



Le Web pour enseigner par projets et favoriser la collaboration

Sylvie Ratté, Jocelyne Caron

► **To cite this version:**

Sylvie Ratté, Jocelyne Caron. Le Web pour enseigner par projets et favoriser la collaboration. Revue Internationale des Technologies en Pédagogie Universitaire, Conférence des recteurs et principaux des universités du Québec [CREPUQ], 2004, 1 (2), pp.27-34. edutice-00001380

HAL Id: edutice-00001380

<https://edutice.archives-ouvertes.fr/edutice-00001380>

Submitted on 20 Jan 2006

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.

Sylvie Ratté

École de technologie supérieure
sylvie.ratte@etsmtl.ca

Jocelyne Caron

Université du Québec à Montréal
caron.jocelyne@uqam.ca

Compte rendu de pratique

Résumé

L'utilisation du Web présentée dans cet article vient appuyer une démarche pédagogique fondée sur le partage des savoirs. Dans le contexte d'une simulation industrielle, les étudiants d'un cours de programmation avancée doivent apprendre à partager leur expertise, à communiquer clairement leurs questions, à décrire des problèmes et à faire face à des retournements inattendus. Le Web devient un espace de collaboration pour diffuser les requêtes d'un client fictif, les directives, les travaux préliminaires et les ententes. Les résultats obtenus tendent à suggérer que cette approche globale force l'étudiant à mieux planifier, l'invite à mieux formuler et décrire les problèmes, augmente le niveau de collaboration et facilite la production d'analogies. Un système facilitant la logistique entourant la revue par les pairs (récupération des travaux, construction d'une grille d'évaluation et diffusion des résultats) est également présenté.

Abstract

The use of the Web presented in this article supports a teaching approach based on knowledge sharing. Within the context of an industrial simulation, the students in an advanced programming course must learn to share their expertise, to communicate their questions clearly, to describe problems, and to deal with unexpected reversals. The Web becomes a collaborative space for disseminating the requests of a fictional client, along with directives, preliminary studies, and agreements. The results obtained tend to suggest that this global approach forces the student to plan better, increases the level of cooperation, facilitates the production of analogies, and invites the students to improve how they formulate and describe their problems. A system that facilitates the logistics of peer review (retrieving assignments, constructing an evaluation chart, and disseminating results) is also presented.

Introduction

Entre 1995 et 2001, nous avons élaboré des sites Web associés aux trois cours¹ de programmation offerts par le Service des enseignements généraux de l'École de technologie supérieure (ÉTS). Le but principal était au départ d'offrir à nos étudiants l'ensemble des ressources relatives à chaque cours en un seul espace.

La structure de ces sites n'a rien d'innovateur : chacun contient six grandes sections (Figure 1) qui recouvrent les informations de nature administrative (mémos, aide, liens, résultats) et celles qui concernent les cours proprement dits (théorie, travaux).

Cette structure cache le fait que, parmi toutes les sections présentées, celle qui concerne la théorie est la plus importante. En effet, elle contient présentement plus de 1000 pages de documentation présentées comme des hyperlivres² (Boroni, Goosey, Grinder et Ross, 2001), et couvre plus de 90 % des ressources consacrées au cours : documents thématiques, exercices, illustrations, animations, programmes et résumés. On comprendra aisément que cette organisation générale ne correspond à aucune méthode d'enseignement particulière.

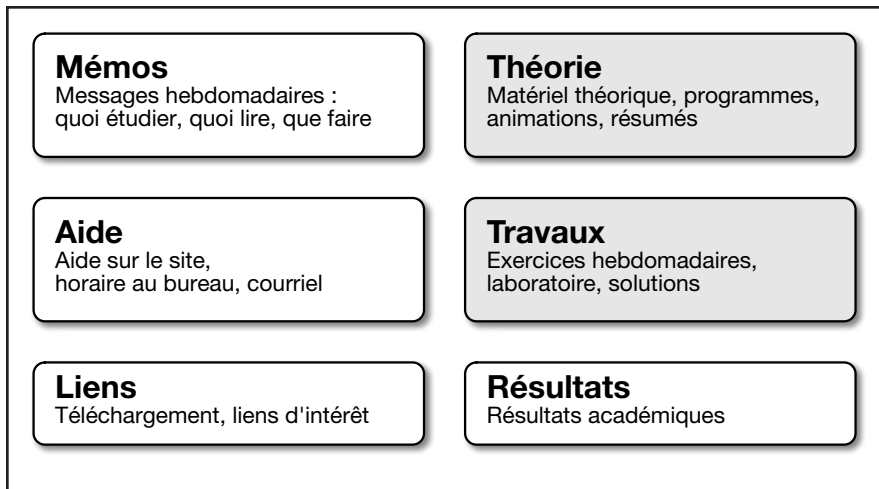


Figure 1. Structure d'un site de base

Les trois sites ont été largement utilisés depuis 1995. Ils sont tellement denses que l'enseignant(e) doit constamment aiguiller les étudiants sur les parties du site à consulter, semaine après semaine. Le contenu de ces sites Web se met à ressembler de plus en plus à un manuel de référence ou à des notes de cours³. De plus, peu importe la quantité de ressources à leur disposition, les étudiants faibles demeurent invariablement faibles et les étudiants forts ne collaborent strictement pas. Alors, comment le Web peut-il réellement contribuer à offrir un apprentissage de qualité tout en favorisant la collaboration? C'est en partie à cette question que le projet décrit dans cet article tente de répondre.

