinquième partie fixera quelques éléments d'évaluation avant la conclusion.

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 el al. 2005) (Wiley et Gurell 2009). D'autres travaux se sont penchés sur l'accès et la navigation dans les ressources (Nash 2005) (Wolpers et al. 2008). Cependant peu s'intéressent à la relation entre ces activités d'édition et de navigation qui ont des objectifs différents, faisant intervenir concepteurs et apprenants.

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 pré-requis. Ce graphe 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 pré-requis dans la représentation destinée à l'apprenant (parfois débutant) pour lui en simplifier la lecture. 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. Nous souhaitons cependant offrir un système qui permet l'évolution d'un ensemble vivant de ressources prenant en compte l'expérience de l'apprenant.

Pourquoi lier édition et navigation ?

Il peut paraître peu conventionnel de vouloir créer une relation entre ces activités à 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. 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.

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.

Regard sur l'existant

Il s'agira dans cette section de se positionner par rapport aux environnements d'accès aux ressources dont certains permettent l'édition selon un cycle de vie spécifique. Nous nous intéresserons aussi aux mécanismes d'accès et de navigation et notamment à leurs interfaces de visualisation. Enfin, nous expliciterons la nécessité des traces dans la relation entre navigation et édition.

Environnements d'accès aux ressources

La plupart des environnements liés à l'utilisation des objets d'apprentissage[1] et à leur co-édition (Connexions, web) permettent de partager les ressources et de suivre leur évolution selon un cycle de vie caractéristique. Ces environnements peuvent prendre la forme de plateformes dédiées à un domaine spécifique comme l'apprentissage du langage Python (PythonLearn, web), ou de plateformes d'apprentissage en ligne LMS[2] de type Moodle, Blackboard, Claroline ou même encore de dépôts de ressources tels que ARIADNE (Klerxk et al. 2010), MERLOT et LORNET. 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.
Notre proposition vise à soutenir l'enseignement et l'apprentissage d'un thème complexe faisant appel à de nombreuses ressources interdépendantes. Nous prévoyons une première implémentation dans le cas d'un environnement dédié à l'apprentissage de la programmation en Python.

Visualisation et navigation dans les graphes

La question de la recherche ou 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 (Hermann et al. 2000). 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.
Pour ce qui est des différentes mises en formes (graph layout), on peut retenir celles qui se basent sur :

- 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)
- ainsi que les arbres à tiroirs (Treemap).

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. 2007) voire sur des questions d'aide à l'accès et à la recherche de ressources d'apprentissage déposées dans des réservoirs

1 Learning objects
2 Learning Management Systems

(Klerkx et al. 2004). Certains travaux soulignent les limites de l'accès et la recherche des ressources par des formulaires (Klerxk et al. 2005) et proposent des stratégies basées sur une visualisation et une navigation interactive. Pour notre part, nous comptons utiliser les types de visualisation ForceDirected, HyperTree et RGraph. Nous estimons que ce genre de visualisation interactive 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.

Traçage des activités d'apprentissage

Une trace est une collection d'observés temporellement situés (Djouad et al. 2010). Dans la littérature des EIAH, le recours aux mécanismes de traces a pour finalité la régulation ou l'auto-régulation de l'apprentissage (Heraud et al. 2005), la modélisation et le profilage de l'apprenant, la personnalisation et l'adaptation de l'apprentissage (Brusilovsky 1999) ou encore la réingénierie des EIAH (Choquet et Iksal 2007). Nous souhaitons exploiter les traces issues des activités de navigation et d'utilisation pour des besoins d'édition pour assurer, *in fine*, une meilleure évolutivité à la ressource dans son ensemble.

Modélisation générale

Notre orientation générale (Fig. 1) 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 sont 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.

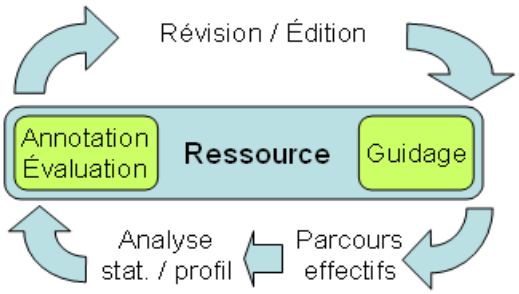

Figure 1 : Relation entre activité d'édition et de navigation

Ils 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…).

