MINISTERE DE L'ENSEIGNEMENT SUPERIEUR, DE LA RECHERCHE SCIENTIFIQUE ET DE LA TECHNOLOGIE

UNIVERSITE DU 7 NOVEMBRE A CARTHAGE
INSTITUT NATIONAL DES SCIENCES APPLIQUEES ET DE TECHNOLOGIE

PROJET DE FIN D'ETUDES

pour l'obtention du

DIPLOME NATIONAL D'INGENIEUR EN SCIENCES APPLIQUEES ET EN TECHNOLOGIE

Filière : Réseaux Informatiques et Télécommunications

Sujet : ***Conception et mise en place d'un module pédagogique pour portails communautaire PostNuke***

Réalisé par :
CHAKROUN Mourad

Entreprise :
Technologie de la Formation et de l'Apprentissage (TECFA), Faculté de Psychologie et des Sciences de l'Education (FPSE), Université de Genève, Suisse

Soutenu le :
31 Mars 2003 à Tunis

Devant le Jury composé de
Président : *LOUKIL Adlène*
Examinateur: *MADDOURI Mondher*
Responsable à l'entreprise: *SCHNEIDER K. Daniel……*
Responsable à l'INSAT: *LAROUSSI Mona*

Année universitaire 2002/2003

RESUME

L'enseignement et l'éducation sont difficilement perméables aux nouvelles technologies de l'information et aux applications en ligne. Les expériences ont montré que les systèmes informatisés n'ont pas toujours trouvé la formule juste pour tous les contextes de l'apprentissage. Beaucoup d'outils comme les CBT (Computer Based Training) ont vu le jour et s'intègrent parfaitement dans le cadre « behaviouriste » du « e-learning ». Ils sont performants pour les enseignements de base (connaissances et procédures élémentaires) mais sont peu efficaces pour l'apprentissage des matières complexes ou du savoir-faire appliqué ayant des bases socioconstructivistes. De plus, rien ou presque n'a été prévu pour ce type d'apprentissage.

Dans la perspective d'une meilleure approche pour renverser cette tendance nous en expliquons les raisons. Nous essaierons ensuite de dégager les critères pédagogiques qui feront que l'on puisse envisager l'emploi efficace des nouvelles technologies de l'information en éducation. Nous nous baserons dans notre démarche sur le socioconstructivisme, l'un des différents courants de pensées psychopédagogiques qui fait partie du champ d'investigation des recherches en cours à TECFA (Technologies de la Formation et de l'Apprentissage). Nous en conclurons que des scénarios pédagogiques intéressants combinant collaboration et recherche peuvent être supportés par des technologies en ligne naissantes tels que les portails communautaires. Nous démontrerons ainsi que des portails, comme la plate-forme PostNuke qui est en cours d'utilisation à TECFA, peuvent très bien être utilisés à des fins d'apprentissage dans l'éducation

L'un de ces scénarios est l'Argue Graph. C'est un système d'apprentissage collaboratif qui permettra à l'enseignant de concevoir des scénarios pédagogiques basés sur l'interaction entre individus faisant partie d'un groupe pour réfléchir à un thème particulier et interagir de manière constructive.

Nous expliciterons à travers ce document les différentes caractéristiques de l'Argue Graph et nous décrirons les différentes étapes de la conception et de la mise en place du système qui traduira ce scénario en un programme supporté par la plate-forme PostNuke.

Table des matières

TABLE DES MATIERES 3

CHAPITRE 1 : CADRE DU PROJET 5

1) Les technologies de l'information et de la communication et l'éducation: rappel historique et principes généraux 6

2) La notion de CSCL (Computer-Supported Collaborative Learning): de nouvelles perspectives pour l'apprentissage en ligne 7
- 2.1) Les différents courants psychopédagogiques 7
  - 2.1.1) Le socioconstructivisme 7
  - 2.1.2) L'apprentissage par projet 8
- 2.2) Le CSCL (Computer-Supported Collaborative Learning) : définitions 9

3) Qu'est ce que l' Argue Graph et pourquoi? 10
- 3.1) But du projet 10
- 3.2) Cadre du projet : Présentation de TECFA et TECFA Seed 11
- 3.3) Le catalogue TECFA Seed et le scénario Argue Graph 11
  - 3.3.1) Notion de scénario et d'activités d'apprentissage 12
  - 3.3.2) Le scénario Argue Graph 14

CHAPITRE 2 : ETUDE PRELIMINAIRES ET OBJECTIFS 15

1) Introduction 16

2) Le script Argue Graph 16
- 2.1) Les phases du script 16
- 2.2) Pourquoi un nouvel Argue Graph ? 17

3) Les portails communautaires 18
- 3.1) Les fonctions principales d'un portail 19
- 3.2) Éléments techniques d'un portail 20
- 3.3) Modules et blocs 21

4) Cahier des charges 22

CHAPITRE 3 : ETUDE DETAILLEE ET CONCEPTION 23

1) Introduction 24

2) Spécification des besoins fonctionnels du module Argue Graph 24
- 2.1) Vue globale 24
- 2.2) Vue détaillée 25
  - 2.2.1) Système d'authentification, de permissions et de profils utilisateurs 25
  - 2.2.2) Interface administrateur 25
  - 2.2.3) Interface utilisateur 29

3) Description de la conception 30

3.1) Modélisation du contexte du système 30
- 3.1.1) Les acteurs 30
- 3.1.2) Les messages 31

3.2) Modélisation des besoins fonctionnels 32
- 3.2.1) Identification des cas d'utilisation 32
- 3.2.2) Diagrammes de collaboration 34
- 3.2.3) Identification des classes 36
- 3.2.4) Diagramme de déploiement 38

CHAPITRE 4 : REALISATION 39

1) Introduction 40

2) Environnement technique 40

2.1) PHP ... 40
2.2) PostNuke ... 40
2.3) MySQL ... 41

3) Description du développement du module ArgueGraph ... 41
3.1) Caractéristiques du développement d'un module PostNuke ... 41
3.2) Architecture d'un répertoire module PostNuke ... 42
3.3) L'API PostNuke ... 43
3.3.1) Gestion des erreurs ... 44
3.3.2) Statut de l'API PostNuke : en constante progression ... 44
3.4) L'API pnHTML ... 44
3.4.1) Exemple de l'utilisation de la classe pnHTML ... 45
3.4.2) Statut de l'API pnHTML : progression lente ... 45
3.5) Développement du module « ArgueGraph » ... 46
3.5.1) Le répertoire « ArgueGraph » ... 46
3.5.2) Description du fichier « pntables.php » ... 47
3.5.3) Description du fichier « pninit.php» ... 48
3.5.4) Principe de développement ... 49
3.5.5) Navigation à travers les interfaces du module ... 50
3.5.6) Gestion de la sécurité du module ... 51
3.5.7) Génération des graphes ... 51
3.5.8) Algorithme de répartition des groupes en paires ... 53
3.5.9) Décodage des fichiers XML ... 53
3.5.10) Module « WebChat » ... 55

CHAPITRE 5 : DEPLOIEMENT ET EVALUATION ... 56

1) Introduction ... 57

2) Installation ... 57

3) Déploiement et tests ... 58
3.1) Administration du module ... 58
3.2) Exécution du scénario Argue Graph ... 59

4) Evaluation du travail ... 61
4.1) Accomplissements ... 61
4.2) Améliorations et perspectives ... 62

DISCUSSION EST CONCLUSION ... 63

ANNEXES ... 64

ANNEXE 1 : LE MODULE « FAVOURITE LINKS » ... 65

ANNEXE 2 : LE FICHIER « MYCLASS.PHP » ... 66

ANNEXE 3 : ANALYSEUR XML ... 68

BIBLIOGRAPHIE ... 70

Chapitre 1 : Cadre du projet

1) Les technologies de l'information et de la communication et l'éducation: rappel historique et principes généraux

Au cours de la dernière décennie, les ordinateurs et les réseaux ont entraîné une métamorphose radicale aussi bien d'un point de vue culturel et social qu'administratif, politique et économique. Or, l'enseignement est difficilement perméable aux innovations technologiques. Celles-ci pourraient cependant constituer le tremplin idéal pour accéder à un enseignement tourné vers l'avenir et à une formation permanente compte tenu des énormes possibilités qu'elles offrent.

Cela fait plus de 20 ans que les ordinateurs sont utilisés dans l'éducation. L'enseignement assisté par ordinateur a commencé avec les CBT (Computer Based Training) et les CAI (Computer Aided Instruction) qui étaient les premiers systèmes déployés dans l'enseignement utilisant des ordinateurs.

Avec ces premiers systèmes, l'instruction n'était pas individualisée aux besoins de l'étudiant : la façon dont on faisait naviguer l'étudiant à travers le matériel éducatif était purement algorithmique comme par exemple « si vous avez bien répondu à la question 21, allez à la question 54; sinon allez à la 32 ». Les logiciels dits « Drill and practice », qui se basent sur des banques de données d'exercices stockés et dont l'interaction avec l'utilisateur est de type réactif, ainsi que les didacticiels, qui reposent sur des dialogues interactifs consistant à mémoriser et à réaliser des séquences de procédures associées à certains concepts, font partie de ce type de systèmes. Ils sont efficaces pour l'apprentissage du calcul mental ou de tables de grammaires mais le sont moins pour d'autres domaines parce qu'ils ne favorisent pas les initiatives de l'apprenant et ne fournissent pas le même genre d'attention individualisée qu'un étudiant recevrait d'un précepteur humain (Bloom 84). Pour qu'un système d'éducation informatisé fournisse une telle attention, il doit soit raisonner à la fois sur le domaine de connaissances et sur celui qui doit apprendre, soit donner plus de contrôle à l'apprenant et faire intervenir l'enseignant à des fins d'encadrement et de modération.

De nouveaux systèmes capables d'exercer une forme de tutorat ont donc vu le jour et ont permis d'augmenter la motivation et les résultats des étudiants. Nous pouvons citer comme exemple les logiciels se basant sur le « coaching » qui fait glisser le rôle de l'instructeur vers celui de guide ou entraîneur du savoir, ou bien les simulateurs qui permettent à l'apprenant d'explorer une situation dans toute sa complexité, de prendre des décisions et d'en observer les conséquences.

Ces systèmes, et encore de nombreux autres que nous n'avons pas cités, peuvent se trouver sous diverse formes, allant des micro mondes à certaines nouvelles technologies de l'information et de la communication (NTIC). Les technologies Internet et particulièrement le « World Wide Web » exercent même une influence croissante sur l'enseignement, notamment parce qu'elles véhiculent souvent des stratégies pédagogiques nouvelles et parce qu'elles permettent d'organiser des formations plus flexibles dans le temps et l'espace (Schneider 2001).

Dans notre projet, nous nous concentrons plus particulièrement sur les technologies éducatives qui s'insèrent dans le cadre d'un apprentissage avancé combinant initiative de l'apprenant et contrôle de l'enseignant. Ces technologies doivent pouvoir fournir des outils permettant un apprentissage en ligne.

Nous partirons du postulat que tout apprentissage avancé se basant sur Internet doit :

- Permettre une formation efficace et vitale pour pouvoir survivre culturellement et économiquement dans le "global village" de demain.
- Donner une formidable impulsion à l'apprentissage de la pensée, à l'autoformation, mais aussi à l'apprentissage actif, constructif, différencié et créatif.
- Prendre en compte les matières enseignées, le mode d'apprentissage et les enseignants eux-mêmes.
- Permettre non seulement le télé-enseignement, mais aussi le développement social et culturel.
- Motiver et favoriser la communication entre les enseignants, les élèves et les parents. L'introduction de réseaux informatiques dans l'enseignement peut constituer un énorme facteur de stimulation pour tous les jeunes (Frimout & Wynants 2000).

Dans la perspective d'une meilleure approche de ce que doivent permettre ces technologies, il est indispensable de passer en revue les différentes théories de l'apprentissage et les stratégies pédagogiques qui y sont associées en vue de développer une architecture efficace de formations, officielles et non officielles, qui permettront tant aux enseignants qu'aux élèves d'épanouir pleinement leurs talents créatifs, sociaux et intellectuels.

2) La notion de CSCL (Computer-Supported Collaborative Learning): de nouvelles perspectives pour l'apprentissage en ligne

2.1) Les différents courants psychopédagogiques

Il existe, entre tant d'autres, quatre grandes écoles de pensée psychopédagogique qui se prononcent sur les mécanismes d'apprentissage et de stratégies pédagogiques. Le béhaviorisme stipule qu'apprendre signifie acquérir un comportement par le biais de mécanismes de renforcement. Les CBT et plus généralement l' « e-learning » sont des technologies béhavioristes. Ils sont performants pour les enseignements de base (connaissances et procédures élémentaires) mais sont peu efficaces pour l'apprentissage des matières complexes ou du savoir-faire appliqué. Les approches socioculturelles tendent à penser que les interactions sociales qu'un individu expérimente façonnent une partie importante de son apprentissage. Certaines approches de la cognition, quant à elles, prônent l'insertion de l'apprenant dans une communauté de pratique. Enfin, le constructivisme postule que l'on construit ses connaissances en faisant ou encore en construisant des objets.

Du constructivisme et de la socio-cognition découle l'approche socioconstructiviste.

2.1.1) Le socioconstructivisme

La définition du socioconstructivisme est assez large. Il s'agit avant tout d'une conception de l'apprentissage qui met en avant, d'une part, l'importance d'une construction des connaissances basées sur ce que l'apprenant sait déjà et, d'autre part, l'interaction de l'individu avec son environnement social. Ces théories puisent leur source dans le constructionnisme, le socio-culturalisme et la socio-cognition.

Bien que présentant un certain nombre de différences, les pédagogies socioconstructivistes ont quelques dénominateurs communs.

On peut isoler, avec Wilson et Lowry (2001), trois principes clé pour une utilisation efficace d'Internet pour l'apprentissage :

- Fournir un accès à des sources d'information riches ;
- Encourager une interaction fructueuse avec des contenus ;
- Amener les sujets à relever ensemble des défis, se soutenir, se répondre.

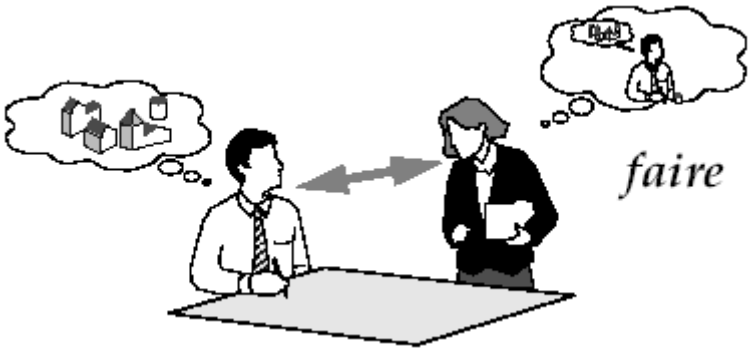


Figure 1 : Socio-constructivisme : recherche et collaboration

Mais il ne suffit pas de placer les sujets dans des situations de collaboration pour qu'un apprentissage efficace ait lieu. L'enseignant doit mettre en place des scénarios structurés pour obtenir ce résultat. Un scénario est une séquence de phases et de rôles spécifiques à jouer. La technologie aura donc, dans ce contexte, davantage pour but d'aider les apprenants à mener à bien des tâches complexes que de délivrer du matériau de cours (Dillenbourg 2002).

Il existe un ensemble important de modèles pédagogiques qui s'insèrent dans le cadre du socioconstructivisme tels que l'apprentissage par projet et l'apprentissage collaboratif assisté par ordinateur. Un facteur réunit ces deux pédagogies : la communauté d'apprenants. La communauté est définie comme un réseau constitué d'individus ayant des pratiques et des buts communs. Cette communauté va entourer et aider l'apprenant à construire ses connaissances (Bielaczyc & Collins, 1999). Le facteur communauté est plus prononcé dans la pédagogie de l'apprentissage par projet dont nous décrivons les principes généraux ci-dessous.

2.1.2) L'apprentissage par projet

L'apprentissage par projet est un modèle qui se démarque de l'enseignement traditionnel en se recentrant davantage sur l'apprenant qui se voit assigner des projets à réaliser. Les projets permettent ainsi de rendre les apprenants plus autonomes dans le processus d'apprentissage et les poussent davantage à la recherche, à la communication et à la collaboration. La notion de projet est par ailleurs centrale dans le cadre des pédagogies socioconstructivistes.

Synteta (2001) définit l'apprentissage par projets comme suit :

- Des expériences d'apprentissage engageantes qui incluent les apprenants dans des projets complexes ancrés dans la réalité et au travers desquels ils développent et appliquent des compétences et du savoir ;
- Un apprentissage qui incite les apprenants à avoir recours à de nombreuses sources d'information et disciplines afin de résoudre des problèmes ;

- Un apprentissage dont les buts sont identifiés et formulés, mais par rapport auxquels les enjeux et les résultats des processus d'acquisition des connaissances ne sont ni prédéterminés, ni entièrement prévisibles ;
- Des expériences au travers desquelles les apprenants apprennent à gérer et à manipuler des ressources telles que le temps et divers matériaux (logiciels, systèmes technologiques…).

L'apprentissage par projet nécessite donc un investissement important de l'apprenant qui doit déterminer la problématique qui lui est proposée à travers le projet, suivre un plan de travail décomposé en plusieurs étapes pour aboutir aux résultats espérés, puiser dans toutes les ressources possibles pour ses recherches, communiquer avec la communauté pour essayer de « profiter » de l'expérience des autres apprenants, collaborer afin de trouver des solutions « ensemble » à travers l'échange d'idées et interagir avec les professeurs pour tout ce qui est suivi temporisé du projet.

2.2) Le CSCL (Computer-Supported Collaborative Learning) : définitions

La libre collaboration ne permet pas systématiquement l'apprentissage. Une des meilleures façons d'améliorer l'efficacité de l'apprentissage collaboratif est d'engager les étudiants dans des interactions structurées faisant partie de scénarios bien définis. L'apprentissage collaboratif assisté par ordinateur ou CSCL est basé sur cette notion de scénario. Le scénario est une séquence de phases bien définies ayant plusieurs attributs. Ces phases servent à réguler la collaboration et à l'encadrer afin qu'elle puisse aboutir à un apprentissage que l'on souhaite efficace. L'idée est donc de structurer la collaboration afin d'aboutir à un résultat optimal.

Plusieurs études ont montré que la réussite de l'apprentissage collaboratif dépend de plusieurs facteurs, dont la composition du groupe des apprenants, la tâche à effectuer et le média de communication. Cependant, ces facteurs multiples interagissent les uns avec les autres d'une manière si complexe qu'il est difficile d'en garantir un effet d'apprentissage. C'est pour cela que le contrôle de l'efficacité de la collaboration s'est quelque peu déplacé des préconditions de la collaboration au processus collaboratif lui-même (Dillenbourg, Backer, Blaye & O'Malley, 1995). En d'autres termes, il faut directement influencer les interactions des apprenants en augmentant la fréquence des conflits, en poussant à l'élaboration d'explications pertinentes et à la construction de l'argumentation, en encourageant l'entente mutuelle…

La régulation des interactions dans le processus d'apprentissage collaboratif est donc très importante, néanmoins, sa structuration reste aussi primordiale puisqu'elle permet l'émergence d'interactions productives en agissant sur la composition des groupes et la manière de procéder. La régulation de l'apprentissage collaboratif reste complexe. Elle nécessite l'intervention du facteur humain car agit surtout sur le dynamisme social du groupe. La traduction de l'intervention humaine en processus informatisé peut être très coûteuse en ressource pour un système d'apprentissage. Cependant on pourra confier à un tel système la structuration de l'apprentissage collaboratif à travers des applications scripts ou des interfaces de communication structurées.

Un scénario est une séquence d'instructions qui détermine comment les membres d'un groupe doivent interagir, comment ils doivent collaborer et comment ils doivent résoudre le problème. Quand un professeur engage les étudiants dans un apprentissage collaboratif, il leur donne des instructions du type « faites l'activité par groupes de 3 ». Ce type d'instructions est rarement accompagné de consignes explicitant la manière avec laquelle les apprenants doivent

travailler ensemble. Un scénario permet à l'enseignant d'établir un contrat plus explicite didactiquement avec ses étudiants et de mieux structurer leur mode de collaboration. Ce contrat peut être véhiculé à travers une séquence d'instructions, ou bien être englobé dans un environnement CSCL.