Il faut cependant comprendre que le projet s'inscrit d'abord dans une volonté de combler les lacunes que nous percevons depuis plusieurs années dans notre enseignement et auxquelles nous voulons remédier, car elles contribuent, selon nous, à reproduire un modèle d'expert en programmation qui ne correspond pas aux qualités qu'un professionnel devrait posséder. Nous mentionnons ici les lacunes les plus évidentes observées dans l'apprentissage de l'informatique (et qui pourraient facilement être généralisées à plusieurs autres domaines).

1. Programmation sans planification

Les étudiants n'ont pas appris à se donner d'abord une vue d'ensemble du problème informatique à résoudre. Ils s'attellent à l'ordinateur sans même avoir apprécié la complexité du problème. Ils voient la programmation comme une suite non organisée de « bribes » de code qui vont s'imbriquer les unes dans les autres tôt ou tard.

2. Démarche compétitive sans partage

Lors de la réalisation des travaux pratiques, il est fréquent d'observer que les étudiants plus expérimentés font souvent bande à part. Ils ne partagent leur expertise qu'avec l'enseignant, qu'ils considèrent alors comme un de leurs pairs. Comme ils se trouvent dans un contexte universitaire (académique), ils veulent de bonnes notes et adoptent une attitude de compétition. Cette attitude peut facilement se maintenir en milieu de travail, et elle peut être néfaste. Aujourd'hui, les projets informatiques sont d'une telle envergure en industrie qu'il nous faut favoriser chez les étudiants, dès leur formation, la capacité à collaborer, à reconnaître la valeur du travail des autres, à apprendre et à chercher. Ce constat est lourd de conséquences car il suppose une remise en question des pratiques actuelles en enseignement supérieur.

3. Difficulté à produire des analogies

Peu d'étudiants savent établir un lien direct entre la présentation abstraite d'un concept dans un cours et une réalité physique. Il est également fréquent de constater leur grande difficulté à voir les similitudes entre un problème nouveau et un autre qu'ils ont déjà programmé. Par exemple, dans le projet décrit plus loin, le concept de pile est présenté dans un cours théorique alors que dans le projet de programmation, on fait mention de conteneurs entreposés en rangées ayant la caractéristique de ne rendre accessible que le dernier conteneur entré, ce qui représente clairement une structure de pile. Pourtant, peu d'étudiants établissent immédiatement le lien.

4. Difficulté à formuler, à décrire des problèmes

La grande difficulté de toute solution programmée réside souvent dans la partie préliminaire au codage proprement dit. Le problème et ses ébauches de solution doivent être énoncés clairement, décrits, structurés. On observe que très peu d'étudiants prennent le temps de réfléchir à un problème de manière approfondie, de le décomposer et d'en identifier les aspects complexes. Encore moins prennent-ils le temps de décrire et d'expliquer leurs essais. Ils se rabattent très souvent sur les pistes de solution offertes par l'enseignant, sans poser plus de questions. Ils adoptent très vite une attitude uniforme devant tout problème à résoudre, comme si le fait qu'une méthode ait marché une fois en faisait la méthode universelle.

C'est en ayant en tête ces difficultés que le projet a été développé. Il propose une méthode d'enseignement spécifique et l'utilisation d'un site Web adapté à l'expérience d'apprentissage des étudiants. Le groupe ayant participé à ce projet, réalisé dans le cadre d'un cours de programmation avancée en C++, comprenait 17 équipes de 3 à 5 étudiants chacune.

La méthode d'enseignement ainsi que le type d'utilisation et de gestion du site Web associé sont présentés à la section suivante. La deuxième section est consacrée à la présentation de résultats plus généraux. Une discussion couvrant les principales caractéristiques de l'approche proposée forme le contenu de la troisième section. Puisque le mode d'évaluation par les pairs utilisé dans ce projet est primordial, nous présentons à la quatrième section un système informatique permettant de faciliter la logistique de telles évaluations. Une brève conclusion termine cet exposé.

1. Méthode d'enseignement

1.1 Enseignement par projets

Nous avons donc adopté une méthode d'enseignement construite autour de la réalisation d'un projet d'envergure qui nécessite une connaissance globale de la matière pour être abordé dans son ensemble. Nous avons ainsi conçu le projet autour des quatre lacunes observées (pour des idées similaires, voir Boyle, 2000; Hadjerrout, 1999; Turner et Zachary, 1999), que nous examinons à la lumière des solutions que nous avons choisi d'adopter.