Dans notre vision du cycle de la conception et de l'édition d'une unité jusqu'à son intégration dans un parcours, 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 orientations. La gestion des droits se fait selon le modèle proposé par (connexions, web) 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. En ce qui concerne la validation pédagogique d'une unité, nous nous sommes inspirés des états fournis par ADL SCORM (SCORM, web). Sur la figure 2, 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 la valider. Les actions suivantes sont optionnelles et consistent à *soumettre du code*[3] (si l'unité l'indique) et *consulter les composants* de l'unité pour la parcourir en totalité.

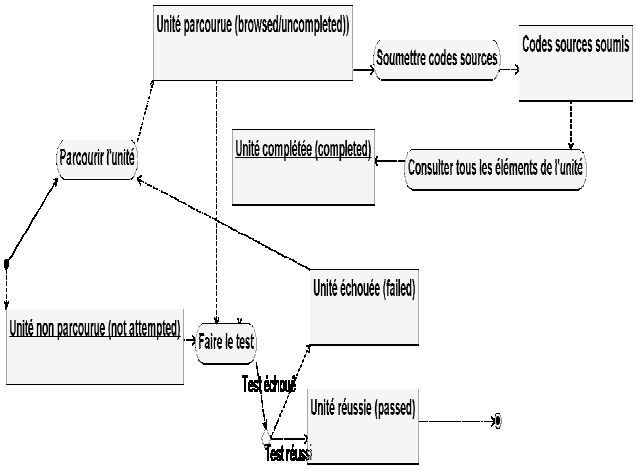

Figure 2 : États d'une unité et sa validation

3 Pour être spécifique à l'apprentissage de la programmation, nous prévoyons d'ajouter un état supplémentaire (code submitted).

Les principaux éléments du modèle ayant été décrits dans cette partie, nous présentons dans la suivante, l'approche méthodologique suivie.

Approche et méthodologie

Dans le but de mettre en œuvre nos éléments de réflexion, nous avons recensé et testé divers outils libres de représentation interactive des graphes. Ces tests nous ont permis de vérifier l'adéquation de ces outils aux types de graphes qui nous concernent. À partir du choix d'outil qui s'est dégagé, nous avons commencé à implémenter un prototype.

Choix d'un outil de visualisation interactive de graphes

L'identification des outils de visualisation des graphes nous a permis de cibler les outils proposant des bibliothèques libres. Nous avons finalement opté pour JIT[4]. Ce choix est dû aux riches types de visualisations et performances vu qu'il s'appuie sur une technologie cliente (Javascript). Il utilise comme entrée un format JSON[5] pour la représentation des données.

Intégration d'un outil et début de prototypage

Tout en commençant à développer un prototype, nous avons intégré des visualisations du graphe des ressources et quelques sous-graphes représentant d'éventuels parcours. Les figures 3, 4 et 5 illustrent les rendus issus de ForceDirected, HyperTree et RGraph.

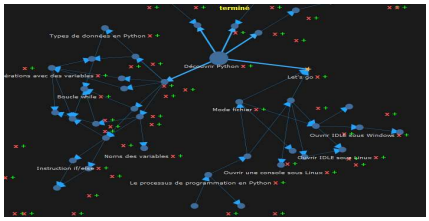

Figure 3 : Visualisation issue de ForceDirected

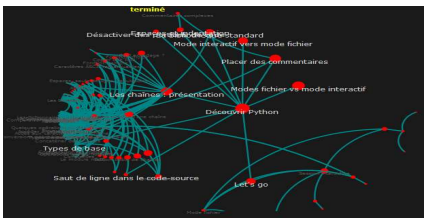

Figure 4 : Visualisation issue de HyperTree

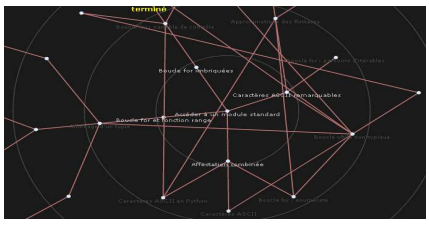

Figure 5 : Visualisation issue de RGraph

Traces liées à l'édition et à la navigation

Le prototype que nous sommes en train de développer sous Zend framework 1.11.3 propose (Fig. 6) de tracer les activités relatives à la navigation dans les ressources et d'autres spécifiques à leur édition. Ces traces sont sauvegardées dans des tables MySQL. En terme 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 Fig. 2). Cette fonction est en cours d'intégration. 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 doivent être incluses. Tous ces éléments paraissent nécessaires pour pouvoir gérer l'évolution de la structure globale du graphe. Ainsi, nous pouvons penser qu'une évaluation d'une unité aura un impact sur l'évolution globale du graphe. Par exemple, si N évaluations globales (comprenant 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 si cela concerne le contenu de l'unité, soit automatiquement s'il s'agit de réordonner un parcours.