Syntaxe d'un scénario CSCL :

Un scénario CSCL est une séquence de phases à travers lesquelles les étudiants doivent passer. Dans un système informatisé, le scénario est sous-tendu par un script qui détermine le comportement du programme en fonction des différentes options. Un script peut être linéaire. La phase (n+1) est toujours précédée par la phase (n). Il peut ne pas être linéaire dans la mesure où les apprenants peuvent sauter quelques phases. Il est cependant plus simple de recourir à des scripts linéaires.

Chaque phase du scénario a cinq attributs :

- La tâche à effectuer : répondre à des questions, résoudre un problème ;
- La composition du groupe : comment est constitué le groupe, qui est avec qui ;
- La distribution des tâches : qui fait quoi ;
- Le mode d'interaction : comment communiquent A et B (face à face, en ligne, etc.) ;
- Le temps : durée de la phase, son rang parmi les autres phases.

Il existe plusieurs scénarios d'apprentissage collaboratif qui peuvent être intégrés dans l'environnement d'un programme CSCL.

Dans le cadre de mon stage, qui s'est déroulé à TECFA Seed (Université de Genève), j'ai été amené à concevoir et à mettre en place l'un de ces programmes: l'Argue Graph.

Nous reviendrons plus en détail sur les principes du script de l'Argue Graph dans le chapitre suivant.

3) Qu'est ce que l' Argue Graph et pourquoi?

L'Argue Graph est un système d'apprentissage collaboratif qui permettra à l'enseignant de concevoir des scénarios pédagogiques CSCL basés sur l'interaction entre individus faisant partie d'un groupe pour réfléchir à un thème particulier et interagir de manière constructive sur un thème particulier que l'enseignant peut définir à son gré.

3.1) But du projet

L' Argue Graph est un script qui a déjà été développé à deux reprises par TECFA, Université de Genève. Les deux versions de l'Argue Graph étaient autonomes dans la mesure où elles tournaient de manière indépendante sur réseau. Mais entre temps (en 2001), TECFA a rejoint le projet européen SEED qui a pour but d'offrir aux enseignants un support pour concevoir des activités d'apprentissage innovantes et de tirer profit de la technologie offerte par les portails communautaires- que ce soit dans le contexte de leurs classes ou au sein de communautés virtuelles – pour collaborer, créer et gérer des contenus. L'idée était donc de concevoir et de mettre en place une nouvelle version de l'Argue Graph qui puisse s'intégrer dans un portail communautaire utilisé par TECFA, en l'occurrence « PostNuke ».

3.2) Cadre du projet : Présentation de TECFA et TECFA Seed

TECFA (Technologies de la Formation et de l'Apprentissage) est une unité active dans le domaine des technologies éducatives. Elle fait partie de la FPSE (Faculté de Psychologie et de l'Education) de l'Université de Genève, Suisse.

Figure 2 :Logo TECFA

Les recherches à TECFA couvrent un large champ d'investigation, incluant : les implications cognitives liées aux nouvelles technologies de l'information et de la communication, l'apprentissage collaboratif, les effets cognitifs des logiciels éducatifs, la collaboration assistée par ordinateur, les systèmes d'information en éducation, le multimédia, l'apprentissage et l'enseignement à distance.

Depuis l'automne 1994 TECFA offre un diplôme post gradué (niveau DESS) en Sciences et Technologies de l'Apprentissage et de la Formation (le "STAF"). TECFA a apporté son soutien à de nombreuses formations, et a participé à plusieurs projets européens. Depuis sa création, ce diplôme combine activités en présentiel (6 semaines par an) et séquences de formation à distance. TECFA a été un pionnier dans la conception d'un campus virtuel favorisant une approche constructiviste.

TECFA collabore à plusieurs projets de recherches dans le domaine de l'éducation. TECFA Seed en fait partie. En effet, Seed est un projet européen qui a pour but l'intégration du changement culturel dans le système scolaire à travers la génération de communautés engagées dans l'intégration des innovations éducatives et technologiques. Ce projet a vu le jour le 1er Avril 2001 et continuera jusqu'à Avril 2004. Il est composé de 7 partenaires dont 4 universités : l'Université de Genève, l'Université d'Athènes, l'Université d'Hull et l'Université de Duisburg.

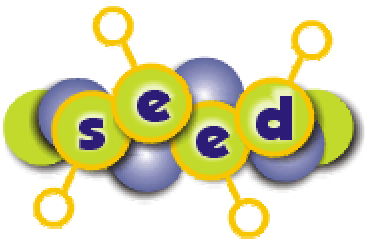


Figure 3 : Logo Seed

Chaque partenaire du projet s'est vu attribuer des tâches particulières. TECFA a pris l'initiative d'élaborer un catalogue regroupant des activités éducatives inspirées de scénarios pédagogiques riches pouvant être adoptés par les enseignants.

3.3) Le catalogue TECFA Seed et le scénario Argue Graph

Les C3MS (Community, Content and Collaboration Management Systems) sont des outils modulaires permettant de configurer des sites web communautaires interactifs (portails). Inspirés par la technologie des « weblogs », par les systèmes simples de gestion de contenus ainsi que par les collecticiels permettant de partager des fichiers ou des calendriers, ces outils

présentent des caractéristiques particulièrement intéressantes pour servir de support à l'élaboration de scénarios pédagogiques riches (Schneider, Synteta & Freté 2002). Un scénario CSCL, tel que celui de l'Argue Graph peut être traduit sous la forme d'un ensemble séquence de briques C3MS.

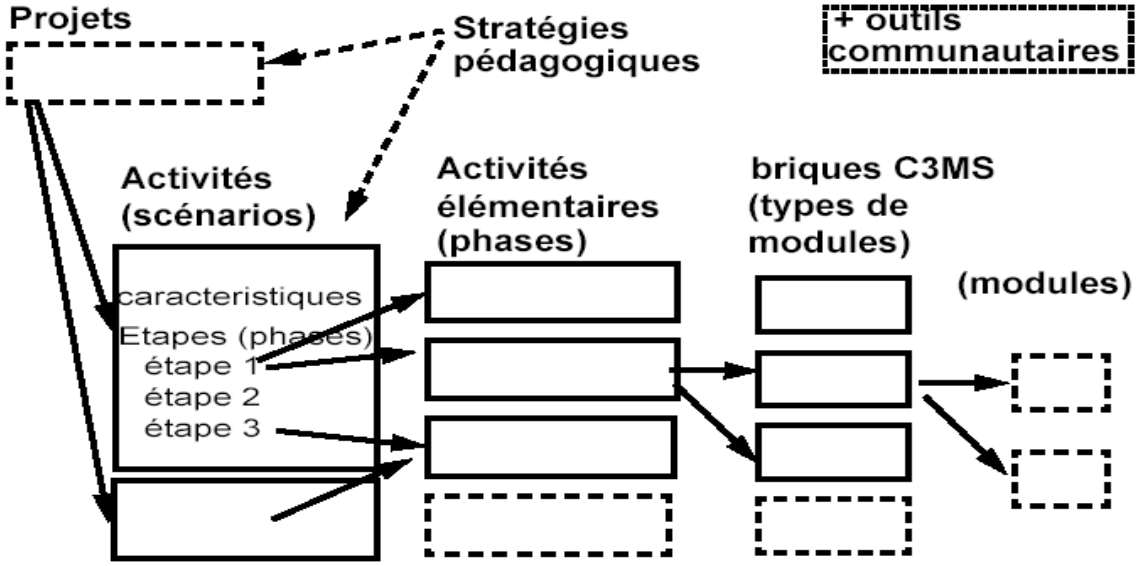


Figure 4 : Structure d'un scénario pédagogique dans le catalogue TECFA Seed

Le catalogue TECFA Seed regroupe un ensemble d'activités et scénarios socioconstructivistes et offre aux enseignants une infrastructure leur permettant de s'approprier cette technologie afin de concevoir des activités de groupe et leurs propres scénarios pédagogiques centrés sur l'apprenant. Les activités sont structurées sous forme d'unités que nous appellerons « briques » C3MS et qui peuvent être utilisées et assemblées dans les portails communautaires. Les professeurs peuvent donc choisir un certain nombre de briques en fonction de leurs objectifs d'apprentissage et les assembler dans n'importe quel ordre en fonction de leurs objectifs d'apprentissage.

3.3.1) Notion de scénario et d'activités d'apprentissage

Un scénario pédagogique est composé d'un certain nombre d'activités. Chaque activité doit générer de la curiosité et de la motivation chez l'apprenant. Elle doit être basée sur l'exploration, la recherche d'information, l'expérimentation et la formulation d'hypothèses. Les apprenants doivent être actifs et créatifs et être amenés à discuter et coopérer avec les autres pour trouver des solutions. Chaque scénario sera décrit dans le catalogue à travers six attributs :

- **But :** But du scénario (script);
- **Public:** La tranche d'âge à laquelle est destinée le scénario ;
- **Description:** Description du scénario ;
- **Durée:** Estimation de la durée totale du déroulement du scénario ;
- **Notes:** Eclaircissements ou conseils d'utilisation ;
- **Étapes:** Description des différentes activités faisant partie du scénario.

À ces étapes correspondent un ou plusieurs actes simples réalisables à l'aide des différentes briques C3MS.

Ce scénario est tiré du catalogue officiel TECFA Seed. Il reprend le scénario des listes de références.

(1) Liste de références

But:
Recherche sur Internet, catalogage, conceptualisation, synthèse...
Public:
11 ans et plus
Description:
Les élèves doivent travailler sur un thème qu'ils ne maîtrisent pas, dans le cadre d'un projet. Ils ont pour mission de créer ensemble une liste de sites Internet utiles à sa réalisation, de les cataloguer et d'en décrire le contenu de manière détaillée
Durée:
De quelques jours à quelques semaines
Notes:
Selon le niveau des élèves, le professeur peut choisir de définir les catégories avant la recherche.
Etapes:
1- Le professeur introduit le thème en donnant des pistes et en demandant aux élèves de réfléchir aux différents aspects du sujet.
"IntroWork", "BrainStorm"
2- Les élèves cherchent sur Internet en utilisant plusieurs moteurs et annuaires de recherche et les conservent les liens pertinents dans les favoris.
"SearchWeb", "KeepReference"
3- Les élèves tentent de dégager un certain nombre de catégories et sous-catégories pour ce thème "CreateCategories"
4- On met en commun les résultats en les hiérarchisant

Nous remarquons que dans la partie « Etapes », le scénario est subdivisé en activités simples. Pour ce scénario, l'une des activités simples est « SearchWeb » qui se base sur une brique C3MS de type moteur de recherche. Cette activité, ainsi que toutes les autres sont référencées par la suite dans le chapitre « catalogue des actes simples ».

Pour l'acte simple « SearchWeb », le chapitre « catalogue des actes simples » fournit cette description.

(26) SearchWeb

Rechercher des informations sur Internet
Briques C3MS:
• "Search Web"
Il s'agit d'utiliser un moteur de recherche. Il est possible d'inclure un formulaire directement dans le portail (et de restreindre les domaines recherchés)

L'acte simple « SearchWeb » est à son tour associé à une brique C3MS « Search Web ». Cette brique est référencée dans le tableau des briques C3MS comme suit :

(5) Search Web

Noms de Software:
Google search, PN DMOZ Search, Websearch
Description fonctionnelle:
Recherche sur Internet

3.3.2) Le scénario Argue Graph

Le scénario Argue Graph a été inclus dans le catalogue TECFA Seed. Les spécifications du scénario telles qu'elles apparaîtront dans le catalogue seront :

(16) Argue Graph

But:
Argumentation, conceptualisation, explicitation des idées, explicitation, analyse, synthèse…
Public:
À partir de 15 ans
Description:
Il s'agit d'amener les élèves à se confronter aux points de vue des autres en organisant des discussions par paires ayant des opinions divergentes à propos d'un thème particulier.
Durée:
Trois à quatre séances.
Notes:
Les questions doivent être « ouvertes »
Étapes:
1- Le professeur utilise un quiz existant ou en crée un nouveau.
2- Les élèves répondent individuellement au questionnaire en explicitant leurs choix.
3- Les élèves visualisent le graphe généré et en discutent.
"Interact"
4- En fonction des réponses et de leur illustration sur le graphe, le système ou le professeur forme des paires d'élèves opposées.
5- Les élèves interagissent par paires pour se mettre d'accord sur une réponse.
"Interact"
6- Le système génère un nouveau graphe et les élèves comparent l'évolution de leurs opinions.
"Interact"
7- Le professeur organise une séance de débriefing afin de permettre aux élèves de structurer leurs nouvelles connaissances.
"BrainStorm", "Interact", "SendFeedBack"
8- Les élèves rédigent une synthèse de ce qu'ils auront appris durant cette activité. "EditGlossary", "CreatePage", "CreateCategories", "CoEdit"

L'Argue Graph, tel que décrit dans le catalogue, est le scénario CSCL le plus complexe.

Avant de décrire en détail la conception du nouveau système, il nous faut explorer de plus près les caractéristiques et les spécifications du script Argue Graph.

Chapitre 2 : Etude préliminaires et objectifs

1) Introduction

Dans cette partie, nous allons nous intéresser de plus prés aux spécificités du script Argue Graph. Nous allons d'abord détailler les différentes phases qui le composent, nous citerons dans un second temps les spécificités que nous voudrons rajouter par rapport aux anciennes versions et nous conclurons en établissant un cahier des charges du projet Argue Graph.

2) Le script Argue Graph

L'Argue Graph est un système d'apprentissage collaboratif qui permettra à l'enseignant de concevoir des scénarios pédagogiques CSCL basés sur l'interaction entre individus. Nous en décrivons les caractéristiques dans ce qui suit.

2.1) Les phases du script

Le script Argue Graph est basé sur un questionnaire à choix multiples se rapportant à un thème particulier. Le professeur détermine, selon le thème, ce que peuvent en être les points de vue extrêmes qui seront au nombre de quatre. Ces quatre positions opposées font références aux quatre extrémités d'un repère orthonormé dans un espace à deux dimensions.

Pour chaque question, le professeur prévoira au moins deux réponses. Il est important de signaler qu'aucune réponse n'est bonne ou mauvaise dans le concept Argue Graph . En effet, chaque réponse reflète une certaine manière de voir les choses par rapport au thème choisi et se rapproche plus au moins de l'un des quatre points de vue extrêmes déjà déterminés. C'est pour cela que pour chaque réponse, le professeur déterminera un couple de coordonnées qui placera l'opinion véhiculée par la réponse sur le système axial du thème. Les coordonnées doivent varier entre –1 et 1.

Du côté des utilisateurs, qui sont en l'occurrence les apprenants, il s'agira de répondre au questionnaire. À la fin, la position de l'apprenant par rapport au sujet traité sera calculée en additionnant les coordonnées de chacune de ses réponses. Les coordonnées du couple total situeront l'apprenant sur l'espace bidimensionnel du thème.

Le script se déroule en 5 phases :

1. Chaque étudiant répond au questionnaire en ligne. Pour chaque choix, l'étudiant tape un argument dans une zone de texte libre.

2. Dès que tous les étudiants ont répondu à toutes les questions, le système produit un graphe sur lequel ils sont positionnés. Les positions sont déduites de la somme des coordonnées de leurs réponses. Les étudiants analysent ensuite le graphe et en discutent. Le système ou le professeur, réparti ensuite les étudiants par paires en mettant ensemble, autant que possible ceux qui présentent des opinions les plus divergents.

3. Les paires ainsi composées répondent ensuite ensemble au même questionnaire qu'à l'étape 1 et donnent de nouveau des arguments. Le système affiche les réponses et arguments émis lors de l'étape 1. Ils en discutent et essaient de collaborer pour trouver un terrain d'entente afin de renvoyer une réponse commune au système.

4. Dès que toutes les paires d'étudiants ont répondu à toutes les questions en mode duo, le système produit un nouveau graphe. Les positions des paires sont déduites de la somme des coordonnées de leurs réponses communes. Sur le graphe apparaîtront les positions des étudiants lors de la phase solo et les positions lors de la phase duo. Les étudiants constateront l'évolution de leurs opinions. Un récapitulatif du questionnaire et des statistiques concernant la séance Argue Graph écoulée seront affichés et utilisés par le professeur pour un débriefing en face-à-face.

5. Chaque étudiant écrit une synthèse de tout ce qui s'est dit et de ce qu'il a appris à travers son expérience Argue Graph. La synthèse doit être structurée dans un cadre théorique introduit par le professeur pendant le débriefing.

Ce script à été utilisé avec succès pour enseigner la relation entre les théories de l'apprentissage et le design des logiciels éducatifs (Jermann & Dillenburg, 1999). Ce script peut être généralisé dans tous les domaines pour lesquels des théories multiples coexistent. Trois points sont à retenir de l'Argue Graph:

- Le script intègre des activités en ligne ainsi que des activités dites face-à-face ;
- Le script n'est pas à 100% collaboratif: il inclue une phase d'interactions entre les étudiants (3) mais aussi des phases individuelles (1 et 5) et une phase collective (4) ;
- Le but de ce script est de créer des conflits entre les étudiants et de les engager dans des échanges d'idées afin d'enrichir leur vision d'un domaine donné.

2.2) Pourquoi un nouvel Argue Graph ?

Comme cela a été cité précédemment, deux versions du script ont été testées à TECFA. L'une a été programmée en PHP et la seconde en utilisant des applets Java. Ces applications tournaient de manières autonomes sur réseau et malgré quelques lourdeurs constatées pour la version Java, elles donnaient des résultats satisfaisants. Cependant, elles étaient peu malléables et peu documentées ce qui compliquait la compréhension de leurs codes source. De plus, ces deux versions étant très peu portables en dehors de TECFA, l'Argue Graph n'aurait donc pas pu être intégré dans le catalogue Seed des activités pédagogiques.

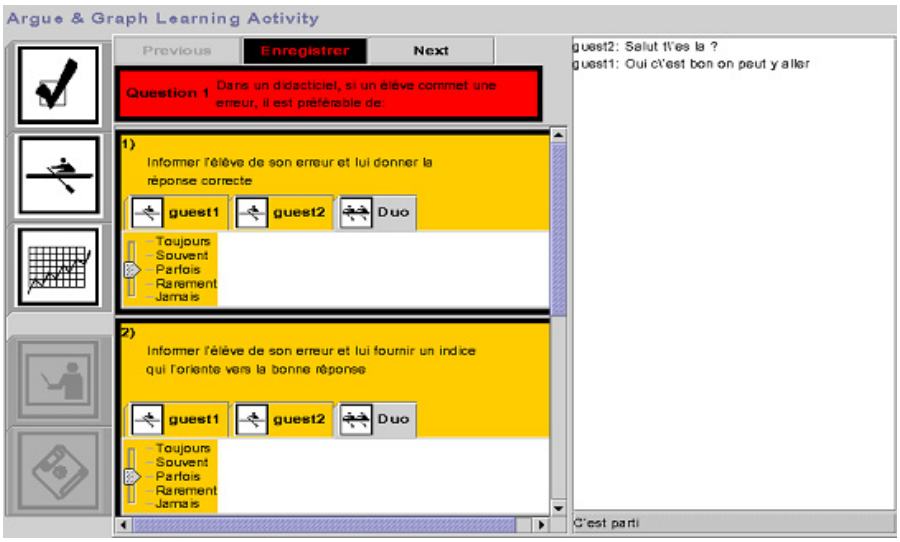


Figure 5 : Ancien Argue Graph réalisé avec des applets Java

En effet, tout scénario décrit dans ce catalogue doit pouvoir être intégré au sein d'un portail communautaire sous forme de briques C3MS. Il en découle que l'intégration d'un tel scénario nécessite la programmation d'un module complexe compatible avec la technologie du portail qui va l'intégrer. Il était donc nécessaire de reprogrammer l'Argue Graph en fonction du portail qui allait l'héberger. Sous cette nouvelle forme, l'Argue Graph aurait une plate-forme unique qui permettra de l'utiliser facilement Il suffira aux utilisateurs d'installer le portail adéquat et d'y intégrer le module pour en profiter. Pour TECFA, le portail choisi est le portail PostNuke. Nous y reviendrons plus tard dans la partie réalisation et allons au préalable nous intéresser de plus près à cette technologie que sont les portails communautaires.