Afin d'amener les étudiants à planifier avant de programmer, aucune formulation précise ni aucune description du problème ne sont fournies pendant le semestre. Par leurs questions et leurs réactions à certaines requêtes, ce sont les étudiants qui définissent le problème. Les nouvelles directives apparaissent donc graduellement sur le site. Cette manière de procéder contribue à créer une atmosphère de coopération et un intérêt soutenu. Ce qui apparaît sur le site est toujours pertinent.

Pour les amener à partager leurs expertises et à coopérer au lieu d'entrer constamment en compétition, trois méthodes sont utilisées. D'abord, chaque travail préliminaire est publié sur le Web. Ensuite, tous les étudiants doivent travailler en équipe. Finalement, chaque équipe doit évaluer le travail des autres équipes lors d'une évaluation par les pairs. Cette manière de faire permet aux équipes plus fortes d'approprier d'autres façons d'envisager des solutions et à ceux en difficulté, de repartir sur le bon pied pour la suite. Encore une fois, le site évolue au gré des travaux publiés, ce qui renforce le sentiment d'avoir pris sur l'apprentissage.

Pour les amener à faire des analogies, la théorie, les exercices et les exemples sont présentés lorsque cela devient nécessaire. Conséquemment, le matériel apparaît sur le Web de manière graduelle. Puisque plusieurs étudiants éprouvent ce type de difficultés, il nous semblait nécessaire de ne pas noyer les étudiants sous les informations qui nécessitent, pour trouver les réponses correctes, de savoir justement faire des analogies. Présenter ainsi les contenus de façon morcelée aide beaucoup les étudiants qui ont de la difficulté à identifier les liens entre les contenus théoriques et leur application.

Finalement, pour les amener à formuler et décrire des problèmes, le projet est construit de telle sorte que les étudiants doivent poser des questions et rechercher des informations s'ils désirent avancer. Parce qu'il n'y a aucune formulation écrite du problème, les étudiants doivent participer à toutes les séances de cours et de laboratoire, car une bonne question pourrait être posée à tout moment. Pour compléter le tout, ils doivent collaborer avec un client fictif envers qui ils sont redevables. Ce dernier peut alors introduire des modifications-surprises, ce qui nous permet de contrer ce que Weber-Wulff (2000) appelle des *code warriors*⁴ et de favoriser ainsi la collaboration. Le site est justement utilisé pour diffuser les demandes de ce client.

De manière concrète, le projet de programmation consistait à simuler une aluminerie. Cette simulation doit cependant être comprise à deux niveaux. D'abord, il s'agissait de développer un programme simulant les opérations dans une aluminerie (voir la Figure 2). Ensuite, la création du programme simule le contexte industriel en exigeant des étudiants qu'ils s'adaptent constamment à de nouvelles requêtes et données.

L'objectif principal était d'évaluer la possibilité d'augmenter la production. Les étudiants devaient ainsi construire leur programme de telle manière que différen-

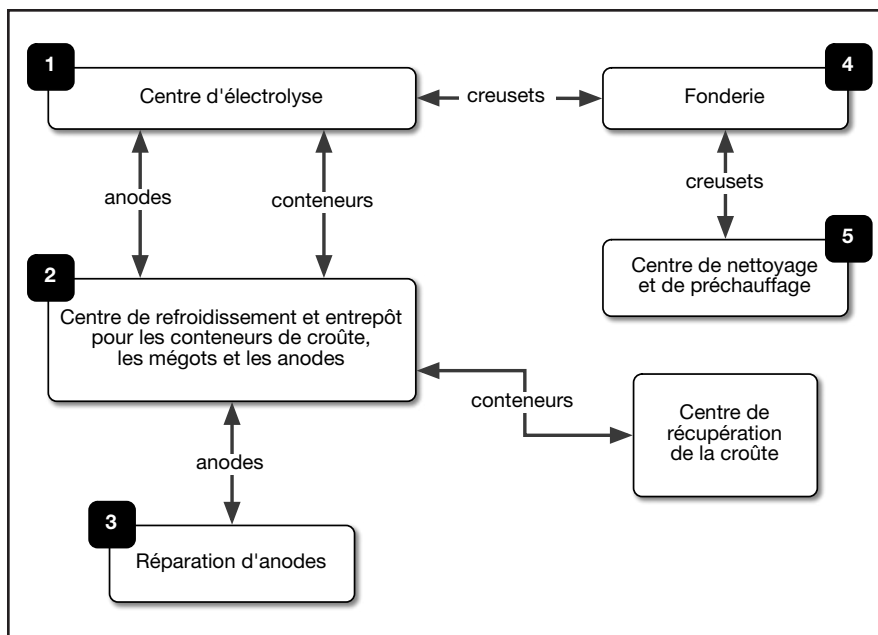


Figure 2. Organisation de l'usine

tes hypothèses puissent être testées (augmentation de l'espace, modifications aux techniques d'entreposage, augmentation du rythme de travail, etc.).