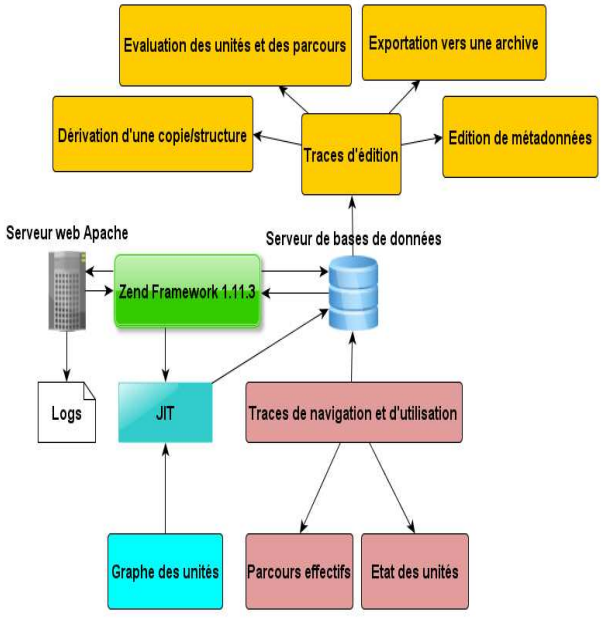

Figure 6 : Processus de génération des traces d'édition et de navigation

Eléments d'évaluation

Nous souhaitons vérifier à présent la validité de certaines fonctionnalités et déterminer quel type de visualisation est le plus adapté à nos besoins.

Eléments d'évaluation analytique

Afin de mesurer quelques indications sur l'utilisabilité, nous utilisons quelques principes basés sur une démarche analytique (Nogry et al. 2004) en nous appuyant sur des check-lists pour vérifier la conformité

[4] Javascript Infovis Toolkit

[5] JavaScript Object Notation

de ce qui est développé et en spécifiant des cas d'utilisation. Le système permet un accès à différents utilisateurs et une interface et des privilèges adaptés à chacun des profils d'utilisation (concepteur, tuteur, étudiant). Selon ses privilèges, chacun peut éditer les métadonnées d'une unité et évaluer certaines ressources. La navigation dans le macro-graphe se fait selon la visualisation HyperTree. Pour ce qui est des cas d'utilisation, nous pouvons imaginer un étudiant souhaitant parcourir certaines unités. Il emprunte pour cela une navigation "libre" en visitant des unités selon son choix. Les événements seront tracés en termes de chemins parcourus. Un autre cas d'utilisation serait de suivre un parcours proposé par un enseignant, et regarder si l'étudiant l'a bien suivi ou s'il a eu recours à d'autres unités.

Comparatif des temps d'exécution

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 Gephi (Gephi, web) 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 a consisté à générer dix graphes aléatoires et à comparer les exécutions pour les trois algorithmes choisis : HyperTree (Fig. 4), ForceDirected (Fig. 3) et RGraph (Fig. 5) 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.

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 une 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. Les résultats comparatifs obtenus (Fig. 7) donnent une estimation de la complexité moyenne opérationnelle et permettent surtout de discerner le comportement de chacun des algorithmes pour de plus grands graphes. Ceci nous a 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). 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. 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. 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 modeler et personnaliser la vue qui s'adapte le mieux à ses préférences.