3) Les portails communautaires

Les portails rassemblent une grande variété de ressources utiles dans une simple page Web. Un portail intègre des dispositifs qui lui permettent non seulement de collecter de l'information, mais lui ajoute une autre dimension l'habilitant à construire des communautés. Les portails offrent de multiples services et des interactions avec les personnes et l'information. Ces services sont facilement accessibles à travers une interface centralisatrice dont une partie reste disponible à tout moment. Ces portails sont des technologies naissantes et évoluent très rapidement ce qui fait qu'elles n'ont pas beaucoup été utilisées dans l'éducation jusqu'ici. TECFA a décelé un important potentiel dans ces portails; c'est ainsi qu'ils ont décidé d'employer cette technologie et de profiter au maximum de ce qu'elle offre.

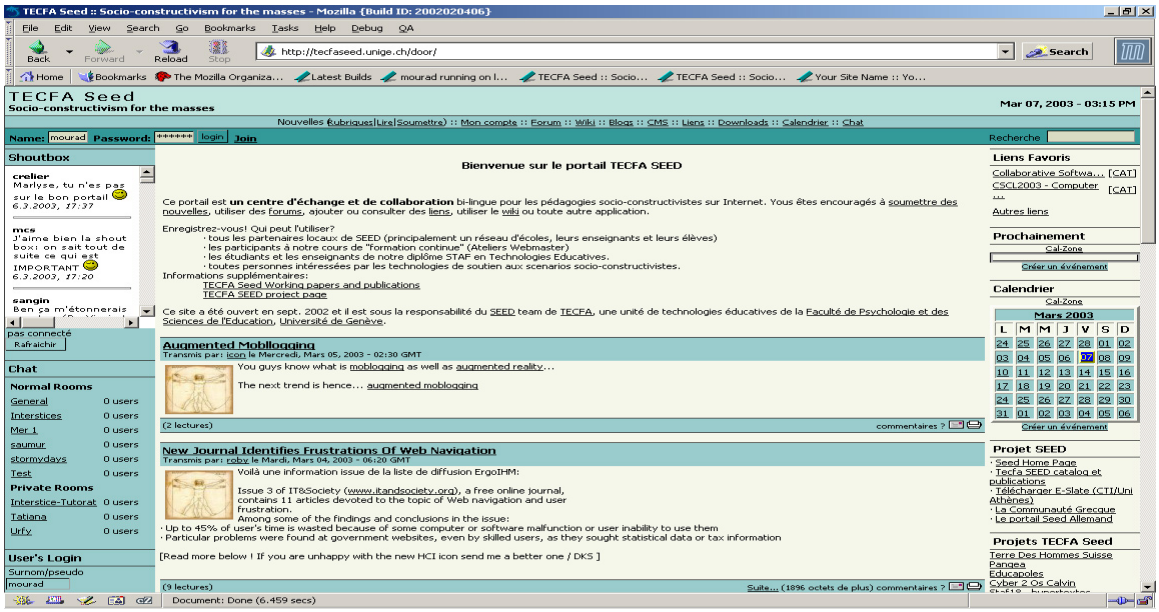


Figure 6 : Portail TECFA Seed

Typiquement, ces systèmes offrent des fonctionnalités comme le weblogging, les news (employé par des nouveaux portails majeurs comme Slashdot), les forums classiques, un calendrier d'événement, des systèmes de partage de liens, etc. En outre, ils mettent en application divers dispositifs d'organisations automatiques basés sur la modération, les calculs et la réputation. Les administrateurs et les utilisateurs peuvent configurer le système par l'intermédiaire d'interfaces en ligne. Les professeurs sont intuitivement familiers avec beaucoup de ces dispositifs même s'ils ne les employaient pas sur un seul système et/ou sur Internet et ils sont donc disposés à les adapter plus ou moins aisément.

3.1) Les fonctions principales d'un portail

Les portails communautaires ont deux grandes fonctions qui sont la gestion et l'échange. La notion d'échange est d'ailleurs le principe fondamental de la survie de la communauté. Nous dressons, dans ce qui suit, un tableau récapitulatif des fonctionnalités d'un portail communautaire.

Fonction	**Outils**
Gestion de contenus	▪ Content Management Systems (CMS) ▪ Système de News (rubriques) ▪ Hypertextes collaboratifs (Wikis) ▪ Albums, Glossaires et autres outils spécialisés
Échange de connaissances	▪ Système de News (avec discussion) et leur syndication ▪ Systèmes de partage de fichiers avec annotations ▪ Systèmes de tickets
Échange d'arguments	▪ Forums et/ou News ▪ Chats, shoutboxes
Gestion et support de projets	▪ Outils de gestion de projets, ▪ Outils "Virtual Teams" ▪ Calendriers
Gestion de connaissances	▪ Gestionnaires de FAQ, de Liens etc. ▪ Outils de recherche, box "top 10" etc.
Gestion de la communauté	▪ Profil et identification des membres ▪ Systèmes de réputation ▪ Trace d'activités des membres (contributions) ▪ Calendrier d'événements

Toutes les fonctionnalités citées dans ce tableau font que lorsqu'un internaute passe par un portail communautaire, et même s'il n'en est pas membre, son passage est comptabilisé et retenu pour les statistiques. Il se passe donc toujours quelque chose dans un portail communautaire (contrairement aux pages Web statiques), ce qui explique la réussite de cette technologie. Nous pouvons en outre noter que les outils décrits dans ce tableau sont en fait des briques C3MS comme les forums ou les « Wikis ».

Figure 7 : L'éditeur hypertexte collaboratif Wiki

3.2) Éléments techniques d'un portail

Le tableau suivant dresse les spécificités techniques des portails communautaires.

Administration du site & des contenus avec un outil d'administration	▪ Gestion de la forme et autres personnalisations ▪ Gestion des blocs et des applications ▪ Modération (admission de contenus) ▪ Logs ▪ …
Autorisation & Sécurité Catégories d'utilisateurs de base: administrateur, utilisateur, invité. Droits différenciés pour chaque application: administration, modification, ajout, consultation, ...	▪ Personnalisation ▪ Nom de compte (unique pour toutes les applications) ▪ Time out pour les sessions ▪ Filtres de contenus ▪ Administration et gestion de contenus distribués ▪ Support SSL (Secure Socket layout) ▪ ..... ▪
Mécanismes d'extensions « Application Programmer's Interfaces » (API) pour développeurs Facilité d'installation	▪ Add-ons ▪ Plugins ▪ Extensions non-conformes ▪ …
Fonctionnalités de base L'engin des nouvelles est au centre (principe d'organisation par section) et chronologique	▪ Contenus "interactifs" selon le modèle du journal ▪ Forums (parfois) ▪ Importation de titres provenant d'autres sites ▪ Boites pour d'autres informations ▪ Engins de recherche imbriqués
Extensions Beaucoup de « petites applications » Internet sont en train d'être intégrés dans les portails La plupart des portails ont un site dédié aux extensions. On distingue souvent entre modules officiels, modules bêta et modules non supportés.	▪ Bulletin Boards ▪ Forums ▪ Mini-questionnaires ▪ Calendrier d'événements ▪ Glossaire ▪ Links manager ▪ Chat et Instant Messaging (communication synchrone) ▪ Gestionnaire de photos et de musique ▪ Content management qui permet d'entrer de l'information structurée ▪ Hypertextes collaboratifs (Wikis) ▪ Recommandations ▪ ....
Utilitaires additionnels pour utilisateurs	▪ Permissions selon outil et appartenance à un groupe ▪ Multi-langage ▪ Notifications par mail ▪ Information sur les utilisateurs (connexions, contributions, réputations) ▪ Système de réputation

3.3) Modules et blocs

Les portails communautaires sont constitués par des modules et des blocs.

Les modules représentent des applications complexes qui font appel généralement à une interface d'administration (accessible seulement aux administrateurs du site ou aux utilisateurs ayant des droits d'administration) et une interface utilisateur. L'interface administrateur permet de paramétrer les applications en fonction des buts et objectifs des activités qu'elles engendrent. Le module Galerie Photo, par exemple, a une interface administrateur qui gère les droits d'accès des utilisateurs aux différents albums photos du site. Du côté des utilisateurs, un membre n'ayant pas droits de lecture sur un album peut ne même pas savoir qu'il existe...

Les blocs sont par contre des cadres enveloppant des mini-programmes qui prennent place en général sur les parties latérales du portail. Un bloc a peu ou pas de fonctionnalités pour les utilisateurs simples. Ils servent à informer, à situer ou à mettre en valeur des informations importantes. Nous pouvons citer comme exemple de blocs le bloc « Favourite Links» que j'ai programmé pour le compte de TECFA en guise de familiarisation avec les portails PostNuke. Ce bloc permet d'afficher à droite du portail TECFA Seed les liens favoris du moment (Une description de ce bloc se trouve à l'annexe 1 de ce document page 64). Un autre exemple de blocs plus connu : le bloc « Qui est en ligne » qui affiche une liste des membres connectés au site. Les modules peuvent avoir leurs propres blocs.

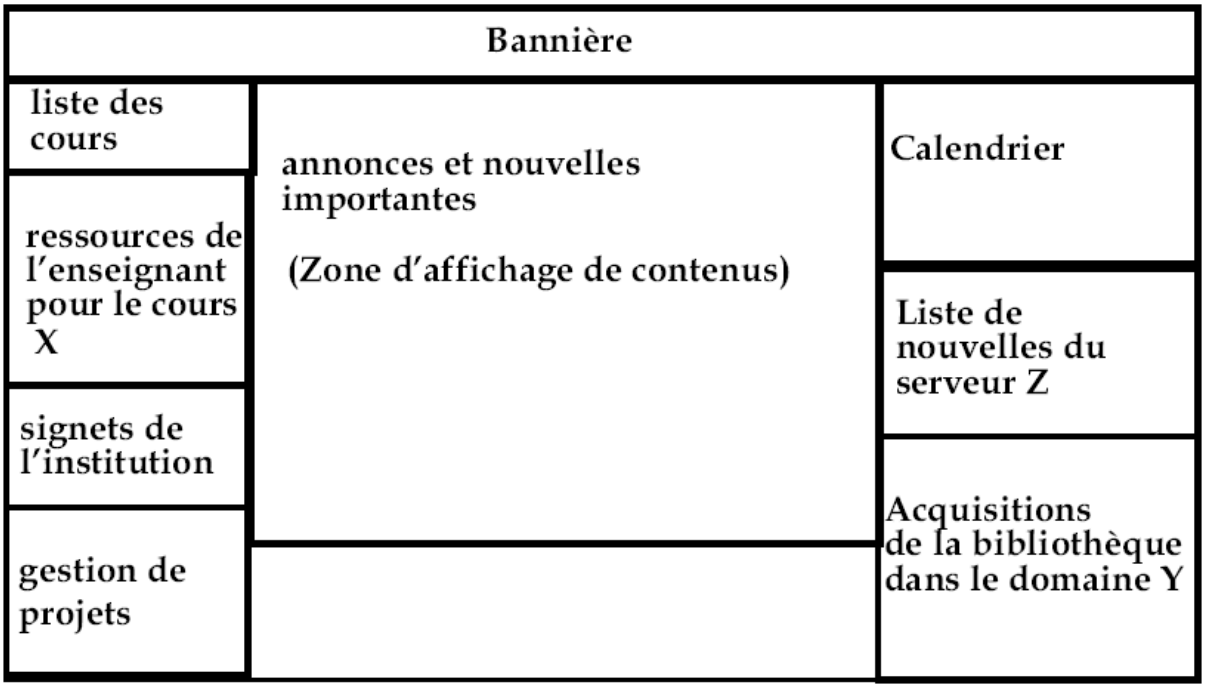


Figure 8 : Aspect général d'un portail communautaire

Un portail communautaire est en fait une mosaïque de blocs qu'un administrateur peut modeler à sa façon. La figure 7 illustre en gros l'aspect que peut avoir un portail communautaire.

4) Cahier des charges

Après avoir décrit les caractéristiques du script Argue Graph et les spécificités des portails communautaires, nous pouvons établir le cahier des charges du projet Argue Graph que l'on m'a attribué à TECFA.

Le but du projet est donc de concevoir et de mettre en place un module « ArgueGraph » pour des portails PostNuke.

Le module à concevoir devra offrir les services suivants :

- Une interface pour la gestion des questionnaires ;
- Une interface pour la modération et le déploiement du script côté administrateur ;
- Une interface pour le déploiement du script côté utilisateur ;
- Une interface pour l'échange de questionnaires sous forme de fichiers.

Le dernier point concernant l'échange de questionnaires est un service supplémentaire par rapport aux deux anciennes versions de l'Argue Graph. L'ajout de ce service est cependant naturel puisqu'il rentre dans le cadre des échanges et de la collaboration au sein de la communauté, principes fondamentaux des portails communautaires. Ce service d'échange permettra aux professeurs d'échanger des fichiers-questionnaire qu'ils pourront réutiliser, modifier et appliquer à leur propres étudiants.

Le module devra être sous la forme d'un package réunissant tous ses fichiers et répertoires ainsi que des fichiers explicatifs concernant son installation et son utilisation.

Chapitre 3 : Etude détaillée et conception

1) Introduction

Dans ce chapitre, nous allons tout d'abord spécifier les besoins du module Argue Graph à concevoir à travers une étude détaillée des fonctionnalités que devra offrir l'application. Nous enchaînerons ensuite par la modélisation des besoins fonctionnels de l'application qui représente en gros la partie conception du projet.

2) Spécification des besoins fonctionnels du module Argue Graph

2.1) Vue globale

L'objectif de l'application à développer consiste à offrir aux professeurs un outil qui leur permette de déployer le script Argue Graph et d'organiser ainsi plusieurs scénarios pédagogiques CSCL.

L'application devra supporter deux modes de déploiement :

- Le mode en ligne dans le cas où les apprenants devront discuter, collaborer et chercher des solutions communes à distance;
- Le mode face à face dans le cas où les apprenants sont « en classe ».

Ainsi, le professeur aura le choix de faire dérouler ses phases 1 ou 3 ou les deux en classe ou à « la maison ».

D'un autre coté, l'application étant destinée à être intégrée dans un portail communautaire du type «PostNuke», elle aura donc deux interfaces principales : une interface administrateur et une interface utilisateur.

L'interface administrateur accessible aux professeurs intégrera toutes les fonctionnalités nécessaires à la gestion et au déploiement du script Argue Graph.

Les fonctionnalités que devra offrir l'interface administrateur sont :

- Gestion des questionnaires : ajout, modification, suppression ;
- Gestion des scénarios d'utilisation : programmation et déprogrammation des scénarios, organisation des groupes d'utilisateurs pour chaque scénarios ;
- Modération des séances : formation de paires de partenaires pour le mode duo (phase 4), accès aux statistiques et aux résultats des différentes phases ;
- Gestion des téléchargements et les échanges de fichiers : création de fichiers XML à partir des questionnaires, publication de ces fichiers afin qu'ils puissent être téléchargés par d'autres professeurs.

L'interface utilisateur devra offrir à l'étudiant ou au participant d'un scénario Argue Graph les fonctionnalités suivantes :

- Une interface conviviale pour le déploiement du scénario : répondre au questionnaire, collaborer, analyser les résultats, etc. ;
- Une interface téléchargements : un espace réservé aux fichiers questionnaires XML publiés par les administrateurs du système.

2.2) Vue détaillée

Le module Argue Graph à concevoir vise deux catégories d'utilisateurs. D'un coté, les administrateurs (professeurs, modérateurs, assistants, etc.) et de l'autre, les utilisateurs simples (apprenants, étudiants, élèves, etc.).

Cette classification implique la nécessité de passer par un système d'authentification et de sécurisation afin de réserver les privilèges d'administration seulement aux personnes autorisées.

2.2.1) Système d'authentification, de permissions et de profils utilisateurs

Pour restreindre l'accès au module Argue Graph, il est nécessaire de prévoir un système d'authentification et de permissions. Le système d'authentification permettra de ne laisser l'accès au module qu'aux personnes connectées au portail. Le système de permissions restreindra l'accès à l'interface administrateur aux administrateurs et aux personnes habilités à cet effet ; l'interface utilisateur étant accessible à tout membre ayant une session ouverte sur le portail.

Nous ne nous attarderons pas sur cette partie du système puisque les portails communautaires sont dotés de systèmes d'authentification et de permissions qui gèrent les sessions des utilisateurs inscrits et les droits d'accès aux différents modules et interfaces. L'accès au module Argue Graph sera donc géré par le système d'authentification du portail qui l'hébergera et non pas par le module lui-même.

2.2.2) Interface administrateur

Les privilèges d'administration du module sont confiés automatiquement à l'administrateur du portail. D'autres membres du portail peuvent obtenir ces privilèges et ainsi accéder à cette interface. Les professeurs, les assistants ou toute personne voulant déployer des scénarios Argue Graph devra avoir accès à cette interface qui propose un grand choix de fonctionnalités.

Gestion des questionnaires

- Génération manuelle d'un questionnaire

L'administrateur fournit le nom, la description, une URL de soutien (optionnelle) et une amorce théorique (optionnelle) qui peut être visualisée par les apprenants soit au moment de répondre aux questionnaire où pendant le débriefing. Il doit aussi définir le nom des quatre extrémités des axes de son espace bidimensionnel (voir Chapitre2 au 2.1 Spécification et phases du script) et le poids de chacun des axes. Par défaut, chaque axe a un poids égal à 1. Ensuite, l'administrateur doit introduire ses questions et les réponses correspondantes. Pour chaque réponse, il peut définir une URL et une note de soutien. Il doit ensuite déterminer les coordonnées de chaque réponse sur le repère défini précédemment. Chaque réponse a des coordonnées 2D qui peuvent varier entre –1 et 1. Deux réponses peuvent avoir les mêmes coordonnées.

L'administrateur peut décider si oui ou non, ses URL et notes de soutien doivent être affichées au moment où l'élève répond à ses questions où au moment du débriefing.

Pour chaque questionnaire correct et validé, un fichier du questionnaire en XML est automatiquement crée par le module et placé dans un dossier à part pour la gestion des téléchargements (voir plus loin).

- Génération de questionnaires par copies

L'administrateur pourra copier un questionnaire existant dans la base de données. Ceci est possible en recopiant tous les champs du questionnaire source.

- Génération de questionnaires par fichiers XML

L'administrateur peut utiliser un questionnaire conçu par d'autres utilisateurs sur d'autres site. Ceci est possible à travers des fichiers questionnaire au format XML. Le module pourra télécharger ces fichiers, les décoder (parsing), les valider puis les introduire dans la base de données du système.

- Modification des questionnaires

L'administrateur aura la possibilité de modifier les questionnaires existants. Les modifications portent sur les caractéristiques du questionnaire (nom, description, etc.) et sur le contenu (questions et réponses).

Il est important de signaler que lorsqu'un questionnaire a été déjà utilisé ou lorsqu'il est en cours d'utilisation (ce qu'on qualifiera de questionnaire actif), il n'est pas possible de le modifier pour des raisons de cohérences. Dans ce cas là, il est possible de modifier le questionnaire en procédant à une recopie du questionnaire original (voir Génération de questionnaires par copies). La copie aura évidemment un nom différent du nom du questionnaire original.

- Suppression d'un questionnaire

Un questionnaire ne peut être supprimé que s'il n'est pas en cours d'utilisation. La suppression d'un questionnaire entraînera automatiquement la suppression de son historique d'utilisation et des graphiques qui lui appartiennent.

Gestion des séances Argue Graph

Nous appellerons séance d'utilisation ou séance Argue Graph, le déploiement d'un scénario Argue Graph sur un groupe d'apprenants. Cette partie gère donc les séances d'utilisation des questionnaires.

- Programmation d'une séance Argue Graph

L'administrateur doit sélectionner le groupe d'apprenants qui devra participer à la séance Argue Graph. Le nombre de participants doit impérativement être pair. L'administrateur devra aussi déterminer qui modérera la séance. En général, c'est le professeur qui a cette fonction. Au cas où le nombre d'utilisateurs est impair, le modérateur sera amené à participer à la

séance Argue Graph pour laquelle il aura deux rôles : celui du modérateur et celui de l'apprenant.

Une fois tout cela validé, la séance est programmée et restera en état d'attente d'activation. Les questionnaires ne peuvent être utilisés deux fois par le même groupe d'utilisateurs.

- Déprogrammation et suppression d'une séance Argue Graph

L'administrateur ne peut déprogrammer une séance que si elle n'est pas entamée. Il peut cependant supprimer une séance à n'importe quel moment. Dans ce cas, tout ce qui se rapporte à la séance, comme par exemple les résultats ou les graphiques, seront automatiquement supprimés.