1.2 Utilisation du site Web

Notre expérience d'utilisation de sites Web associés à des cours nous a appris qu'un site où toutes les ressources sur le projet peuvent être récupérées d'un seul coup ne profite qu'à une minorité d'étudiants (en fait, les plus forts). Ainsi, le site du projet propose, en début de semestre, une structure vide organisée autour du concept de projet, et construite autour de deux grands blocs (Figure 3). La section Ressources contient les notions théoriques à étudier et augmente graduellement au cours de la session avec les exemples et les documents présentés en classe, le matériel général demeurant accessible sur le site de base, présenté à la Figure 1. Des exercices liés au projet sont suggérés chaque semaine. La section Projet courant regroupe les informations concernant le programme à réaliser. On y trouve quatre rubriques : Calendrier, Client, Réalisations et Documentation.

La rubrique Documentation est entièrement consacrée aux références fournies par les étudiants sur le projet. Nous conservons ainsi l'enthousiasme des étudiants et aug-

mentons leur implication dans leur propre apprentissage. Pourquoi? Premièrement, le matériel acquiert une nouvelle pertinence : il se trouve contextualisé, directement lié à l'expérience d'apprentissage des étudiants. Deuxièmement, les étudiants ont le net sentiment qu'ils contribuent au développement du cours en fournissant des notes, des corrections ou des ajouts divers.⁵ Troisièmement, les étudiants sentent que l'enseignant travaille autant qu'eux et que le site est vraiment important puisqu'il leur est destiné.

Pour réaliser le projet, les étudiants doivent respecter quatre étapes, apparentées à la réalisation d'un projet professionnel : la documentation, l'ébauche de solutions, le prototype et la livraison.

Documentation.— L'objectif de cette étape est de produire une analyse détaillée de l'usine. Les étudiants doivent donc acquérir la terminologie appropriée afin de parler clairement du projet. Ils doivent aussi comprendre les interactions générales entre les salles de l'usine afin de produire l'analyse orientée objet (OO) correspondante. Lors du premier cours de la session, le client, qui n'est pas un expert en informatique, fait une courte présentation de l'usine et des besoins. Les étudiants sont alors introduits au site du projet et au site

de base associé au cours. Ils inscrivent leur équipe et nomment un responsable des communications. Chaque équipe peut dès lors être contactée par tous ceux et celles qui sont directement impliqués dans la réalisation du projet.

Ébauche de solutions.— Les étudiants utilisent à cette étape les informations obtenues à la suite de la réalisation de l'étape 1, et commencent la construction d'un programme couvrant les interactions entre trois salles essentielles (les salles 1, 2 et 3 de la Figure 2)⁶.

Prototype.— Le prototype de l'étape 3 consiste en un programme relativement complet (incluant la salle 4). Ce programme prend également en compte les nouvelles requêtes du client, entre autres les calculs de consommation électrique et les modifications aux méthodes d'entreposage⁷.

Livraison.— Finalement, la livraison du produit consiste à compléter les interfaces selon les directives du client et à présenter le produit fini à un auditoire d'utilisateurs potentiels ainsi qu'à un groupe de professionnels devant hypothétiquement prendre en charge le programme.

À chaque étape, trois types d'informations sont disponibles : 1) les données du client (présentations orales et requêtes écrites); 2) les notions théoriques, les exemples et les exercices vus en classe et en laboratoire; 3) les résultats obtenus par l'ensemble de la classe lors de l'étape précédente.

Lors de la réalisation de chaque étape spécifique, les étudiants sont également guidés par des périodes de questions/réponses en classe et des discussions par courriel ou autrement. Les étudiants peuvent également communiquer avec le client par courriel lorsque nécessaire.