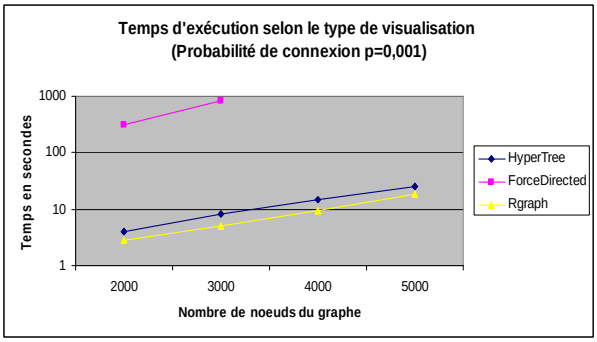

Figure 7 : Temps d'exécution selon le type de visualisation

Conclusion

Nous avons présenté une approche permettant d'établir des liaisons entre les activités de conception ou d'édition des ressources d'apprentissage et les activités de navigation. Cela s'inscrit d'une part dans l'aide à la navigation dans un graphe de ressources interdépendantes notamment à travers la bibliothèque JIT qui propose une navigation dynamique, et de l'autre dans l'aide à la conception et l'édition de ces ressources. Ces activités qui concernent des acteurs différents (concepteurs, tuteurs et apprenants) peuvent être liées pour permettre une évolution du graphe des unités comme ressource globale d'apprentissage. Le prototype qui est en cours de développement mettra en œuvre la validation des unités par les apprenants et le mécanisme de traçage des activités. Nous envisageons d'impliquer les concepteurs et les étudiants dans une démarche participative. Cette démarche participative permet de confronter le prototype avec les usagers finaux. Nous tiendrons compte du retour d'usage sans pour autant nous attarder sur les aspects plus fins d'IHM et d'ergonomie.

Références

Livres

Paquette, G. (2002). *L'ingénierie du télé-apprentissage, pour construire l'apprentissage en réseaux*, Presses de l'Université du Québec, mai 2002, 490 pages.

Articles de Revue

Brusilovsky, P. (1999). Adaptive and Intelligent Technologies for Web-based Education. In C. Rollinger and C. Peylo (eds.), Special Issue on Intelligent Systems and Teleteaching, *Künstliche Intelligenz*, 4, 19-25.

Burgos, D., Arnaud, M., Neuhausser, P., & Koper, R. (2005). IMS Learning Design : la flexibilité pédagogique au service des besoins de l'e-formation. *Revue de l'EPI*.

Choquet, C., et Iksal, S. (2007). Modeling tracks for the model driven reengineering of a tel system. Journal of Interactive Learning Research, vol. 18, p. 161-184.

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. In TSI. 29(6). pp. 721-741.

Nash, S. S. (2005). Learning objects, learning object repositories, and learning theory: Preliminary best practices for online courses. Interdisciplinary Journal of Knowledge and Learning Objects, 1, 217–228.

Pernin, J.-P. (2003). Objets pédagogiques : unités d'apprentissage, activités ou ressources ?, Revue Sciences et Techniques Educatives, Hors série Ressources numériques, XML et éducation, pp 179-210, avril, éditions Hermès.

Wiley, D., & Gurrell, S. (2009). A decade of development… Open Learning: The Journal of Open and Distance Learning, 24(1), 11-21.

Actes de Conférences

Catteau, O., Vidal, P. & Broisin, J. (2007). A 3D Representation of Relationships between Learning Objects. In C. Montgomerie & J. Seale (Eds.), *Proceedings of World Conference on Educational Multimedia, Hypermedia and Telecommunications 2007* (pp. 4262-4271). Chesapeake, VA: AACE.

Heraud J.-M., Marty J.-C., France L., Carron T. (2005). Helping the Interpretation of Web Logs : Application to Learning Scenario Improvement. Workshop AIED'05, Amsterdam.

Herman, I., Melancon, G., & Marshall, M. S. (2000). Graph visualization and navigation in information visualization: A survey. IEEE Transaction on visualization and computer graphics, 6(1), 24–43.

Klerkx, J., Duval, E., & Meire, M. (2004). Using information visualization for accessing learning object repositories. Eighth International Conference on Information Visualisation. April 2004. (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. Proceedings of World Conference on Educational Multimedia, Hypermedia & Telecommunications (p. 4281–4287).

Miled, M. (2011). Les ressources complexes pour l'apprentissage: comment exploiter les traces d'utilisation et soutenir l'édition collaborative? Poster à la Conférence EIAH 2011, Mons, Belgique.

Nogry, S., Jean-Daubias, S. & Ollagnier-Beldame, M. (2004). Évaluation des EIAH: une nécessaire diversité des méthodes. Proceedings of TICE (p. 265–271).

Ortiz, P. (2012). Hypermédia adaptatif à épistèmes pour l'apprentissage des langages de programmation. Poster à RJC EIAH 2012, Amiens.

Wolpers M., Memmel, M., Klerkx, J., Parra, G., Vandeputte, B., Duval, E., Schirru, R., and Niemann, K. (2008). Bridging Repositories to form the MACE Experience, New Review of Information Networking, 14:102–116, Taylor & Francis Group, LLC.

Références sur le web

Connexions - Sharing Knowledge and Building Communities http://cnx.org/

Gephi - Gephi, an open source graph visualization and manipulation software http://gephi.org/

PythonLearn - Self-paced learning Python http://www.py4inf.com

SCORM - Advanced Distributed Learning Initiative http://www.adlnet.org