Modération des séances

Cette fonctionnalité est celle qui se rapporte le plus au déploiement du script Argue Graph. En effet, la modération d'une séance Argue Graph consiste à lancer et à suivre le déroulement des différentes phases du scénario, choisir et valider la répartition des groupes en paires lors de la phase 3 et temporiser l'affichage des résultats et statistiques.

Il est donc nécessaire de prévoir une classification des différents statuts du scénario qui nous permette de déterminer l'état d'avancement d'une séance Argue Graph.

- Statut des séances Argue Graph

Une séance Argue Graph passe par plusieurs états qui dépendent en fait des cinq phases du script. Une séance programmée mais pas encore entamée sera notée « En attente ». Une séance achevée sera marquée par « Terminée ». Une séance entamée mais non terminée sera notée « En cours ».

Comme la séance « En cours » passe par plusieurs phases, il est judicieux de subdiviser ce statut en plusieurs états secondaires qui permettront à l'administrateur de savoir exactement où en sont les choses. Nous dressons un tableau qui explicite exactement les différents états et sous-états d'une séance Argue Graph. Nous y décrivons aussi les conditions nécessaires pour passer d'un état à un autre.

Etat	Code	Sous-état	Code	Conditions de passage
En attente	0			
Entamée		Phase 1 en cours (Mode Solo)	11	Activation de la séance
		Phase 1 terminée Résultats non communiqués	110	Tout le groupe a répondu à toutes les questions
		Résultats de la phase 1 En attente de la phase 2	119	Résultats publiés Répartition en paires validée
		Phase 2 en cours (Mode Duo)	12	Activation de la Phase 2
		Phase 2 terminée (Mode Duo) Résultats non communiqués	120	Toutes les paires ont répondu à toutes les questionnaires
Terminée	2			Publication des résultats finaux

Tableau 1: Les différents états d'une séance Argue Graph

- Répartition du groupe en paires de partenaires

Après avoir terminé l'étape solo, les apprenants ne pourront passer à l'étape duo qu'après que le professeur ait validé la répartition du groupe en paires. Une liste de paires est automatiquement générée par un algorithme du module une fois la phase 1 terminée. Cette liste pourra être utilisée par le professeur. Néanmoins, celui-ci pourra s'il le veut, répartir le groupe en paires manuellement.

- Résultats et statistiques

L'affichage des résultats est très important lors des deux phases du script. Un bon affichage des résultats de chaque phase argumenté par des statistiques, ne peut que favoriser et enrichir davantage les débats et donc l'apprentissage collaboratif.

Lors de l'étape solo, les résultats seront affichés sous forme de graphique. Le graphe représentera la position de chaque participant par rapport au thème du questionnaire. Nous appellerons ce graphe le **graphe solo**.

Lors de l'étape duo, l'affichage des résultats comprendra la liste de toutes les paires d'utilisateurs, un accès direct au détail de leurs réponses et arguments ainsi que le graphique final qui représentera les deux positions (Mode Solo et Mode Duo) de chaque participant par rapport au thème du questionnaire. L'accent sera mis sur la variation de la position de chaque utilisateur entre les deux phases. Nous appellerons ce second graphe le **graphe duo**. Un récapitulatif du questionnaire ainsi que des statistiques sur les réponses émises lors de la séance alimenteront davantage le débat.

Les résultats de chaque groupe, sauf en cas de suppression par l'administrateur, sont sauvegardés et sont accessibles depuis l'historique des questionnaires.

Gestion des groupes d'utilisateurs

Cette fonctionnalité n'était pas prévu au départ. Cependant, nous avons pensé que les professeurs auront besoin d'utiliser des groupes d'étudiants qui ne varient pas souvent. En effet, un professeur ayant par exemple trois classes d'étudiants aura, pour des raisons pratiques, tendance toujours recours aux mêmes trois groupes pour ses scénarios Argue Graph. C'est pour cela que le module intégrera des fonctionnalités de gestions de groupes d'utilisateurs.

Les membres des groupes d'utilisateurs seront puisés depuis la table des utilisateurs du portail. La réutilisation des mêmes groupes pour d'autres scénarios en sera davantage facilitée.

Le gestion des groupes comprendra des fonctionnalités d'ajout, de modification et de suppression de groupes.

Gestion des téléchargements

Pour renforcer l'aspect de collaboration et d'échange communautaire, il était donc nécessaire de prévoir un outil qui puisse permettre aux professeurs de s'échanger leurs expériences Argue Graph. Ceci sera possible grâce au gestionnaire de téléchargements.

Nous savons que lorsque les questionnaires sont validés, des fichiers XML leur correspondant sont automatiquement générés. L'administrateur peut décider si oui ou non, le fichier peut être publié sur l'interface utilisateur afin qu'il puisse être téléchargé par la communauté utilisant Argue Graph.

Le gestionnaire de téléchargements est bijectif aussi dans la mesure où il permet de décoder des fichiers questionnaire XML, de les introduire dans la base de données du système et donc d'utiliser des questionnaires « étrangers ».

2.2.3) Interface utilisateur

Cette interface est accessible en ouvrant une session sur le portail communautaire et ne nécessite aucune permission particulière. Au lancement de cette interface, l'utilisateur (apprenant, élève, étudiant, etc.) a accès à une liste de toutes les séances où il est ou a été programmé.

Le statut de chaque séance affiché sur cette liste (voir tableau de la page 27) permettra de diriger l'apprenant vers une fonctionnalité de l'interface utilisateur.

Si l'état de la séance est « En attente » (Etat 0), l'apprenant ne pourra rien faire. Autrement, à chaque autre état ou sous-état correspond une fonctionnalité que nous décrivons dans ce qui suit.

Répondre au questionnaire en phase 1 (mode solo) : Etat11

L'interface affiche une question à la fois. La page de réponse comprendra la question, les réponses correspondantes, une zone de texte pour les arguments et, si applicable, des notes et des URL de soutien. L'utilisateur peut répondre aux questions de façon aléatoire. Il peut modifier ses réponses tant qu'il n'a pas validé et envoyé sa copie de réponse au serveur.

Récapitulatif des réponses de la phase 1 (mode solo) : Etat 110

Tant que les autres utilisateurs n'ont pas terminé de répondre à la phase 1 et que l'administrateur n'a pas validé la répartition des utilisateurs en paires et publié les résultats, l'interface utilisateur affichera un récapitulatif des questions, réponses et arguments du participant.

Résultats de la phase 1 (mode solo) : Etat 119

L'utilisateur n'accède à ses résultats que lorsque l'administrateur décide de publier les résultats de la phase 1. Les résultats se composent du graphe solo, d'un récapitulatif des questions, réponses et arguments du participant lors de la phase 1 et d'un tableau listant la répartition du groupe en vue de la phase duo.

Répondre au questionnaire de la phase2 (mode duo) : Etat 12

Deux options se présentent pour répondre au questionnaire : continuer sur la même station de travail ou continuer sur celle du partenaire, sachant qu'une seule copie de réponses sera envoyé au serveur. Si le travail est fait en classe et que le professeur insiste pour que les deux partenaires se mettent ensemble pour collaborer et envoyer les réponses communes, l'un des

deux participants quittera son poste et rejoindra son partenaire. Dans ce cas là, il décidera de continuer sur la machine de son collègue. Si le travail se passe à distance, le module devra fournir un outil do communication entre les deux partenaires comme par exemple un simple chat box. Là aussi, l'un des deux partenaires décidera de répondre aux questions pour la paire.

Répondre aux questions à deux se déroule comme en mode solo, sauf qu'en plus d'avoir les questions et les réponses, les deux partenaires ont accès à leurs réponses et arguments respectifs de la phase 1. Ceci a pour but de créer un conflit d'opinion entre les deux partenaires qu'il faudra résoudre en collaborant afin d'envoyer une réponse et une argumentation communes.

Récapitulatif des réponses de la phase 2 (mode duo) : Etat 120

Tant que les autres paires d'utilisateurs n'ont pas terminé de répondre à la phase 2 et que l'administrateur n'a pas publié les résultats, l'interface utilisateur affichera un récapitulatif des questions, réponses et arguments de la paire en phase 2 et des deux partenaires en phase 1.

Résultats et statistiques finaux : Etat 2

L'utilisateur n'accède aux résultats finaux que lorsque l'administrateur décide de les publier.
Les résultats finaux se composent du graphe duo, d'un récapitulatif des toutes réponses et argumentations lors des deux phases, d'une liste des paires qui ont répondu et d'un lien vers une interface statistique.

3) Description de la conception

Une fois les besoins fonctionnels recensés, nous pouvons entamer la phase de conception. Pour la conception du module Argue Graph nous avons opté pour la modélisation UML (Unified Modeling Language). Les raisons qui ont motivé notre choix sont les suivants :

- UML permet de bien cadrer l'analyse en représentant le système selon différentes vues complémentaires aux diagrammes ;
- UML permet d'exprimer et d'élaborer des modèles objet, indépendamment de tout langage de programmation.

Dans cette partie, nous allons représenter une perspective du module à travers quelques diagrammes UML qui nous serviront par la suite de support pour la phase de réalisation.

3.1) Modélisation du contexte du système

Dans cette étape de modélisation, il s'agit de décrire le contexte du système. Pour se faire, il faut identifier les acteurs qui vont agir sur le système et les messages qu'ils vont échanger avec le module.

3.1.1) Les acteurs

Un acteur est une entité externe qui agit sur le système (opérateur, autre système...) en consultant son état ou en le modifiant. En réponse à l'action d'un acteur, le système fournit un service qui correspond à son besoin. Les acteurs peuvent être classés (hiérarchisés).

Les acteurs (utilisateurs) qui vont interagir avec le module Argue Graph sont:

- L'administrateur : c'est le professeur, assistant ou encadreur qui va gérer les questionnaires et modérer les séances Argue Graph.
- Les participants : ce sont les apprenants en général qui vont participer aux séances Argue Graph.

Un administrateur peut participer aux séances Argue Graph. Dans ce cas, il endosse les deux rôles prévus pour les utilisateurs de ce module.

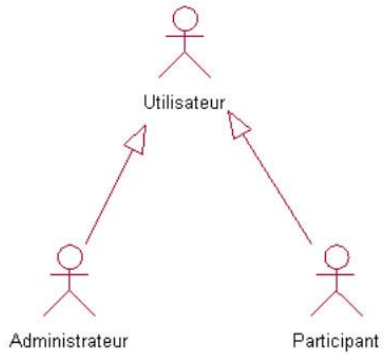


Figure 9 : Les acteurs du module Argue Graph

Le figure 8 nous donne déjà quelques indications sur l'une des classes de notre module : la classe utilisateur (user).

3.1.2) Les messages

L'interaction entre le système et les acteurs se fait à travers les messages. Un message envoyé par un acteur permettra de déclencher une série d'événements dans le système. L'activité du système générera un résultat transmis à l'acteur sous forme de message. Le diagramme de contexte dynamique de l'application permet de visualiser les interactions de l'application avec les acteurs.

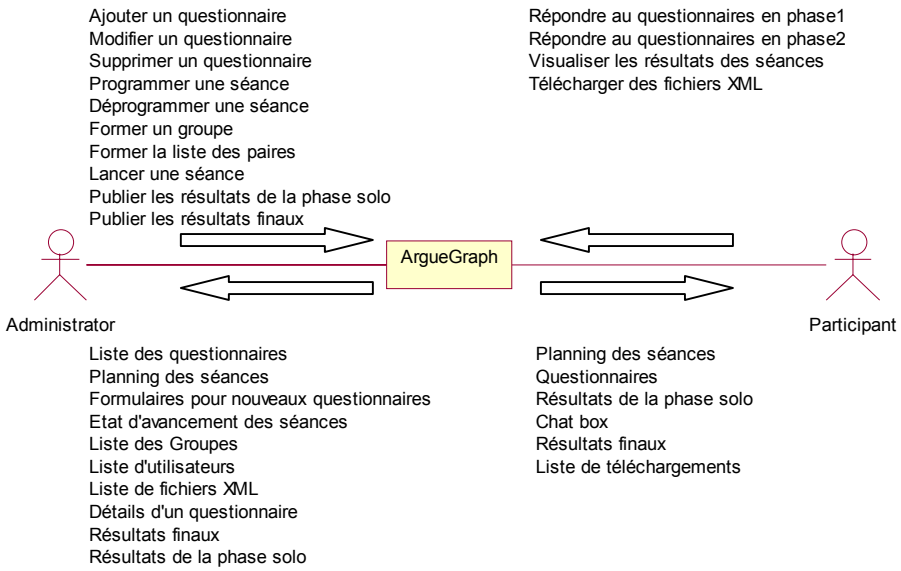


Figure 10 : Diagramme de contexte dynamique de l'Argue Grpah

3.2) Modélisation des besoins fonctionnels

La modélisation des besoins fonctionnels est la première étape du processus de développement orienté objet. Elle permet de compléter le premier recensement des besoins (effectué pendant l'étude préliminaire) en identifiant les différents cas d'utilisation et ceci en les décrivant et organisant en blocs fonctionnels. Cette étape permet aussi de préparer l'analyse orientée objet en identifiant les classes candidates pour chaque cas d'utilisation

3.2.1) Identification des cas d'utilisation

Les cas d'utilisations (use cases) permettent de structurer les besoins des utilisateurs et les objectifs correspondants d'un système. Ils centrent l'expression des exigences du système sur ses utilisateurs : ils partent du principe que les objectifs du système sont tous motivés. Les cas d'utilisations se limitent aux préoccupations "réelles" des utilisateurs ; ils ne présentent pas de solutions d'implémentation et ne forment pas un inventaire fonctionnel du système.

Les cas d'utilisation servent donc à identifier l'interaction des utilisateurs du système (acteurs) avec le système. Ils permettent de classer les acteurs et structurer les objectifs du système. Ils servent de base à la traçabilité des exigences d'un système dans un processus de développement intégrant UML.

Les uses cases peuvent être organisés en paquetages (packages). Ils regroupent des éléments de modélisation, selon des critères purement logiques. Les paquetages peuvent donc regrouper plusieurs cas d'utilisations qui partagent les mêmes ressources, les mêmes utilisateurs, le même thème ou la même philosophie.

Pour le module Argue Graph, deux grands paquetages principaux se sont dégagés :

- Administration du module ;
- Déploiement du script.

3.2.1.1) *Package « Administration du module »*

Le package « Administration du Module » regroupe les cas d'utilisations qui sont réservés aux administrateurs du module. Ils sont au nombre de trois.

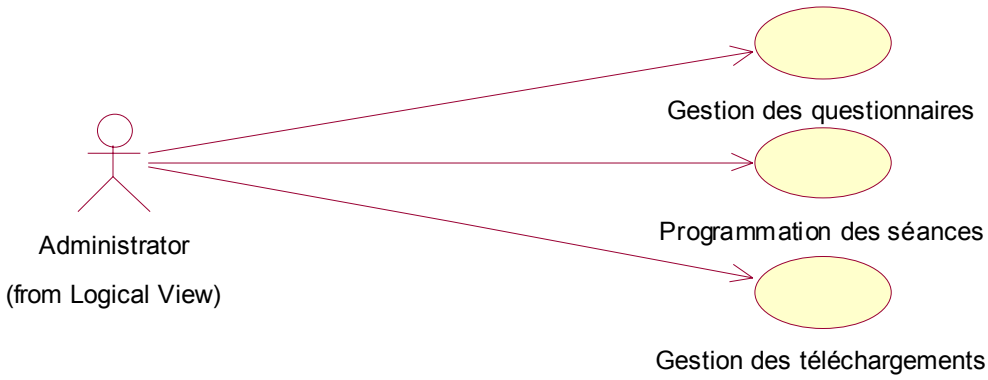


Figure 11 : Package « Administration du module »

Gestion des questionnaires

Ce cas d'utilisation regroupe toutes les fonctionnalités de gestion des questionnaires : ajout manuel, ajout par copie, ajout par fichier XML, modification, suppression, validation et activation.

Programmation de séances

Ce cas d'utilisation regroupe toutes les fonctionnalités qui se rapportent à la programmation des séances. Nous pouvons encore subdiviser ce cas d'utilisation en deux sous-parties :

- Gestion des groupes de participants : ajout, suppression et modification des groupes ;
- Programmation des séances : programmation et déprogrammation.

Gestion des téléchargements

Ce cas d'utilisation regroupe toutes les fonctionnalités qui gèrent les échanges des fichiers-questionnaire XML : publication d'un fichier, suppression d'un fichier et téléchargement d'un fichier.

3.2.1.2) *Package « Déploiement du script »*

Ce package regroupe les cas d'utilisation directement impliqués dans le processus de déploiement du script Argue Graph. Les acteurs de ce package sont les administrateurs pour la modération des séances et les participants aux séances Argue Graph pour le déploiement du script côté apprenant.

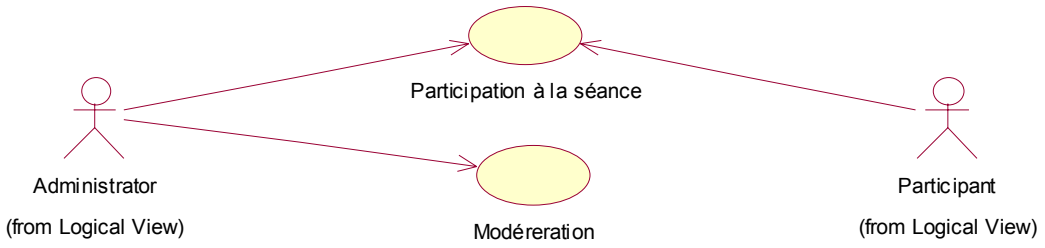


Figure 12 : Package « Déploiement du script »

Participation à la séance

Ce cas d'utilisation regroupe les fonctionnalités nécessaires au déploiement du script côté participant : formulaire de réponse en mode solo, affichage des résultats de la phase solo, formulaire de réponse en mode duo et affichage des résultats et statistiques finaux.
L'administrateur peut participer à l'Argue Graph en tant que participant simple.

Modération

La modération est exclusive à l'administrateur. Les fonctionnalités de ce cas d'utilisation sont donc: le suivi de la progression du script (pendant toute la durée du scénario), la répartition du

groupe d'apprenants en paires, la publication des résultats de la phase solo et la publication des résultats finaux.

3.2.2) Diagrammes de collaboration

Les diagrammes de collaboration décrivent des interactions entre objets qui peuvent être des instances de classes ou des acteurs. Ils peuvent nous aider à mieux cerner les classes qui vont participer à chaque package. Un diagramme de collaboration peut être obtenu à partir d'un digramme de séquence qui est plus simple à réaliser. En effet, les diagrammes de séquences permettent de représenter des collaborations entre objets d'un point de vue temporel, on y met l'accent sur la chronologie des envois de messages. Ils sont ainsi plus simples à réaliser étant donné qu'ils illustrent les cas d'utilisation et qu'ils se rapprochent le plus de la façon dont le programmeur conçoit le déroulement chronologique de son application.

Dans ce qui suit, nous présentons quelques diagrammes de collaboration pour chaque package afin d'en déduire les classes participantes au module.

3.2.2.1) *Package « Administration du module »*

La création d'un nouveau groupe de participants est une des fonctionnalités du package « Administration du module ». Le diagramme de séquence suivant récapitule chronologiquement les interactions entre les différentes classes du module pour aboutir à la création et à la sauvegarde d'un nouveau groupe de participants.