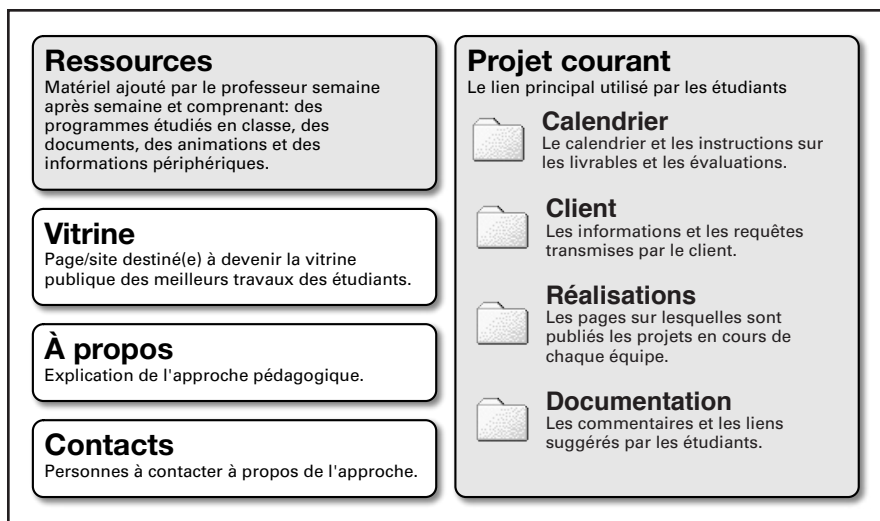


Figure 3. Structure du site du projet

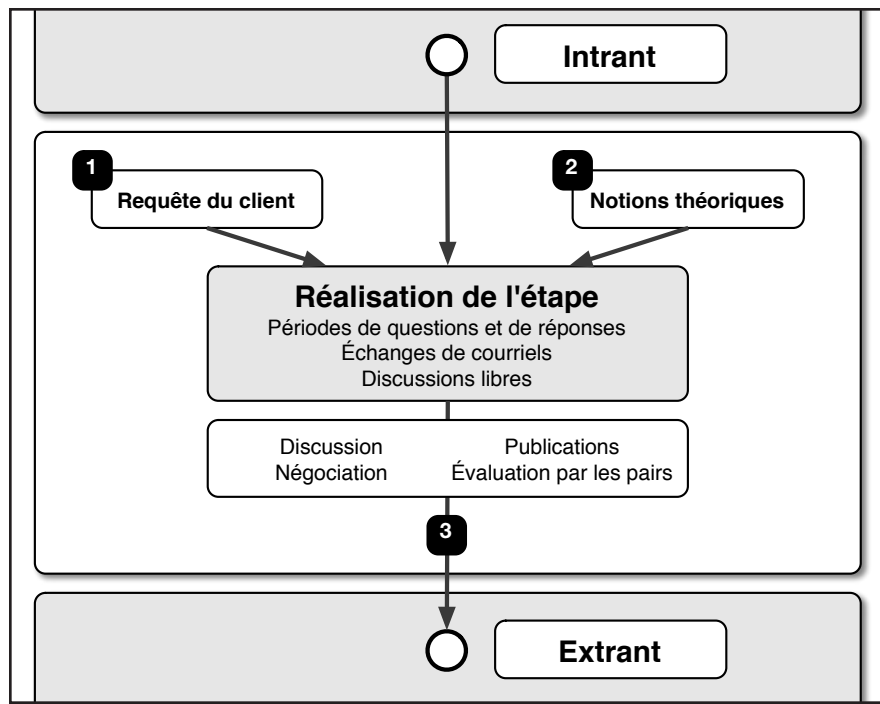


Figure 4. Déroulement d'une étape

Dans ce contexte, les interventions du client peuvent être utilisées de quatre manières différentes :

1. Pour présenter un problème de manière non technique, toujours afin d'éviter de structurer le problème ou d'offrir inconsciemment des pistes de solution et des représentations informatiques (étape 1);
2. Pour répondre aux questions de façon à établir ou non leur pertinence (étape 1);
3. Pour introduire des modifications-surprises afin de contrer les *code warriors* (étapes 2 et 3);
4. Pour faire des demandes exagérées afin que les étudiants en arrivent à estimer le temps nécessaire, à voir rapidement les implications sur leur programme et à argumenter (étapes 3 et 4).

À la fin de chaque étape, les équipes discutent de la qualité des solutions proposées et négocient, pour finalement en arriver à un consensus sur la direction générale à adopter pour la suite du projet, tout en laissant la liberté à chaque équipe de travailler dans son propre style ou approche.

À ce stade, les programmes et les documents sont publiés sous la rubrique Réalisations du site. Les équipes faibles sont alors encouragées à emprunter le code ou les éléments d'analyse d'une autre équipe. À cette fin, l'enseignant rencontre individuellement chaque équipe afin de vérifier comment avance le travail et de donner des conseils. À la fin du semestre, les projets finaux sont publiés et les meilleurs sont récompensés d'une médaille virtuelle, aspect sur lequel nous reviendrons en conclusion.

Au début de chaque cours, un résumé des questions reçues par courriel ou autrement est présenté et une discussion s'engage pour faire le point. Ces courtes séances permettent à l'enseignant d'orienter sa présentation. Lorsque les étudiants ne posent pas de questions susceptibles de mener à l'introduction du matériel théorique nécessaire, l'enseignant présente une ou deux notions théoriques suivies d'une discussion de son utilisation possible dans le contexte du projet. Par exemple, au début d'un cours portant sur l'implémentation d'une file, un étudiant a demandé si

l'on pouvait utiliser une telle structure de données pour gérer les différents sites de coulée dans la fonderie (salle 4).

Lors de certaines étapes, nous avons eu recours à des évaluations par les pairs (voir Gehringer, 2001, pour d'autres applications). Par exemple, à la suite de l'étape 1, chaque équipe devait évaluer l'analyse de trois autres équipes. Pour ce faire, elle devait remplir une grille d'évaluation élaborée par le professeur et accessible par formulaire sur le Web. Les travaux à évaluer étaient également disponibles sur le site.

À la toute fin du semestre, trois différents types d'évaluation ont été utilisés. D'abord, les équipes devaient répondre à un ensemble de questions hypothétiques du type « qu'est-ce qui arrive si... ». Ensuite, chaque équipe devait présenter son produit oralement en adoptant d'abord un discours adapté à une clientèle de futurs utilisateurs et, ensuite, un discours adapté à un groupe de professionnels (l'idée est également suggérée dans Pollock, 2001). Finalement, les étudiants se retrouvaient devant un examen final dont trois des questions étaient directement liées à la compréhension du programme que leur équipe avait réalisé.

2. Résultats généraux

Comme nous le mentionnions en introduction, le groupe comprenait 17 équipes de 3 à 5 étudiants chacune. Sur cinq équipes faibles (notes très basses lors des deux premières étapes), quatre ont pu relever leur niveau et présenter un travail qualifié de « bon » à « très bon » à la fin du semestre. Sur les huit équipes qualifiées de « moyennes » en début de semestre, une seule n'a pas su relever le défi et est demeurée au même point. Deux équipes parmi celles-ci se sont particulièrement distinguées, leur travail figurant parmi les meilleurs en fin de parcours. Quant aux équipes dites « très bonnes » et « excellentes » en début de semestre, elles ont constamment

interagi avec l'ensemble du groupe et aucune ne s'est isolée pour travailler. Évidemment, il s'agit de résultats préliminaires et une étude plus scientifique doit maintenant être entreprise avec des groupes cibles pour évaluer l'approche globale et surtout comparer les types d'utilisation du Web.

3. Discussion

Cinq caractéristiques distinguent l'approche adoptée ici, tant sur le plan pédagogique que sur celui de la mise à jour du site Web consacré au projet. Elles sont résumées dans le Tableau 1.

Un aspect important traverse l'ensemble des caractéristiques. Il s'agit du partage des solutions par les étudiants, tant dans le cadre d'une évaluation par les pairs que dans celui, plus vaste, de la visibilité des travaux. Ce partage, inspiré par une approche constructiviste de la pédagogie, produit inévitablement des interactions riches de sens pour les étudiants (Hübscher-Younger et Narayanan, 2001). Il installe un climat de collaboration

et de critique saine qui force la participation de tous et renforce le sentiment d'appartenance de l'étudiant au processus d'apprentissage (voir également les travaux d'Evans Sabin et Sabin, 1994, et de Jenkins, 1998). L'étudiant apprend à confronter ses idées et découvre d'autres manières d'aborder les problèmes, augmentant ainsi son bagage de représentations et de solutions.

L'approche présente cependant des inconvénients : la gestion de la revue par les pairs s'est avérée très lourde. Sans appui externe, la collecte des travaux, leur diffusion (ce qui implique qu'il faille les « rendre » anonymes), la distribution aux équipes, la collecte des évaluations puis leur redistribution aux équipes concernées ont constitué des tâches ardues et voraces en temps. C'est pourquoi avant de répéter une telle expérience, nous avons jugé bon de nous munir d'un système qui faciliterait la gestion entourant l'évaluation des travaux par les pairs. Le système est présentement en phase test; sa présentation forme le sujet de la prochaine section.

4. Web-éval

Il est bien connu que l'évaluation par les pairs est une excellente façon d'introduire l'étudiant à la critique, à la collaboration et au respect d'autrui. Savoir critiquer et savoir supporter la critique sont deux qualités essentielles d'un bon chercheur, mais également d'un bon ingénieur (Silversides, 2000).

Le système Web-éval a donc pour objectif de faciliter, à la fois pour l'enseignant et pour l'étudiant, la remise, l'emballage et la distribution des travaux d'une part, et l'évaluation, la récupération et la redistribution des évaluations d'autre part. Ce système régit donc tout l'aspect logistique entourant l'adoption d'une évaluation par les pairs.

Le système comprend un ensemble de trois modules simples décrits au Tableau 2.

C'est l'enseignant qui aiguille lui-même le système sur l'un des quatre modes associés à un travail : mode édition, mode diffusion, mode évaluation, mode redistribution. Nous avons choisi cette gestion manuelle plutôt qu'une gestion automatique (basée par exemple sur des dates fixées par l'enseignant) afin de rendre plus souple la gestion des retards. Ces quatre modes sont résumés au Tableau 3.

Il est à noter que le système préserve constamment l'anonymat des évaluateurs tout en permettant leur identification par l'enseignant, ce dernier pouvant évidemment faire partie des évaluateurs anonymes.

5. Conclusion

Grâce au projet présenté, les étudiants peuvent acquérir des habiletés plus complexes qui favorisent l'émergence de connaissances procédurales et conditionnelles plutôt que seulement déclaratives. Ils modélisent le comportement d'expert dans un environnement contrôlé et multidisciplinaire.