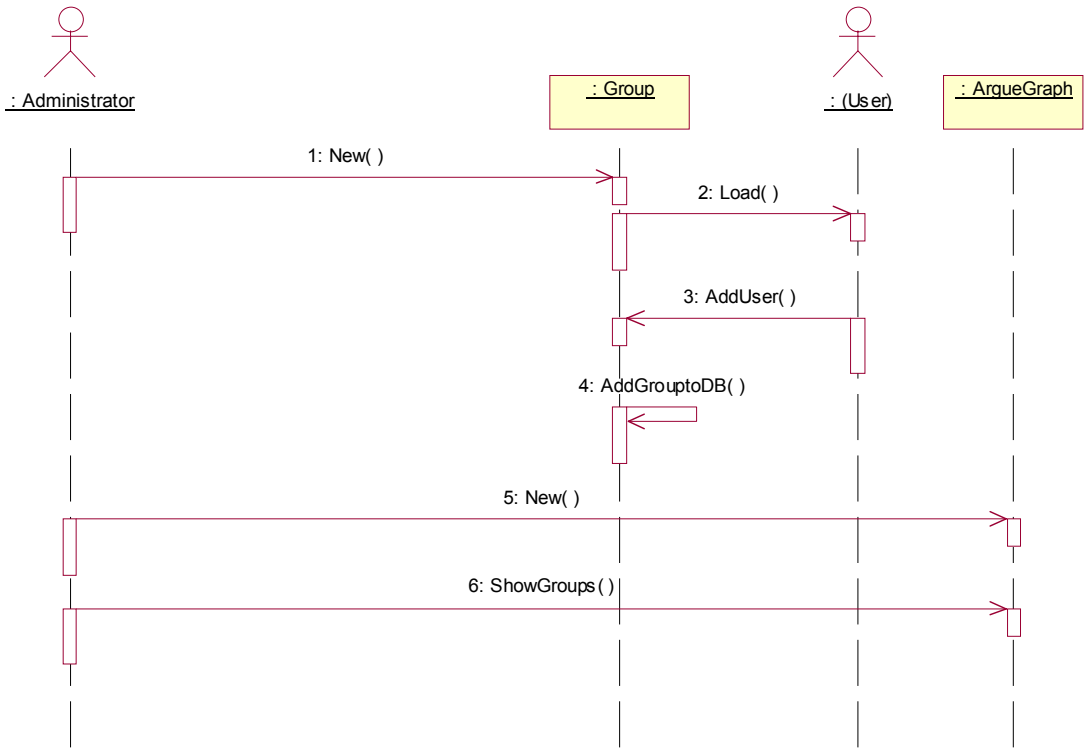


Figure 13 : Diagramme de séquence «Nouveau groupe »

Trois classes participent à cette fonctionnalité : la classe utilisateur, la classe groupe et la classe Argue Graph. Des instances de la classe « user » sont crées et chargées à partir du code de l'utilisateur. Chaque « utilisateur » est rajouté à tableau d'un « Group ». La liste est ensuite sauvegardée dans la base de donnée à l'aide de la méthode « AddGrouptoDB » (en 4). Le tout est affiché par une instance de la classe « ArgueGraph ».

Le diagramme de collaboration « Nouveau groupe » est déduit du diagramme de séquence.

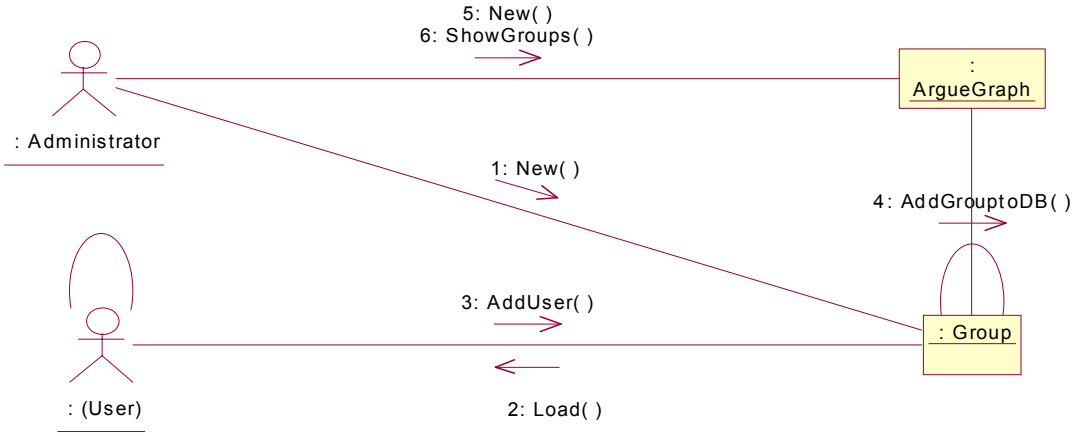


Figure 14 : Diagramme de collaboration «Nouveau groupe »

3.2.2.2) *Package « Déploiement du script »*

Pour ce package, nous présentons deux diagrammes de collaboration. Le diagramme de collaboration suivant concerne la fonctionnalité « Répondre en mode solo » et plus exactement le processus d'affichage du formulaire de réponse à une question donnée en mode solo.

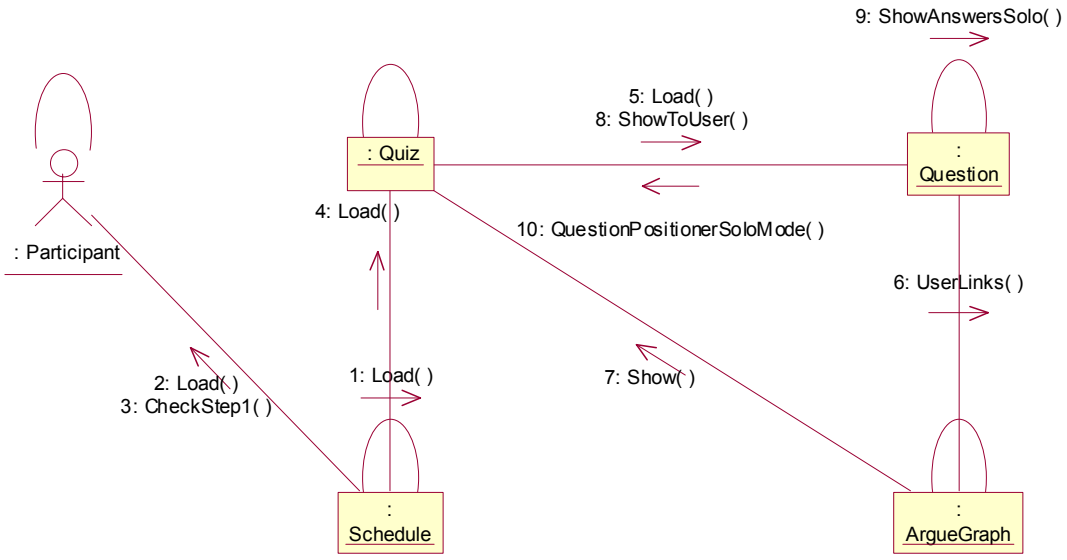


Figure 15 : Diagramme de collaboration «Répondre en mode solo »

Pour ce diagramme, cinq classes sont utilisées. Afin d'afficher le formulaire de réponse, une instance de classe « user » représentant un participant vérifie que celui-ci n'a pas terminé la phase solo d'une séance donnée (classe « Schedule ») à l'aide de la méthode « CheckStep1 » (en 3). Dans le cas où il n'a pas encore renvoyé sa copie des réponses, un objet « ArgueGraph » affiche le formulaire de réponse (« ShowAnswersSolo » en 9) .

Le diagramme de collaboration « Répondre en mode duo » est un peu plus complexe.

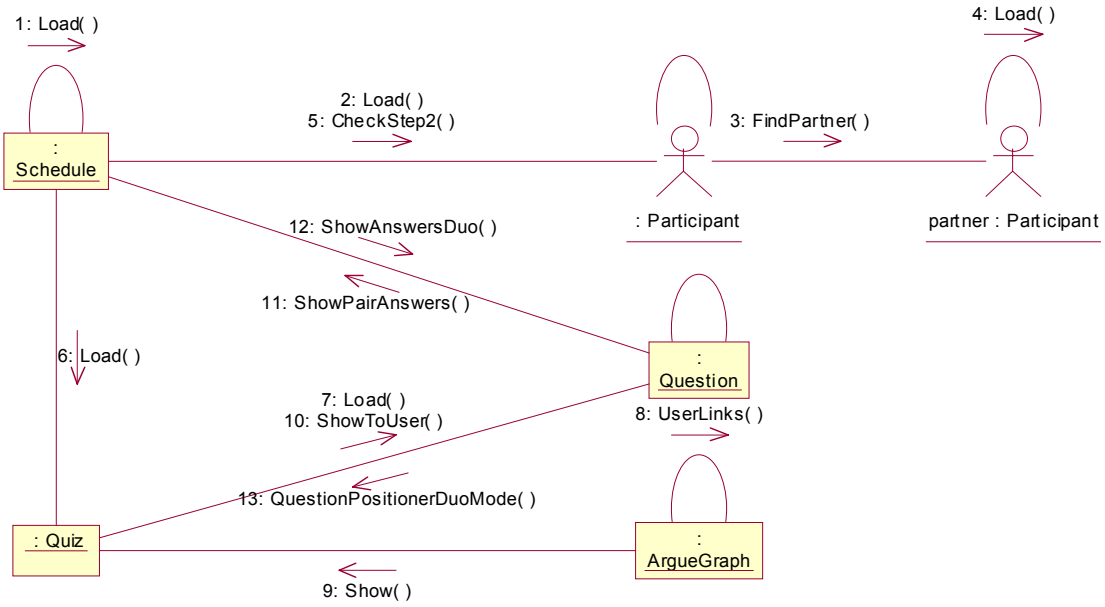


Figure 16 : Diagramme de collaboration «Répondre en mode duo »

Une instance de la classe « user » représentant un participant donné détermine d'abord son partenaire du mode duo grâce à la méthode « FindPartner » (en 3). Ensuite, la méthode « CheckStep2 » (en 5) vérifie que le participant n'a pas validé sa copie des réponses pour le mode duo. Si ceci s'avère vrai, une instance « ArgueGraph » affiche le formulaire de réponse (« ShowAnswersDuo » en 12) ainsi que les réponses individuelles de chaque membre de la paire lors du mode solo (« ShowPairAnswers » en 11).

3.2.3) Identification des classes

Les diagramme de collaboration et les packages des cas d'utilisation nous permettent de déterminer les classes qui vont être utilisées dans le module Argue Graph.

Classe	**Description**
ArgueGraph	Construction et affichage des interfaces.
Quiz	Classe relative aux questionnaires : gestion et affichage.
Question	Classe relative aux questions : gestion et affichage.
Answer	Classe relative aux réponses : gestion et affichage.
User	Classe relative aux utilisateurs du portail : identification.
Group	Classe relative aux groupes de participants : gestion et affichage.
Schedule	Classe relative aux séances Argue Graph : gestion et contrôle du déroulement des séances, affichage des résultats et création de graphes.
Results	Classe relative aux scores : calcul des résultats de chaque étape.
XML	Classe relative aux fichiers-questionnaire XML : gestion et décodage.

Pour le package « Administration du module » les classes participantes sont traduites sur la figure suivante :

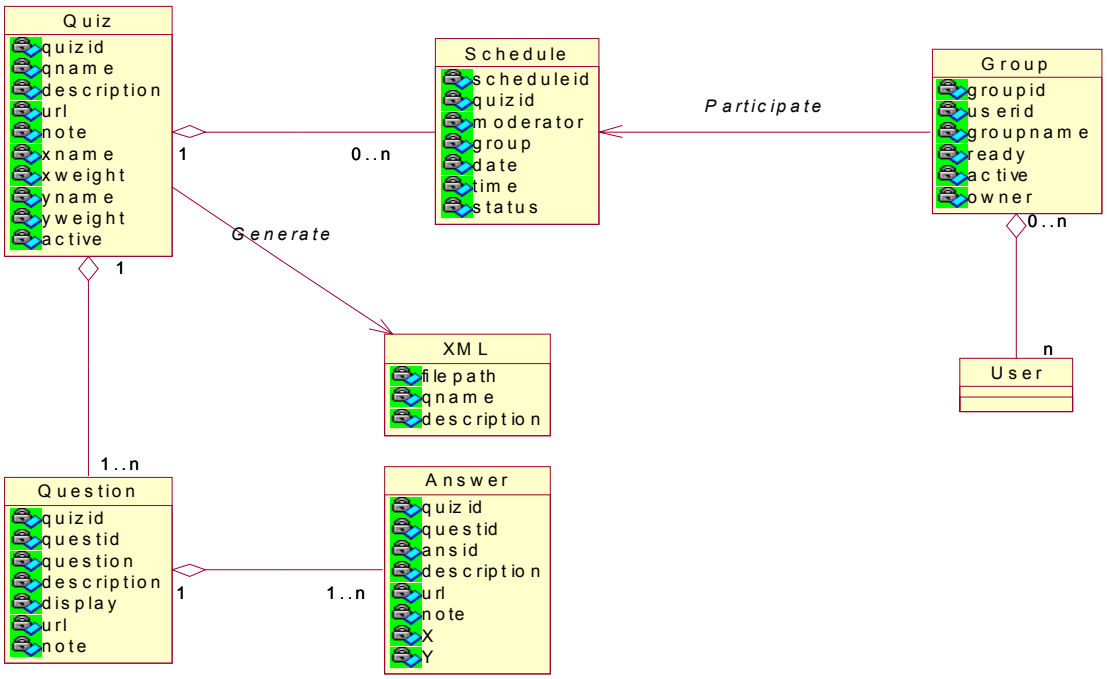


Figure 17 : Diagramme de classes du package « Administration du module »

En ce qui concerne le package « Déploiement du script », les classes participantes sont les suivantes :

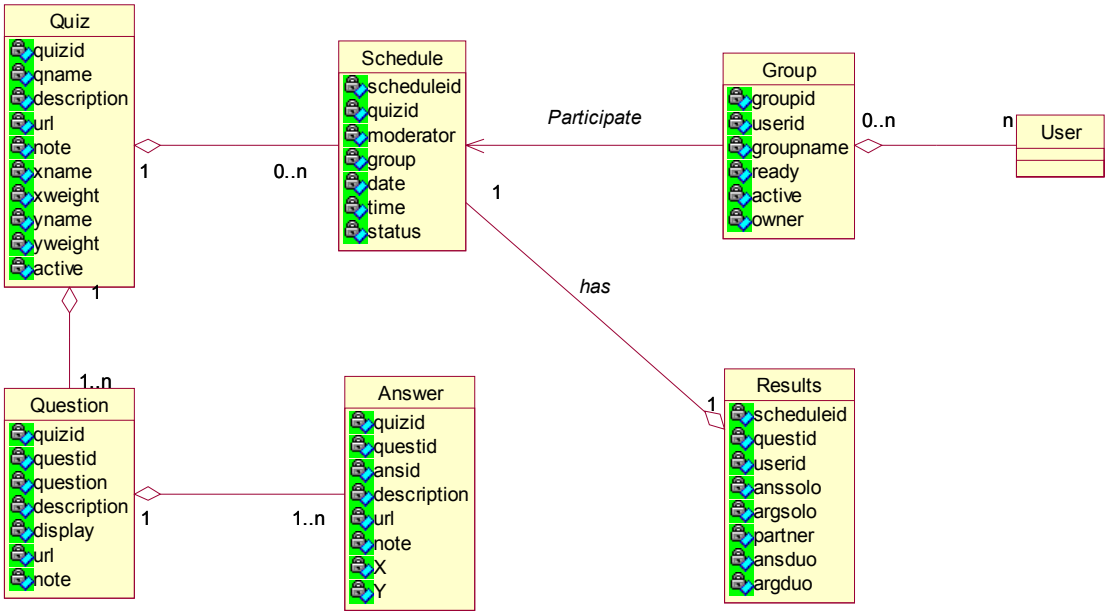


Figure 18 : Diagramme de classes du package « Déploiement du script »

3.2.4) Diagramme de déploiement

Les diagrammes de déploiement montrent la disposition physique du matériel qui compose le système et la répartition des composants sur ce matériel. Les ressources matérielles sont représentées sous forme de nœuds.

Les nœuds sont connectés entre eux, à l'aide d'un support de communication. La nature des lignes de communication et leurs caractéristiques peuvent être précisées. Ils peuvent montrer des instances de nœuds (un matériel précis), ou des classes de nœuds.

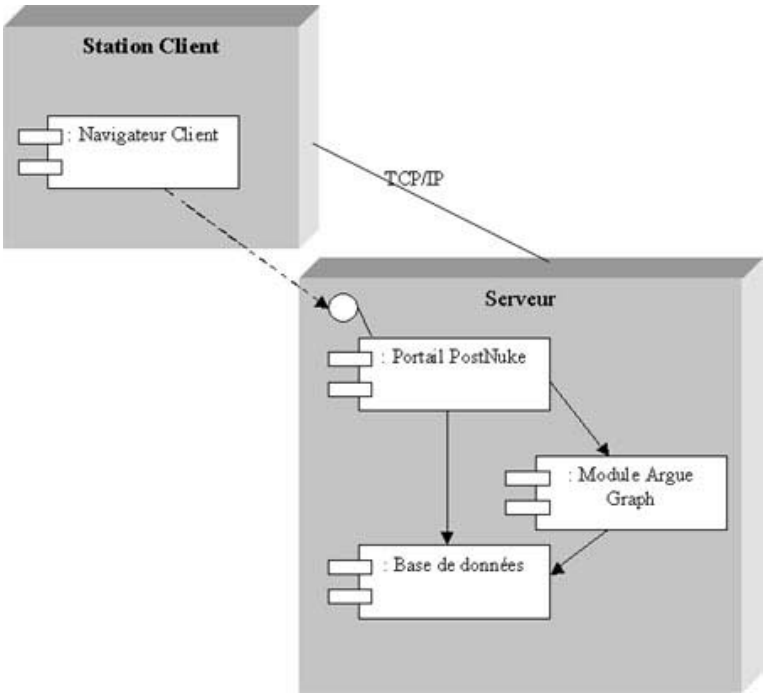


Figure 19 : Diagramme de déploiement du module Argue Graph

La figure 19 illustre clairement la relation triangulaire entre le portail, le module et la base de données. Ainsi, un client qui se connecte au serveur et qui veut accéder au module Argue Graph doit tout d'abord s'authentifier au près du portail. Cette authentification nécessite une connexion entre le portail et la base de données. Une fois authentifié, le portail appelle le module Argue Graph qui se charge en information en se connectant de nouveau à la base de données.

Chapitre 4 : Réalisation

1) Introduction

Une fois l'étude conceptuelle de l'application achevée, nous entamons la description de la phase de réalisation où nous spécifierons les différents outils utilisés et les méthodologies les plus intéressantes qui vont permettre la mise en place du module Argue Graph.

2) Environnement technique

Nous présentons dans cette partie l'environnement technique du module Argue Graph. Nous soulignons le fait que cet environnement est composé d'applicatifs dits Open Source. Le mouvement Open Source fait qu'il est possible à tous d'utiliser et de modifier librement un logiciel. En effet, quiconque est enclin à étudier le code source peut le modifier pour l'adapter à ses besoins. L'utilisation des logiciels libres va de paire avec l'esprit de collaboration et d'échange communautaire.

2.1) PHP

PHP (officiellement "PHP: Hypertext Preprocessor") est un langage de script HTML. Le langage PHP possède les même fonctionnalités que les autres langages permettant d'écrire des scripts CGI, comme collecter des données, générer dynamiquement des pages web ou bien envoyer et recevoir des cookies. Ce qui distingue le PHP des langages de script comme le Javascript est que le code est exécuté sur le serveur. Ainsi, il n'y a aucun moyen de distinguer les pages qui sont produites dynamiquement des pages statiques. La plus grande qualité et le plus important avantage du langage PHP est le support d'un grand nombre de bases de données. Réaliser une page web dynamique interfacant une base de donnés est donc extrêmement simple. Le langage PHP inclus aussi le support des services utilisant les protocoles tels que IMAP, SNMP, NNTP, POP3 ou encore http ainsi que de nombreuses librairies intégrées telles que la librairie graphique GD ou la librairie XML. Des connexions et interactions peuvent êtres supportés utilisant d'autres protocoles.

De plus, contrairement à d'autres langages comme Java, PHP est à la porté des non informaticiens qui savent programmer. Un diplômé en technologies éducatives pourra comprendre du code PHP mais ne pourra généralement pas intervenir sur du code Java plus complexe.

2.2) PostNuke

PostNuke est un projet de développement qui est apparu à partir d'un autre PHP appelé PHPNuke après que les utilisateurs ont exprimé un certain mécontentement à l'égard de certaines de ses fonctionnalités. PostNuke est un système de gestion de contenus (CMS : content management system) conçu avec le langage populaire PHP. CMS réduit le temps d'élaboration, tend à sophistiquer le panneau d'administration et sépare la forme du contenu permettant le travail dans un environnement structuré pour rapidement livrer des contenus tels que les articles, résumés, liens, nouvelles, temps, questions fréquemment posées, chat, gestionnaires de dossier tels que les sections de téléchargement…

La version actuelle de PostNuke (Février 2003) est la 0.723. Elle peut être téléchargée à partir du site www.postnuke.com. Ce site est aussi un espace de validation et d'échanges de modules et de blocs pour portails PostNuke.

Figure 20 : Logo PostNuke

Le bloc « Favourite Links » que nous avons développé (voir Annexe 1 page 64) y est référencé. Le module « Argue Graph » le sera bientôt.