Pertinence du matériel théorique	- Notions théoriques présentées dans le contexte d'un problème à résoudre : les étudiants apprennent ainsi à faire des analogies.
Approche interactive	- Discussions, échanges d'information, évaluation des stratégies adoptées : les étudiants apprennent à décrire des problèmes, à formuler des questions, à synthétiser. Ils apprennent à communiquer.
Projet en constant développement	- Présence d'un client fictif introduisant de nouveaux paramètres, de nouvelles exigences ou contraintes : les étudiants apprennent à s'adapter, à devenir responsables, à être créatifs. - Simulation du comportement d'un professionnel dans un environnement contrôlé en introduisant ponctuellement des requêtes, parfois abusives : les étudiants apprennent à négocier, en expliquant rigoureusement et en répondant dans un langage adapté aux demandes sur les délais, les risques et les difficultés encourues (Singer et Willett, 1993).
Diffusion régulière des travaux	- Présentation orale ou écrite des travaux réalisés à chacune des grandes étapes du projet : les étudiants apprennent à partager leur expertise. Les plus faibles profitent d'une meilleure analyse à chacune des étapes, et voient au fur et à mesure les meilleures solutions retenues.
Évaluation formative	- Évaluation par les pairs : les étudiants apprennent à critiquer et à analyser objectivement le travail des autres et conséquemment, leurs propres travaux.

Tableau 1. Caractéristiques de l'approche

Dans le cadre d'une telle expérience pédagogique, un site Web ne doit pas uniquement être envisagé comme une banque de ressources. Il est nécessaire que ce qui s'y publie soit constamment pertinent (Selim, 2003) et utile pour tous les étudiants, peu importe leur expérience.

Dans un tel contexte, le Web ne se substitue pas à l'enseignant, mais contribue à permettre une pédagogie ayant pour base l'apprentissage par problème. Nous avons cerné trois grands avantages que cet espace Web privilégié devrait procurer :

- UN ESPACE DE COLLABORATION

Le Web constitue un bon moyen de partager des documents (voir Hübscher-Younger et Narayanan, 2001; Zeller, 2000), mais il ne faut pas oublier qu'il n'arrive pas à remplacer toute la richesse et la spontanéité d'une discussion en classe, forum de discussion ou non.

- UN ESPACE COHÉRENT

Sa structure, son interface et ses repères visuels doivent être construits autour de la méthode d'enseignement.

- UN ESPACE DE QUALITÉ

Toutes les ressources publiées doivent convenir au contenu et à l'apprentissage des étudiants, et être toujours adaptées à leur expérience, et non l'inverse. La pertinence plus que la quantité est ce dont les étudiants ont besoin.

Module de gestion des équipes	- construction des équipes - identification du responsable et diffusion de son courriel - identification des membres d'une équipe - possibilité d'ajouter un logo et une présentation générale de l'équipe
Module de diffusion	PAR L'ENSEIGNANT - publication des énoncés de travaux et des exercices PAR LES ÉTUDIANTS - envoi des travaux et exercices à l'enseignant
Module d'évaluation	PAR L'ENSEIGNANT - élaboration d'une ou de plusieurs grilles d'évaluation - choix de la répartition des travaux à évaluer par les autres équipes PAR LES ÉTUDIANTS - soumission des évaluations - redistribution automatique des travaux évalués aux équipes concernées PAR LE SYSTÈME - distribution des travaux à évaluer aux équipes - cueillette et redistribution des évaluations aux équipes concernées et à l'enseignant

Tableau 2. Les trois modules de Web-éval

Il est certain qu'une telle méthode d'enseignement et l'utilisation du Web exigent beaucoup de préparation et d'organisation. La gestion du site lui-même, l'évaluation, la publication des travaux et la diffusion de l'information pertinente alourdissent grandement la logistique du cours.

Cependant, avec l'aide de quelques outils informatiques, on peut alléger substantiellement cette gestion. Le système Web-éval facilitera sûrement l'aspect évaluation par les pairs.

Ce système sera complété prochainement d'un système appelé Web-vitrine facilitant la publication des travaux exceptionnels sur le Web. On sait que la publication des travaux contribue à valoriser la participa-

tion de l'étudiant dans le cadre d'un cours (Boyle, 2000). Cette nouvelle visibilité lui permet de se sentir impliqué dans sa formation et celle de ses pairs tout en donnant une image positive et dynamique de l'institution. On comprendra ici que la notion de « site Web de cours » s'en trouve enrichie, parce que diversifiée et éclatée.

Références

- Boroni, C. M., Goosey, F. W., Grinder, M. T. et Ross, R. J. (2001). Engaging students with active learning resources: Hypertextbooks for the Web. Dans H. Walker, R. McCauley, J. Gersting et I. Russell (dir.), *Proceedings of the thirty-second SIGCSE technical symposium on Computer Science Education* (p. 65-69), New-York : ACM Press.
- Boyle, T. (2000). Constructivism: A suitable pedagogy for information and computing sciences. Dans *Proceedings of the 1st Annual Conference of the Learning and Teaching Support Network Centre for Information and Computer Sciences* (p. 166-220).
- Gehring, E. F. (2001). Electronic peer review and peer grading in computer-science courses. Dans H. Walker, R. McCauley, J. Gersting et I. Russell (dir.), *Proceedings of the thirty-second SIGCSE technical symposium on Computer Science Education* (p. 139-143), New-York : ACM Press.
- Hadjerrouit, S. (1999). A constructivist approach to object-oriented design and programming. Dans B. Manaris (dir.), *Proceedings of the 4th annual*

Mode 1 Édition	- installation d'énoncés de tous types de travaux par l'enseignant (il peut s'agir d'une étape dans un plus gros projet), les étudiants n'y ayant pas accès à ce stade
Mode 2 Diffusion	- téléchargement par les étudiants des énoncés - envoi des travaux
Mode 3 Évaluation	- envoi par les équipes des grilles d'évaluation complétées
Mode 4 Redistribution	- redistribution des évaluations proprement dites Note : Les discussions en classe peuvent alors débiter.

Tableau 3. Modes associés à un travail

SIGCSE/SIGCUE ITiCSE conference on Innovation and technology in computer science education (p. 171-174), New York : ACM Press.

Hübscher-Younger, T. et Narayanan, N. H. (2001). Features of shared student-created representations. Communication présentée au *Artificial Intelligence in Education Workshop - External Representations in AIED: Multiple Forms and Multiple Roles*. Récupéré le 22 novembre 2004 de <http://www.psychology.nottingham.ac.uk/research/credit/AIED-ER/hubscher.pdf>.

Jenkins, T. (1998). A participative approach to teaching programming. Dans G. Davies et M. O'Digeartaigh, (dir.), *Proceedings of the 6th annual conference on the teaching of computing and the 3rd annual conference on Integrating technology into computer science education: Changing the delivery of computer science education*, (p. 125-129), New-York: ACM Press.

Pollock, L. (2001). Integrating an intensive experience with communication skills development into a computer science course. Dans H. Walker, R. McCauley, J. Gersting et I. Russell (dir.), *Proceedings of the thirty-second SIGCSE technical symposium on Computer Science Education* (p. 287-291), New-York: ACM Press.

Sabin, R. E. et Sabin, E. P. (1994). Collaborative learning in an introductory computer science course. Dans R. Beck et D. Goelman (dir.), *Proceedings of the twenty-fifth SIGCSE symposium on Computer science education* (p. 304-308), New York: ACM Press.

Selim, H. M. (2003). An empirical investigation of student acceptance of course websites. *Computers & Education*, 40, 343-360.

Silversides, A. (2000, juin/juillet). Le respect des pairs. *University Affairs/Affaires universitaires*, 14-17.

Turner, J. A. et Zachary, J. L. (1999). Using course-long programming projects in CS2. Dans J. Prey et B. Noonan (dir.), *Proceedings of the thirtieth SIGCSE technical symposium on Computer science education* (p. 43-47), New York: ACM Press.

Weber-Wulff, D. (2000). Combating the code warrior: A different sort of programming instruction. Dans J. Tarhio, S. Fincher et D. Joyce (dir.), *Proceedings of the 5th annual SIGCSE/SIGCUE ITiCSE conference on Innovation and technology in computer science education* (p. 85-88), New York : ACM Press.

Zeller, A. (2000). Making students read and review code. Dans J. Tarhio, S. Fincher et D. Joyce (dir.), *Proceedings of the 5th annual SIGCSE/SIGCUE ITiCSE conference on Innovation and technology in computer science education* (p. 89-92), New York: ACM Press.

Notes

¹ Le Service offre ainsi un cours de programmation en Visual Basic pour débutant, un cours de programmation intermédiaire en C et un cours de programmation avancée en C++. L'expérience décrite dans cet article et menée à l'été 2000 concerne plus particulièrement ce dernier cours, bien que les commentaires préliminaires de l'introduction touchent l'ensemble de l'offre de cours.

² « A hypertextbook is a computer-based teaching and learning resource that either augments or takes the place of a traditional textbook in a course. [...] In their most rudimentary incarnations, hypertextbooks may simply make effective use of hyperlinks that allow the reader to branch to related portions of the text. [...] More advanced applications may include sound, pictures, video, and/or animated presentations of concepts (e.g. algorithm animations). », p. 65.

³ D'ailleurs, plusieurs étudiants choisissent d'imprimer l'ensemble des documents disponibles (animations incluses) au début de la session dans le but de se constituer des « notes de cours ».

⁴ Ces étudiants expérimentés qui programment tout en deux fins de semaine.

⁵ Les suggestions des étudiants sont d'abord soumises au professeur, qui évalue leur qualité mais surtout leur pertinence.

⁶ La réduction à trois salles est le résultat d'une négociation en classe.

⁷ En contrepartie des ajouts demandés par le client, les étudiants ont négocié l'élimination de la salle 5 de la simulation.