Les principaux critères qui plaident en faveur de l'utilisation de PostNuke sont:

- Un parfait produit/exemple de la communauté open source (implémenté en PHP et MySQL, très populaires et standards) ;
- Architecture modulaire permettant de rajouter facilement de nouveaux logiciels modules ;
- Soutenu par une large communauté de développeurs ;
- Possède une interface multilingue (arabe compris).

Dans la partie « Développement », nous reviendrons plus en détail sur les principales caractéristiques de la programmation sous PostNuke.

2.3) MySQL

MySQL est un système de gestion de bases de données qui est très rapide, fiable et facile à utiliser. Il a été développé à l'origine pour gérer de très grandes bases de données beaucoup plus rapidement que des solutions déjà établies, et a été utilisé avec succès dans des conditions de productions critiques depuis plusieurs années. En développement constant, MySQL offre aujourd'hui un ensemble de fonctionnalités large et riche. Sa rapidité et sa sécurisation en font un outil idéal pour les applications Internet. C'est un système client/serveur qui est constitué d'un serveur SQL multi-threads, qui supporte différents clients et librairies, outils d'administration et interfaces.

3) Description du développement du module ArgueGraph

Dans cette partie, nous décrivons la structure standard de tout module « PostNuke » et les différentes caractéristiques techniques à retenir pour la phase de développement.

3.1) Caractéristiques du développement d'un module PostNuke

Le développement et la mise en place d'un module PostNuke ont été étudiés de manière à permettre d'une part une grande flexibilité de programmation et de déploiement aux programmeurs et d'une autre part une facilité d'accès au module par le portail qui l'héberge, par d'autres modules ou d'autres systèmes distants. PostNuke est, par exemple, doté d'une interface XML-RPC qui permet à d'autres systèmes d'obtenir de l'information directement prélevée dans un module sans passer par une interface Web.

Comme nous l'avons vu précédemment, PostNuke exige la séparation des fonctionnalités utilisateur et administrateur. Cette séparation permet de développer des modules plus proprement. Elle permet d'accélérer la réponse du système dans le sens où le module n'a besoin d'exécuter que la partie du code qui est nécessaire. Les objectifs de sécurisation à travers la séparation des fonctionnalités utilisateur et administrateur sont par ailleurs évidents.

Pour finir, PostNuke a prévu un répertoire nommé « modules » qui regroupe des sous-répertoires correspondant à chaque module hébergé par le portail. En dédiant un répertoire unique pour chaque module, PostNuke permet une meilleure maintenance de tout le système et facilite le processus d'installation et de désinstallation de modules tant pour le développeur que pour l'administrateur du site.

Nous allons nous intéresser de plus près à l'architecture de ce répertoire, étape cruciale avant de commencer le développement du module.

3.2) Architecture d'un répertoire module PostNuke

PostNuke a prévu une architecture standard pour la conception et le développement des modules qu'il va intégrer. Cette architecture est composée de fichiers et de répertoires standards nécessaires à tout module pour pouvoir être installé, configuré, utilisé et le cas échéant supprimé du portail. Le schéma suivant illustre l'architecture standard d'un répertoire module. Nous avons pris pour exemple le répertoire du module ePBL développé par TECFA Seed pour l'apprentissage par projet.

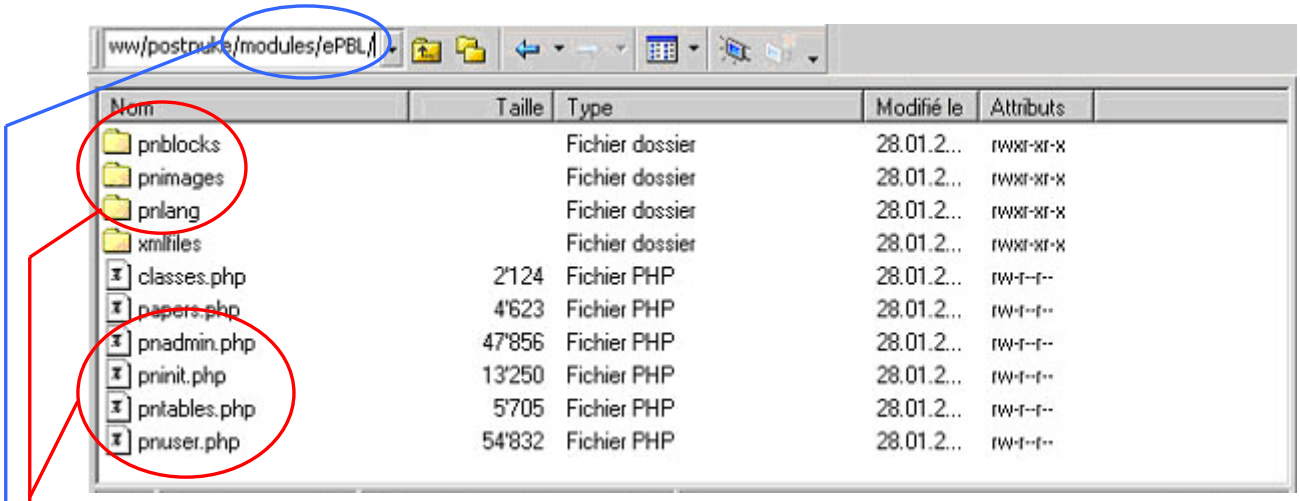


Figure 21: Architecture du répertoire d'un module

Le nom du module est très important. PostNuke détecte un nouveau module dès l'instant où un nouveau sous-réperoire est rajouté dans le répertoire « modules ». Le nom du répertoire servira aussi, comme nous le verrons plus tard, pour la reconnaissance des variables et des fonctions du module.

Les noms de fichiers et répertoires représentent les fichiers et répertoires les plus importants pour un module PostNuke. Nous précisons dans ce qui suit la fonctionnalité de chaque fichier ou répertoire :

pnadmin.php	fichier contenant toutes les fonctions d'administration du module ;
pnuser.php	fichier contenant toutes les fonctions utilisateur du module ;
pninit.php	fichier contenant les fonctions d'initialisation du module ;
pntables.php	fichier contenant toutes les informations concernant les tables de la base de donnée du module
pnimages	répertoire contenant toutes les images du module ;
pnlang	répertoire contenant les fichiers de traductions du module ;
pnblocks	répertoire contenant les fichiers concernant le bloc du module.

Le répertoire « pnlang » a aussi une structure particulière que nous révèlent les deux figures suivantes.

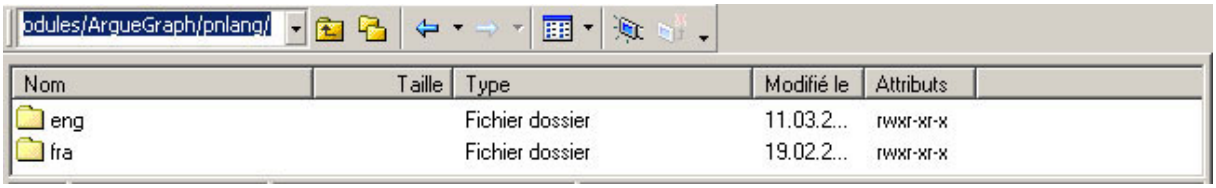


Figure 22 : Architecture du répertoire pnlang

Le répertoire « pnlang » doit contenir au moins un sous-répertoire qui corresponde à une langue donnée. Pour que le module « ArgueGraph » apparaisse en anglais, un sous-répertoire « eng » (figure 22) doit être créé. Pour qu'une langue donnée soit reconnue par le portail, le nom du répertoire qui la représente doit être conforme aux standards d'appellation PostNuke. Pour que le français soit pris en compte, le répertoire de traduction doit être nommé « fra » ; pour l'allemand « deu », etc. Le contenu de ce répertoire est explicité sur le schéma suivant :

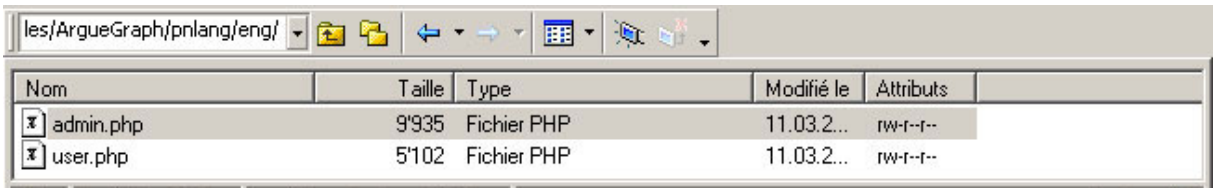


Figure 23 : Structure du répertoire « eng »

Un répertoire langue doit impérativement contenir les deux fichiers suivants :

admin.php contient la traduction du texte affichée par l'interface administrateur;
user.php contient la traduction du texte affichée par l'interface utilisateur.

Des fichiers et répertoires additionnels peuvent être rajoutés comme le montre la figure 21 (page précédente). Il est cependant impossible au système d'accéder aux fonctionnalités du module si les fichiers ou répertoires « obligatoires » ont des noms différents. Nous reviendrons plus loin sur les caractéristiques des fichiers énumérés dans cette partie.

3.3) L'API PostNuke

L'API PostNuke est un ensemble de fonctions relatives au système PostNuke qui permettent aux développeurs d'accéder facilement à de l'information spécifique au portail sans pour autant avoir le besoin de comprendre les mécanismes et l'architecture de celui-ci. L'API PostNuke permet donc d'accéder à des variables internes au portail PostNuke qui couvrent plusieurs domaines tels que les informations concernant les utilisateurs membres, les modules, la sécurité, les sessions, la base de données et l'affichage.

Les fonctions faisant partie de l'API PostNuke commencent toutes par le préfixe « pn ». La fonction « pnDBInit() » par exemple, permet d'initialiser une connexion à la base de données de PostNuke. Une fonction que nous avons beaucoup utilisée dans la phase de développement est « pnUserGetVar('uid') ». Elle permet d'obtenir le numéro de code de l'utilisateur qui est entrain de naviguer sur le portail. Dans le cas où une erreur se produit, le système génère une exception en retour grâce au gestionnaire d'erreurs.

3.3.1) Gestion des erreurs

PostNuke est capable de gérer les erreurs à l'aide d'un système de gestion d'exceptions puissant qui comble les lacunes du langage PHP. PostNuke classe les exceptions en deux catégories : les exceptions-système et les exceptions utilisateur. Les exceptions système sont utilisées par les fonctions de l'API PostNuke, mais peuvent tout aussi être utilisées par le programmeur. Par exemple, il est vivement conseillé d'utiliser l'exception DATABASE_ERROR quand une erreur de base de données se produit. Il est aussi recommandé d'utiliser l'exception BAD_PARM lorsque de mauvais paramètres sont passés aux fonctions des modules développés. Les exceptions utilisateur sont celles que les développeurs créent.

3.3.2) Statut de l'API PostNuke : en constante progression

L'API PostNuke ne comporte pas actuellement toutes les fonctionnalités que souhaiteraient les développeurs. Grâce à la communauté PostNuke, il est possible aux chercheurs et aux développeurs de proposer des idées nouvelles de fonctions encore non disponibles, de les développer, de les soumettre aux tests et aux essais, de les valider et de les rajouter à l'API existant. Les recherches sont en cours et l'API ne cesse de s'enrichir.

3.4) L'API pnHTML

PostNuke ne permet pas d'utiliser les balises HTML comme le permettrait une simple programmation de pages web en PHP. Parmi les raisons qui ont motivé ce choix se trouve le fait que les développeurs PostNuke ont décidé la séparation de l'information de la mise en forme dans le processus de programmation. Pour le portail PostNuke, les thèmes sont gérés par des fichiers spéciaux et des feuilles de styles CSS (Cascading Style Sheet). Cependant, il a fallu trouver une solution de substitution pour permettre aux développeurs de créer des formulaires et créer des liens. L'API pnHTML a été mis en place à cet effet.

pnHTML est en fait une classe qui possède plusieurs méthodes qui permettent de créer et de gérer des interfaces. Dans le tableau suivant, nous présentons quelques méthodes de la classe pnHTML et les balises HTML qui leur correspondent.

Méthode pnHTML	**Fonctionnalité**	**Balise HTML**
Text()	Écrire un paragraphe	
BoldText()	Écrire un paragraphe en gras	
URL()	Éditer des liens	<a>
TableStart()	Créer un tableau	<table>
TableAddRow()	Ajouter une ligne à un tableau	
TableEnd()	Fermer un tableau	</table>
FormStart()	Créer un formulaire	<form>
FormCheckbox()	Placer un champ formulaire de type Checkbox ou radio	<input type 'radio'> ou <input type 'checkbox'>

Tableau 2 : Equivalence des méthodes pnHTML et des balises HTML

Nous remarquerons que l'utilisation de ses méthodes pour la création d'interfaces se rapproche un peu de la programmation d'interfaces en Java.

3.4.1) Exemple de l'utilisation de la classe pnHTML

Dans l'exemple suivant, nous présentons une fonction qui permet de créer, de remplir et d'afficher un tableau.

```
function ShowTable(){

    include('tables.php');
    $output= new pnHtml();
    $output->TableStart('','',1,'100%');

    $output->SetOutputMode(_PNH_RETURNOUTPUT);

    $row1=array();
    $row1[]=$output->BoldText(_QUIZZ);
    $row1[]=$output->FormText('qname',"$this->qname",200,255);
    $output->SetOutputMode(_PNH_KEEPOUTPUT);
    $output->SetInputMode(_PNH_VERBATIMINPUT);
    $output->TableAddRow($row1,'left');

    $output->TableEnd();

    return $output->GetOutput();
}
```

Figure 24 : Exemple de l'utilisation de la classe pnHTML pour la création d'un tableau

Dans cet exemple, un objet « output » de la classe pnHTML représentant une nouvelle interface est créée. Les méthodes TableStart(), TableAddRow() et TableEnd() sont utilisées pour construire le tableau. Pour que le tout soit affiché, la méthode GetOutput() est appelée.

3.4.2) Statut de l'API pnHTML : progression lente

L'API pnHTML n'a pas beaucoup évolué depuis les premières versions de PostNuke. Aussi, la documentation est quasi inexistante ce qui est préjudiciable dans la mesure où certaines méthodes sont encore ambiguës. Dans la figure 24, deux méthodes sont utilisées pour déterminer la disposition des éléments graphiques sur l'interface. Ces méthodes sont « SetOutputMode() » et « SetInputMode() ». Elles agissent sur la manière d'interpréter le texte à afficher qui dépend des quatre constantes suivantes: _PNH_RETURNOUPUT, _PNH_VERBATIMINPUT, _PNH_KEEPOUTPUT, _PNH_PARSEINPUT. Prenons l'exemple de la figure 24 : si l'on oubliait la ligne de code encerclée, le champ formulaire texte serait placé en dehors du tableau. A ce jour, aucune explication n'est fournie concernant l'utilisation de ces méthodes ce qui fait que pour créer des interfaces, il faut procéder par hasard pour trouver la bonne configuration qui permette d'aboutir à une présentation quelque peu convenable.

Par ailleurs, plusieurs fonctionnalités manquent encore comme celle de pouvoir lancer des URL sur une nouvelle fenêtre du navigateur. Ceci est en partie lié au fait que la notion du « target=_blank » n'est plus supportée par le XHTML. Il est cependant possible de remplacer ces attributs par du JavaScript. Une autre fonctionnalité importante n'est toujours pas disponible : on ne peut pas afficher des images avec du pnHTML. Tout ceci nous a posé problème chez TECFA, tant dans le cadre du développement du module « ArgueGraph » que dans le cadre d'autres projets. Pour y remédier, nous avons créé deux nouvelles méthodes inspirées du pnHTML, la première pour afficher des images (myImg()) et la seconde pour lancer des liens dans de nouvelles fenêtres (myURL()). Une nouvelle classe nommée myClass, étendant la classe pnHTML, a été créée et a inclus ces deux méthodes. Nous avons aussi rajouté une troisième méthode hybride (myImgURL()) qui permet de créer des liens image (voir annexe). Ces fonctions nous ont permis d'ajouter une touche de convivialité aux interfaces conçues.

3.5) Développement du module « ArgueGraph »

Une fois les caractéristiques de l'environnement technique recensées, nous pouvons décrire les points les plus intéressants du processus de développement du module de l'Argue Graph.

3.5.1) Le répertoire « ArgueGraph »

Le répertoire du module « ArgueGraph » se présente ainsi :

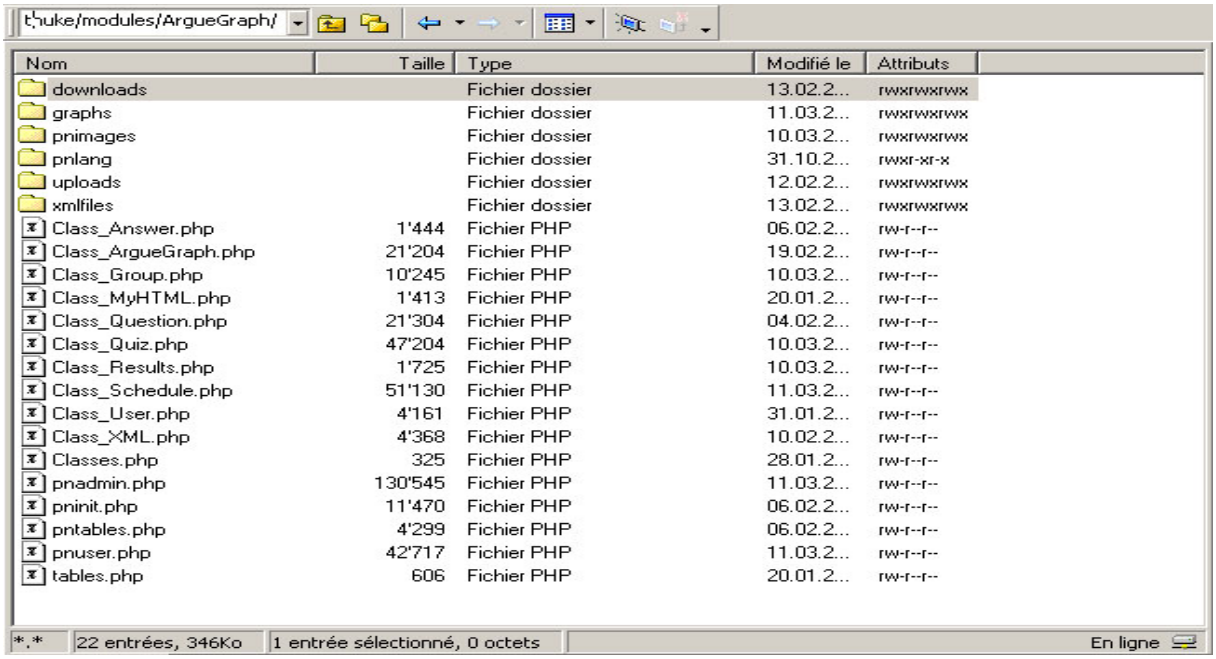


Figure 25 : Le répertoire "Argue Graph "

En plus des fichiers et répertoires traditionnels, nous avons prévu un fichier pour chaque classe issue de l'étude conceptuelle du système. Nous avons aussi rajouté quelques répertoires supplémentaires que nous décrivons dans ce qui suit.

- Le fichier « Class_MyHTML.php » se rapporte à la classe myClass qui étend la classe pnHTML. Le fichier contient les trois nouvelles méthodes rajoutées au pnHTML que nous avons décrit précédemment (au 3.4.2 de ce chapitre).
- Le répertoire « graphs » contient les graphes qui sont générés pour chaque séance Argue Graph. Les graphes sont maintenus tant que les séances sont sauvegardées dans l'historique de l'application.
- Le répertoire « xmlfiles » contient tous les fichiers-questionnaire en XML qui sont automatiquement générés lors de la validation d'un questionnaire.
- Le répertoire « downloads » contient tous les fichiers-questionnaire en XML qui sont destinés au téléchargement et qui vont être listés sur l'interface utilisateur.
- Le répertoire « uploads » contient tous les fichiers-questionnaire en XML externes qui ont été téléchargés par l'administrateur du module et qui vont être analysés, décodés et utilisés par le module.

D'autres fichiers vont être rajoutés ultérieurement : un fichier « ReadMe.txt » pour expliquer comment installer et désinstaller le module et un manuel d'utilisation sous forme de fichiers HTML.

3.5.2) Description du fichier « pntables.php »

Le fichier « pntables.php » est utilisé pour charger les tables du module. En fait, ce fichier se résume à une fonction unique « ArgueGraph_pntables() » qui permet de stocker les informations concernant les tables.

```
function ArgueGraph_pntables()
{
    // Initialise table array
    $pntable = array();

    // Get the name for the template item table.  This is not necessary
    // but helps in the following statements and keeps them readable
    $quizzestable     = pnConfigGetVar('prefix') . '_ArgueGraph_quizzes';
    $questionstable   = pnConfigGetVar('prefix') . '_ArgueGraph_questions';
    $answerstable     = pnConfigGetVar('prefix') . '_ArgueGraph_answers';
    $scorestable = pnConfigGetVar('prefix') . '_ArgueGraph_scores';
    $resultstable = pnConfigGetVar('prefix') . '_ArgueGraph_results';
    $schedulestable = pnConfigGetVar('prefix') . '_ArgueGraph_schedule';
    $groupstable = pnConfigGetVar('prefix') . '_ArgueGraph_groups';

    // Set the table name
    $pntable['ArgueGraph_quizzes']     = $quizzestable;
    $pntable['ArgueGraph_questions']   = $questionstable;
    $pntable['ArgueGraph_answers']     = $answerstable;
    $pntable['ArgueGraph_scores'] = $scorestable;
    $pntable['ArgueGraph_results'] = $resultstable;
    $pntable['ArgueGraph_schedule'] = $schedulestable;
    $pntable['ArgueGraph_groups'] = $groupstable;

    // Set the column names.  Note that the array has been formatted
    // on-screen to be very easy to read by a user.

    $pntable['ArgueGraph_quizzes_column'] =
array('quizid'   => $quizzestable . '.quizid',
      'qname'    => $quizzestable . '.qname',
      'description'=> $quizzestable.'.description',
      'url'      =>$quizzestable.'.url',
      'note'     =>$quizzestable.'.note',
      'xname'         =>$quizzestable.'.xname',
      'xname2'         =>$quizzestable.'.xname2',
      'xweight' =>$quizzestable.'.xweight',
      'yname'         =>$quizzestable.'.yname',
      'yname2'         =>$quizzestable.'.yname2',
      'yweight' =>$quizzestable.'.yweight',
      'active'   => $quizzestable . '.active');
```

Figure 26 : Entête du fichier "pntables.php"

Le schéma précédent montre les sept tables qui vont faire partie de la base de données du système ArgueGraph : « quizzes » pour les questionnaires, « questions » pour les questions, « answers » pour les réponses, « scores » pour les résultats finaux, « results » pour les résultats intermédiaires, « schedule » pour le planning des séances et « groups » pour la gestion des groupes. Deux préfixes sont rattachés au nom de chaque table : « ArgueGraph » et « nuke ». Ce dernier préfixe est général à toutes les tables du portail PostNuke et est retrouvé grâce à la fonction « pnConfigGetVar('prefix') »..

Les informations concernant les tables sont stockées dans une variable prédéfinie par le système qui est le tableau « pntable ». A chaque fois que la fonction « pnDBGetTables() » est appelée par une fonction du module, le système accède à ce fichier et charge cette variable.

Un module externe ou un bloc ne peut par contre pas accéder directement aux tables du module « ArgueGraph ». La fonction « pnModDBInfoLoad() » qui a pour paramètre d'entrée le nom du module cible, fait appel aux variables « ArgueGraph ». C'est seulement à ce moment là que l'on peut utiliser la fonction « pnDBGetTables() » et ainsi charger la variable « pntable ».

3.5.3) Description du fichier « pninit.php»

Le fichier « pninit.php » est utilisé deux fois dans la vie d'un module. La première fois pour initialiser le module dans le portail et la seconde fois pour le supprimer.

Pour initialiser le module, la fonction « ArgueGraph_init() » fait appel au fichier « pntables.php » pour charger la variable « pntable » qui servira à insérer les tables du système dans la base de données du portail.

```
function ArgueGraph_init()
{

    list($dbconn) = pnDBGetConn();
    $pntable = pnDBGetTables();

    $quizzestable = $pntable['ArgueGraph_quizzes'];
    $quizzescolumn = &$pntable['ArgueGraph_quizzes_column'];
    $sql1 = "CREATE TABLE $quizzestable (
                        $quizzescolumn[quizid] tinyint(3) unsigned NOT NULL auto_increment,
                        $quizzescolumn[qname] varchar(255) NOT NULL default '',
                        $quizzescolumn[description] varchar(255) default NULL,
                        $quizzescolumn[url] varchar(255) default NULL,
                        $quizzescolumn[note] varchar(255) default NULL,
                        $quizzescolumn[xname] varchar(255) default NULL,
                        $quizzescolumn[xname2] varchar(255) default NULL,
                        $quizzescolumn[xweight] tinyint(3) unsigned NOT NULL default '1',
                        $quizzescolumn[yname] varchar(255) default NULL,
                        $quizzescolumn[yname2] varchar(255) default NULL,
                        $quizzescolumn[yweight] tinyint(3) unsigned NOT NULL default '1',
                        $quizzescolumn[active] tinyint(1) default '0',
                        PRIMARY KEY  (quizid))";
    $dbconn->Execute($sql1);
    //Check for an error with the database code, and if so set an
    // appropriate error message and return
    if ($dbconn->ErrorNo() != 0) {
        pnSessionSetVar('errormsg', _CREATETABLE1FAILED);
        return false;
    }
```

Figure 27 : Initialisation du module

Ce schéma explicite le mécanisme d'insertion de la table « ArgueGraph_quizzes » dans la base de données. Une connexion avec la base de données du portail est d'abord établie à l'aide de la fonction « pnDBGetConn() » qui crée une instance de la classe « dbconn » prédéfinie par PostNuke. Ensuite, la variable « pntable » est chargée à l'aide de la fonction « pnDBGetTables() ». Enfin, la requête SQL1 pour la création de la table « ArgueGraph » est exécutée par la méthode « Execute() » de l'objet « dbconn » (figure 27). En cas d'erreur, une exception générée et un message est affiché pour l'administrateur.

	Table	Action						Records	Type	Size
☐	nuke_ArgueGraph_answers	Browse	Select	Insert	Properties	Drop	Empty	71	MyISAM	7.4 KB
☐	nuke_ArgueGraph_groups	Browse	Select	Insert	Properties	Drop	Empty	14	MyISAM	2.3 KB
☐	nuke_ArgueGraph_questions	Browse	Select	Insert	Properties	Drop	Empty	19	MyISAM	3.2 KB
☐	nuke_ArgueGraph_quizzes	Browse	Select	Insert	Properties	Drop	Empty	3	MyISAM	2.4 KB
☐	nuke_ArgueGraph_results	Browse	Select	Insert	Properties	Drop	Empty	75	MyISAM	3.5 KB
☐	nuke_ArgueGraph_schedule	Browse	Select	Insert	Properties	Drop	Empty	4	MyISAM	2.1 KB
☐	nuke_ArgueGraph_scores	Browse	Select	Insert	Properties	Drop	Empty	15	MyISAM	2.7 KB

Figure 28 : Les tables du module Argue Graph

Les erreurs d'initialisation du module surviennent en général lorsqu'il y a incohérence entre le fichier « pninit.php » et le fichier « pntables.php ». Il faut donc faire attention à ce que les noms des tables et de leurs champs correspondent dans les deux fichiers.

Pour supprimer le module, la fonction « ArgueGraph_delete() » fait appel au fichier « pntables.php » pour charger la variable « pntable ». Le processus de suppression de l' « ArgueGraph » se traduit par la suppression de ses tables de la base de données du portail et le nettoyage du système de toutes ses variables.

Le schéma suivant est un extrait de la fonction « ArgueGrpah_delete() ».

```
    $sql7 = "DROP TABLE $pntable[ArgueGraph_groups]";
    $dbconn->Execute($sql7);
    // Check for an error with the database code, and if so set an
    // appropriate error message and return
    if ($dbconn->ErrorNo() != 0) {
        pnSessionSetVar('errormsg', _DROPTABLE6FAILED);
        return false;
    }


    // Delete any module variables
    pnModDelVar('ArgueGraph', 'modname');

    // Deletion successful
    return true;

}
```

Figure 29 : Suppression du module

Les tables de l'« ArgueGraph » sont supprimées une à une avec un contrôle d'erreurs systématique. Les variables du modules sont supprimées par la fonction « pnModDelVar() ».

3.5.4) Principe de développement

Le fichier « pnadmin.php » regroupe toutes les fonctionnalités de l'administration du module. Cela va de l'interface d'ajout de questionnaires, à l'interface de publication des résultats des séances Argue Graph, en passant par l'interface d'échanges de fichiers XML. Pour un module aussi complexe que l'« ArgueGraph », tout mettre dans le fichier « pnadmin.php » peut s'avérer hasardeux sur le plan de la programmation. Il est évident que si l'on mettait le code complet de chaque interface ou fonction interne sur ce seul fichier, sa taille serait beaucoup trop importante et l'on s'y perdrait. Nous avons donc opté pour une solution d'allègement : il s'agit de subdiviser chaque fonction du module en un ensemble de briques d'affichage; chaque brique étant programmée sur l'un des fichiers-classes du module.

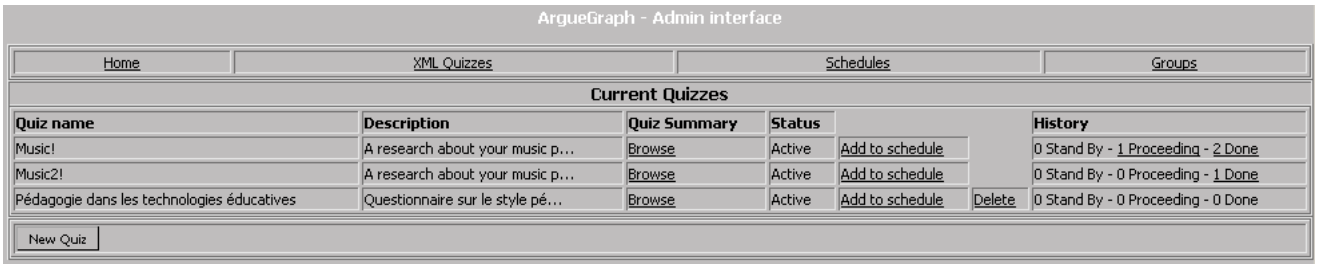


Figure 30 : Interface Administrateur : Menu Principal

Prenons l'exemple du menu principal de l'interface administrateur (figure 30). Nous constatons que cette interface se compose de deux briques distinctes : le menu du haut et le tableau des questionnaires « Current Quizzes ».

Dans la figure suivante, nous explicitons la façon dont nous avons construit cette interface.

```
function ArgueGraph_admin_main() {

  include('tables.php');
  $output = new myclass();
  $AGA= new ArgueGraph();

  // Security check - important to do this as early as possible to avoid
  // potential security holes or just too much wasted processing

  if (!pnSecAuthAction(0, 'ArgueGraph::', '::', ACCESS_ADMIN)) {
    $output->Text(_TEMPLATENOAUTH);
    return $output->GetOutput();
  }
  $modname = "ArgueGraph";
  $output->SetOutputMode(_PNH_KEEPOUTPUT);
  $output->SetInputMode(_PNH_VERBATIMINPUT);
  $output->Text($AGA->Links());
  $output->Text($AGA->MainTable());
  $output->SetInputMode(_PNH_PARSEINPUT);
  $output->SetOutputMode(_PNH_KEEPOUTPUT);


  return $output->GetOutput();
}
```

Figure 31 : Code relatif au menu principal de l'interface administrateur

L'affichage des deux briques du menu principal de l'interface administrateur tient dans les quatre lignes entourées par le cercle du bas. Une instance « AGA » de la classe « ArgueGraph » est créée. Sa méthode « Links() » construit le petit menu de haut tandis que la méthode « MainTable() » construit le tableau récapitulatif des questionnaires. Tout ceci est affiché par la méthode « Text() » de l'objet « output ».

Nous présentons ci dessous, la structure de la méthode « Links() » de la classe « ArgueGraph » qui se trouve dans le fichier « Class_ArgueGraph.php ».

```
  function Links(){
    include('tables.php');

    $output= new pnHTML();
    $result1=$dbconn->Execute("SELECT quizid,qname,description,active FROM $quizzestable");

    $output->Title(_ADMININTERFACE);
    $output->TableStart('','',1);
    $output->SetOutputMode(_PNH_RETURNOUTPUT);
    $row1=array();
    $row1[]=$output->URL(pnModURL('ArgueGraph','admin','main'),_HOME);
    $row1[]=$output->URL(pnModURL('ArgueGraph','admin','XML_quizzes'),_XMLQUIZZES);
    $row1[]=$output->URL(pnModURL('ArgueGraph','admin','schedule',array('status'=>4)),_SCHEDULES);
    $row1[]=$output->URL(pnModURL('ArgueGraph','admin','groups'),_GROUPS);
    $output->SetOutputMode(_PNH_KEEPOUTPUT);
    $output->SetInputMode(_PNH_VERBATIMINPUT);
    $output->TableAddRow($row1,'center');
    $output->TableEnd();

    return $output->GetOutput();
}
```

Figure 32 : Structure de la méthode Links()

3.5.5) Navigation à travers les interfaces du module

Un point intéressant est à souligner dans le code que nous venons de présenter à la figure 32. Ce point concerne la navigation dans un module PostNuke. Pour éditer des liens, il faut faire appel à la méthode « URL » de la classe pnHTML qui a deux attributs: l'URL destination et le texte du lien. La partie encerclée de la figure 32 met en valeur la fonction de base de la navigation dans un programme PostNuke. En effet, pour se déplacer d'une fonction à une

autre dans le fichier « pnadmin.php », il faut utiliser « pnModURL() » de l'API PostNuke. Pour mieux expliciter le fonctionnement de cette fonction, prenons l'exemple de la première ligne de la zone encerclée. Le lien se présente comme suit :

URL(pnModURL('ArgueGraph','admin','main'),_HOME)

Pour trouver son chemin, le système parcourt trois des quatre paramètres d'entrée de la fonction :

- « ArgueGraph » : représente le répertoire du module de même nom;
- « admin » : qui représente l'interface de la fonction destination. Dans notre exemple, le système va accéder au fichier « pnadmin.php ». Par contre, si l'on avait « user » comme paramètre le système accéderait au fichier « pnuser.php » ;
- « main » représente la fonction destination qui est « ArgueGraph_admin_main() ».

Le quatrième paramètre qui n'est pas explicité sur cet exemple et qui est optionnel, est en fait un tableau qui peu contenir des variables nécessaires pour les paramétrage d'une fonction dans une situation donnée.

3.5.6) Gestion de la sécurité du module

Pour accéder à la fonction précédente ou toute autre fonction du fichier « pnadmin.php », il faut que l'utilisateur ait les privilèges d'administration. La partie du milieu encerclée à la figure 31 (page précédente) présente le mécanisme qui permet de restreindre l'accès à ces fonctions aux seuls administrateurs du site. La fonction « pnSecAuthAction() » permet d'atteindre ce but en accédant aux profils utilisateurs stockés dans la base de données du portail d'où elle détermine le type d'accès de l'utilisateur du module. Si celui-ci a un accès du type «ACCESS_ADMIN », il pourra alors administrer le module.

Pour ce qui est des fonctionnalités de l'interface utilisateur gérées par le fichier « pnuser.php », l'accès est restreint aux personnes ayant une session ouverte sur le portail. La fonction « pnSecAuthAction() » est alors appelée par chaque fonction de ce fichier. Elle vérifie que le membre possède bien l'accès « ACCESS_COMMENT » réservé aux personnes connectées au portail et l'autorise donc à naviguer à travers l'interface utilisateur.

3.5.7) Génération des graphes

PHP peut servir à créer et à manipuler des images, dans un grand choix de formats, comme gif, png, jpg, wbmp et xpm. PHP génère directement des images pour le navigateur, avec la librarie GD que nous utiliserons pour créer les graphes du mode solo et du mode duo au format jpg.

Pour créer un graphe, la fonction « ImageCreate() » est appelée. Cette fonction crée une image sans pour autant la sauvegarder sur le disque. On y définit la taille de l'image qui est rectangulaire. Les dimensions de l'image dépendent des poids des axes du questionnaire qu'elle représente : si le poids de l'axe des X est à 2 et si celui des Y est à 1, par exemple, la largeur de l'image sera 2 fois plus importante que sa longueur. La construction propre du graphe commence ici. Il faut savoir que les positions des éléments qui vont être placés sur le graphe se basent sur un jeu de coordonnées 2D dont l'origine se trouve au coin gauche supérieur de l'image.

Pour les graphes du module « ArgueGraph », nous commençons par dessiner les éléments fixes qui sont la grille, le repère du graphe (à travers les axes et les noms des différentes théories qui leurs correspondent), le nom du questionnaire, le nom du modérateur et la légende. La longueur d'un axe est calculée en multipliant le nombre de questions du questionnaire fois le poids de l'axe. Les dimensions des axes sont donc proportionnelles à celles de l'image créée. Ensuite, nous enchaînons par la disposition des couples points-noms des différents participants à la séance Argue Graph selon les scores qu'ils ont obtenu. Si nous sommes en mode solo, nous plaçons simplement un point pour chaque participant et nous écrivons son nom par en dessous. Si par contre nous sommes en mode duo, nous ajoutons au graphe précédent la position de chaque paire d'après son score lors de cette phase. Nous relions ensuite le point représentant la position du participant en mode solo au point représentant la position de la paire dont il a fait partie en mode duo pour mieux souligner l'évolution d'opinion de chacun. La figure suivante illustre un exemple d'un graphe du mode duo.

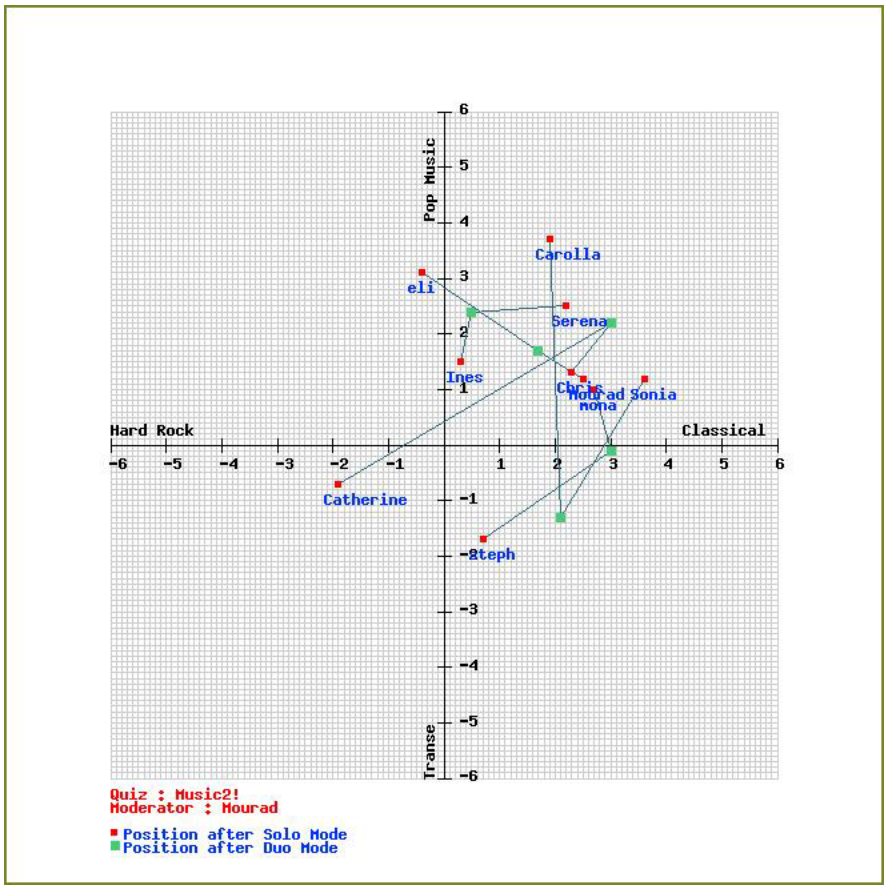


Figure 33 : Exemple d'un graphe du mode duo

Parfois, il arrive que des noms se superposent ou se touchent sur le graphe ce qui rend un peu plus complexe l'analyse des résultats. Pour y remédier, nous avons prévu une petite fonction qui permet de séparer au maximum les noms qui se touchent en les disposant les uns sous les autres à une distance suffisante pour qu'ils soient distincts tout en ne les éloignant pas beaucoup des points qui les représentent.

Une fois l'image terminée, nous pouvons la sauvegarder dans le répertoire prévu pour les graphes. La sauvegarde sous format jpg est possible en utilisant la fonction ImageJPEG() de la librairie GD.

3.5.8) Algorithme de répartition des groupes en paires

Dans les versions précédentes de l' « ArgueGraph », des algorithmes complexes ont été utilisés pour répartir les participants en paires à la suite de la phase solo. Ces algorithmes étaient de gros consommateurs de ressources et de temps. De plus, ils avaient tendance à favoriser les couples de participants ayant une grande divergence d'opinions ce qui faisait que l'on se trouvait parfois avec des couples de personnes trop proches d'opinions. Pour remédier à ces différents problèmes, nous avons décidé de concevoir un nouvel algorithme plus simple et moins coûteux en temps, ayant pour objectif principal une répartition qui élimine en premier temps les formations de couples trop proches d'opinions et qui favorise en second temps les grandes divergences d'opinion. Le principe de cet algorithme est le suivant :

Etape	Description	Prochaine étape
0	On dresse une liste des participants.	1
1	Si la liste n'est pas vide, on calcule pour chaque participant présent dans la liste la somme des distances qui le séparent des autres participants.	2
	Si la liste est vide	4
2	Le participant ayant la somme la plus faible est celui qui est le plus susceptible de se retrouver en paire avec une personne très proche d'opinion de lui. On lui cherche donc le participant qui lui est le plus distant. La paire est ainsi formée.	3
3	On supprime la paire de la liste des participants.	1
4	La liste des paires est prête.	

Cet algorithme donne en général de bons résultats. Cependant, il arrive parfois que l'on puisse configurer une meilleure répartition du groupe en paires manuellement ou que le professeur veuille absolument mettre ensemble la personne A avec la personne B pour X raisons pédagogiques que l'algorithme ne peut prendre en considération. Le module a donc prévu une interface qui permette au modérateur de répartir manuellement son groupe.

3.5.9) Décodage des fichiers XML

La création de fichiers XML est facile, cependant les décoder (parsing) est toute une autre affaire surtout si l'on ne connaît pas la grammaire du document. Pour le module Argue Graph, un fichier-questionnaire XML a une grammaire fixe.

PHP supporte la librairie « expat » qui permet d'analyser mais pas de valider les documents XML. Trois types de codage différents sont supportés, US-ASCII, ISO-8859-1 et UTF-8. Cette extension permet de créer des analyseurs XML puis de définir des points d'entrée pour chaque événement XML. Pour décoder les fichiers XML et transférer leurs contenus dans la base de données du module, nous avons eu recours au gestionnaire d'évènements XML proposé par PHP. Prenons l'exemple du fichier XML suivant :

```
<?xml version='1.0' encoding='iso-8859-1'?>
<ARGUEGRAPH>
<nuke_ArgueGraph_quizzes>
<QNAME>Music!</QNAME>
<DESCRIPTION>A research about your music preferences...</DESCRIPTION>
</nuke_ArgueGraph_quizzes>
<nuke_ArgueGraph_questions>
<QUESTID>1</QUESTID>
<QUESTION>Which group do you prefer?</QUESTION>
</nuke_ArgueGraph_questions>
</ARGUEGRAPH>
```

Nous pouvons analyser ce fichier et en extraire de l'information en utilisant la méthode « parse » de la classe « XML » extraite du fichier « Class_XML.php » :

```
function parse($filepath) {
  global $obj;
  $output= new pnHTML();

  $error=0;

  $xml_parser = xml_parser_create();
  xml_set_element_handler($xml_parser, "startElement", "endElement");
  xml_set_character_data_handler($xml_parser, "characterData");
  if (!($fp = fopen($filepath, "r"))) {
    $error=1;
  }

  while ($data = fread($fp, 4096)) {
    if (!xml_parse($xml_parser, $data, feof($fp))) {
      $error=1;
      }
  }
  xml_parser_free($xml_parser);
  return $error;
}
```

Figure 34 : Méthode parse() de la classe XML

Nous expliquons dans ce qui suit les différentes étapes pour arriver à extraire de l'information du fichier XML précédent.

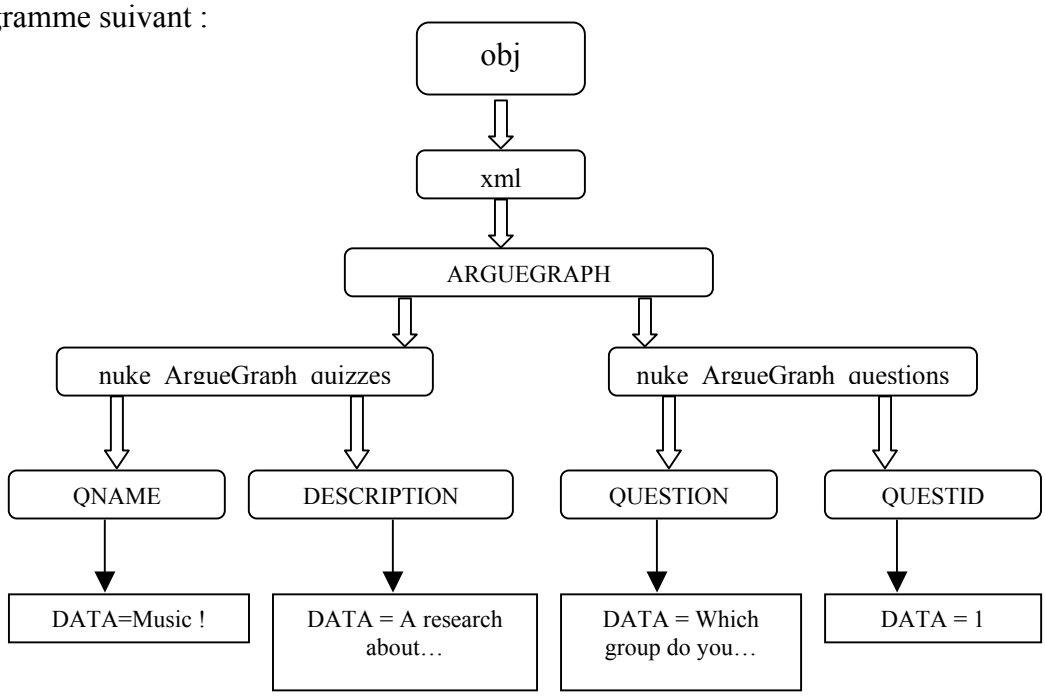
Tout d'abord, la fonction « xml_parser_create() » permet de créer la variable « xml_parser » qui est l'analyseur du fichier XML. La fonction « xml_set_element_handler() » génère un événement à chaque fois que l'analyseur XML rencontre une balise de début ou de fin. Deux gestionnaires sont disponibles : « startElement » pour le début, et « endElement » pour la fin. La fonction « xml_set_character_data_handler ()» collecte l'information qui se trouve entre les balises. Ce processus est répété en boucle jusqu'à la fin du fichier pendant qu'au même moment la variable « xml_parser » se charge en information. Lorsque le processus d'analyse est achevé, le contenu de l'analyseur est passé à la variable globale « obj ». La figure suivante explicite comment extraire l'information à partir de cette variable.

```
function Analyze($filepath,$qname){

    global $obj;
    $obj->tree = '$obj->xml';
    $obj->xml = '';
    $this->parse($filepath);
    $obj1=$obj->xml;
    $obj2=$obj1->ARGUEGRAPH->NUKE_ARGUEGRAPH_QUIZZES;
    $qname=$obj2->QNAME->data;
    $description=$obj2->DESCRIPTION->data;

    $obj3=$obj1->ARGUEGRAPH->NUKE_ARGUEGRAPH_QUESTIONS;
    $questid=$obj3->QUESTID->data;
    $question=$obj3[$i]->QUESTION->data;


}
```

Figure 35 : Extraction de l'information

La variable « obj » est un objet standard prédéfini par PHP qui n'appartient pas un classe précise et qui ne peut avoir que des attributs. Dans le cas de l'analyse du fichier XML, cet objet est un arbre dont les nœuds sont aussi des objets de même nature et dont la profondeur dépend de la profondeur des balises du document XML traité. Dans notre cas, l'objet « obj » ne possède qu'un seul attribut qui est l'objet « xml » correspondant à la première balise du document XML. L'objet « xml » possède une seule feuille qui est l'objet « ARGUEGRAPH »

correspondant à la balise de 2éme niveau <ARGUEGRAPH>. L'objet « ARGUEGRAPH » possède cette fois deux feuilles qui correspondent aux deux balises imbriquées dans la balise <ARGUEGRAPH> (3éme niveau).
La première feuille est l'objet « NUKE_ARGUEGRAPH_QUIZZES » qui possède deux feuilles « QNAME » et « DESCRIPTION ». A ce niveau, les attributs de ces objets sont désignés par « DATA » et ont des valeurs qui correspondent respectivement au nom du questionnaire et à sa description. La structure de l'arbre « obj » est explicitée par le diagramme suivant :



Pour accéder à la valeur de la balise « QUESTID » il faut accéder à la valeur l'attribut « DATA » du nœud « QUESTID » de cette façon :

$description = $obj→xml→ARGUEGRAPH→nuke_ArgueGraph_questions→QUESTID→DATA

3.5.10) Module « WebChat »

Pour la phase duo de l'Argue Graph, il nous a fallu prévoir un système qui permette aux membres de chaque paire de communiquer à distance. Nous aurions pu concevoir un outil de chat intégré mais nous avons préféré utiliser un module « tout fait » qui est le module « WebChat ». Ce module Open Source est disponible au téléchargement pour la communauté PostNuke et est déjà installé sur le portail de TECFA Seed. Ce module permet de créer des salons de discussions publiques et privés. En ce qui concerne le module « ArgueGraph » les discussions vont se passer dans des salons privés. L'utilisation de ce module de chat commence lorsque la liste des paires est validée par le modérateur avant le début de la phase duo. A ce moment précis, un salon de discussion privé est créé pour chaque paire du groupe. Le fichier « pnuser.php » rajoute une ligne dans la table « chatroom » contenant le numéro du salon à créer pour le participant et son partenaire. Ce numéro est calculé en fonction du numéro d'ordre de la séance et du rang de la paire dans la liste des paires.

Les deux modules sont indépendants. L'« ArgueGraph » peut donc tourner normalement sans « WebChat ».

Chapitre 5 : Déploiement et évaluation

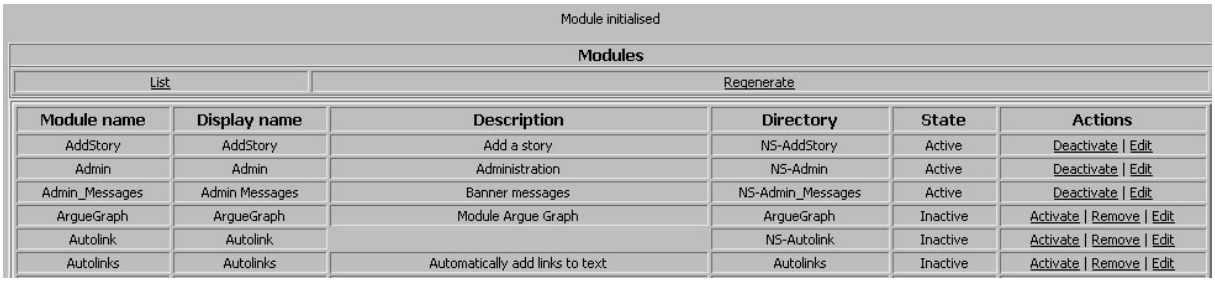
1) Introduction

Dans ce chapitre nous décrivons le processus de déploiement de l'« ArgueGraph ». Nous présentons tout d'abord les différentes phases du processus d'installation du module. Nous enchaînons ensuite par des tests d'utilisation en illustrant les différentes fonctionnalités du système et nous concluons par une évaluation du produit fini.

2) Installation

Le module « ArgueGraph » se présente sous la forme d'un package. Les deux formats les plus utilisés pour l'empaquetage dans la communauté PostNuke sont Winzip (extension « .zip ») et TAR compressé (extension « .tar.gz »).

Pour installer le module dans PostNuke, il faut tout d'abord décompresser le contenu du package dans le répertoire « modules » du portail. L'utilisateur doit ensuite ouvrir une session administrateur sur le portail. Dans le menu « Administration », le lien « Modules » pointe vers une interface qui liste tous les modules (activés ou pas) présents sur le portail. Si l'« ArgueGraph » n'y apparaît pas, le lien « Regenerate » permet au système de scanner à nouveau le contenu du répertoire « modules » et de détecter la présence du nouveau programme. Une fois détecté, le module est rajouté à la liste des modules du portail. Dans la case correspondant à l' « ArgueGraph » de la colonne « Actions », trois liens existent. Le lien « Edit » permet de saisir le nom du programme et sa description telles qu'elles apparaîtront pour l'utilisateur. Le lien « Remove » permet de supprimer le module et son répertoire du portail. Pour continuer avec le processus d'installation du programme, le lien « Initialise » permet d'initialiser le module en créant ses tables dans la base de données du système (voir Description du fichier « pninit.php » page 48). La figure suivante est extraite de l'interface d'installation de gestion tout de suite après l'initialisation de l'« ArgueGraph ».


Module initialised

Modules

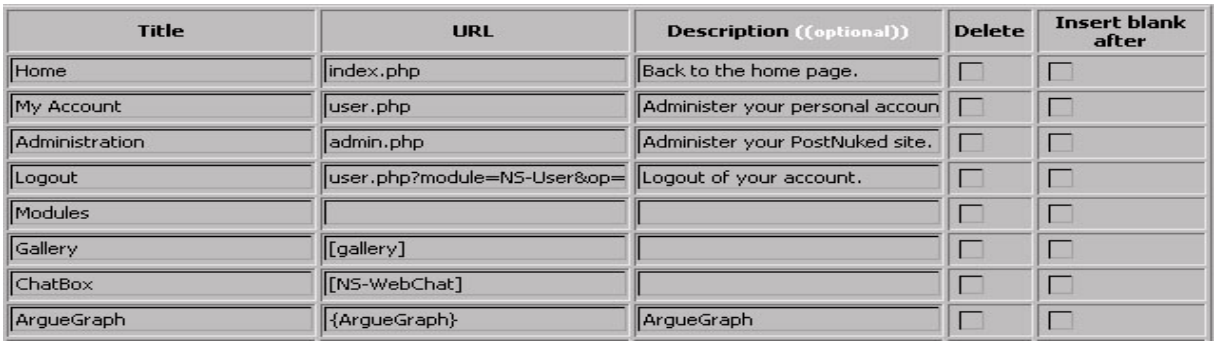
List | Regenerate

Module name	Display name	Description	Directory	State	Actions
AddStory	AddStory	Add a story	NS-AddStory	Active	Deactivate | Edit
Admin	Admin	Administration	NS-Admin	Active	Deactivate | Edit
Admin_Messages	Admin Messages	Banner messages	NS-Admin_Messages	Active	Deactivate | Edit
ArgueGraph	ArgueGraph	Module Argue Graph	ArgueGraph	Inactive	Activate | Remove | Edit
Autolink	Autolink		NS-Autolink	Inactive	Activate | Remove | Edit
Autolinks	Autolinks	Automatically add links to text	Autolinks	Inactive	Activate | Remove | Edit

Figure 36 : Installation du module "ArgueGrpah"

Une fois le module initialisé, il ne reste plus qu'à l'activer en cliquant sur le lien « Activate ». L'« ArgueGraph » apparaît maintenant dans le menu « Administration » du portail. L'interface administrateur du module est maintenant accessible. Ce n'est par contre pas le cas de l'interface utilisateur. Pour permettre aux apprenants d'utiliser le programme, il faut qu'il soit listé parmi les liens d'un des blocs-menu latéraux du portail. Nous pouvons par exemple l'insérer dans le bloc « Main Menu ». Dans ce cas, il faut retourner à l'interface d'administration du portail et accéder à la rubrique « Blocks ». Une liste des blocs du portail apparaît alors. Un lien vers « Main Menu » permet d'accéder à la liste de tous les liens que comporte ce bloc. Cette liste se présente sous la forme d'un tableau. Chaque ligne du tableau comporte 5 champs : le titre du module, son URL, sa description (optionnelle), une case à cocher qui permet de supprimer le lien et une seconde case à cocher permettant d'espacer d'une ligne deux liens du bloc « Main Menu ».

La figure suivante illustre l'allure du tableau de gestion du bloc « Main Menu ».



Title	URL	Description ((optional))	Delete	Insert blank after
Home	index.php	Back to the home page.	☐	☐
My Account	user.php	Administer your personal accoun	☐	☐
Administration	admin.php	Administer your PostNuked site.	☐	☐
Logout	user.php?module=NS-User&op=	Logout of your account.	☐	☐
Modules			☐	☐
Gallery	[gallery]		☐	☐
ChatBox	[NS-WebChat]		☐	☐
ArgueGraph	{ArgueGraph}	ArgueGraph	☐	☐

Figure 37 : Ajout d'un lien "ArgueGraph" dans le bloc "Main Menu"

Pour pouvoir accéder à l'interface utilisateur, une URL de forme spéciale doit être introduite. En effet, PostNuke a prévu une notation standard pour les modules de nouvelles génération : ce sont les modules dont les fonctionnalités sont accessibles via les deux fichiers « pnadmin.php » et « pnuser.php ». Les fonctionnalités des modules de l'ancienne génération (par exemple le module « Gallery ») sont elles accessibles à travers le seul fichier « index.php ». L'URL de l'interface utilisateur du module « ArgueGraph » doit donc être de la forme {ArgueGraph} ; les accolades renvoyant le système automatiquement vers la fonction « main » du fichier « pnuser.php ». Une fois les autres champs remplis et le formulaire validé, le lien apparaît dans le bloc « Main Menu ».

Nous avons installé le module « ArgueGraph » avec succès sur plusieurs portails dont le portail TECFA Seed (http://tecfaseed.unige.ch/door).

3) Déploiement et tests

La phase de tests du module a commencé bien avant la fin de la phase de réalisation. En effet, le développement du module s'est fait en plusieurs étapes. Nous avons tout d'abord développé les fonctionnalités de gestion des questionnaires puis celles de la programmation des groupes et des séances. Nous avons ensuite développé les fonctionnalités de l'interface utilisateur et nous avons fini par mettre en place le service d'échange de fichiers XML. A la suite de chacune de ces phases nous avons entrepris des séances de tests pour vérifier, corriger et valider le travail qui venait d'être fait. Une fois le module achevé nous l'avons soumis à des tests à plus grandes échelles en faisant appel à des collaborateurs TECFA.

3.1) Administration du module

Nous avons testé à deux reprises les fonctionnalités administratives du site. Nous avons demandé à deux collaboratrices TECFA de créer des questionnaires et de programmer des séances Argue Graph pour un groupe donné d'utilisateurs. Le premier test a été entrepris par une collaboratrice scientifique (mathématicienne) qui à déjà développé des modules PostNuke pour TECFA tel que le projet ePBL (Projet Based Learning). Le deuxième test a été entrepris par une pédagogue (professeur d'anglais) qui n'a que quelques notions en programmation.

Nous avons soumis aux deux collaboratrices un questionnaire qui a déjà été utilisé lors des versions précédentes de l' « ArgueGraph ». Il s'agit d'un questionnaire ayant pour thème la pédagogie dans les technologies éducatives. Ce questionnaire comporte dix questions. Nous leur avons ensuite demandé de choisir un groupe de dix personnes et de les programmer pour

une séance Argue Graph. Ces deux collaboratrices devaient faire partie du groupe de participants tout en étant elles-mêmes les modératrices des séances.

Les tests se sont plutôt bien déroulés. Il y a eu tout de même quelques hésitations de la part de la deuxième collaboratrice lorsqu'il s'agissait de naviguer à travers les fonctionnalités du système. Ces hésitations sont naturelles pour toute personne nouvelle au module. Un fois l'expérience renouvelée, ces hésitations finissent par disparaître. Il est donc recommandable d'établir un manuel d'utilisation qui décrit le fonctionnement du système surtout pour les personnes les plus réticentes à l'intégration de l'informatique dans l'éducation.

La première collaboratrice a bien construit son questionnaire et a facilement programmé sa séance Argue Graph. Le sommaire du questionnaire achevé se présente comme suit :

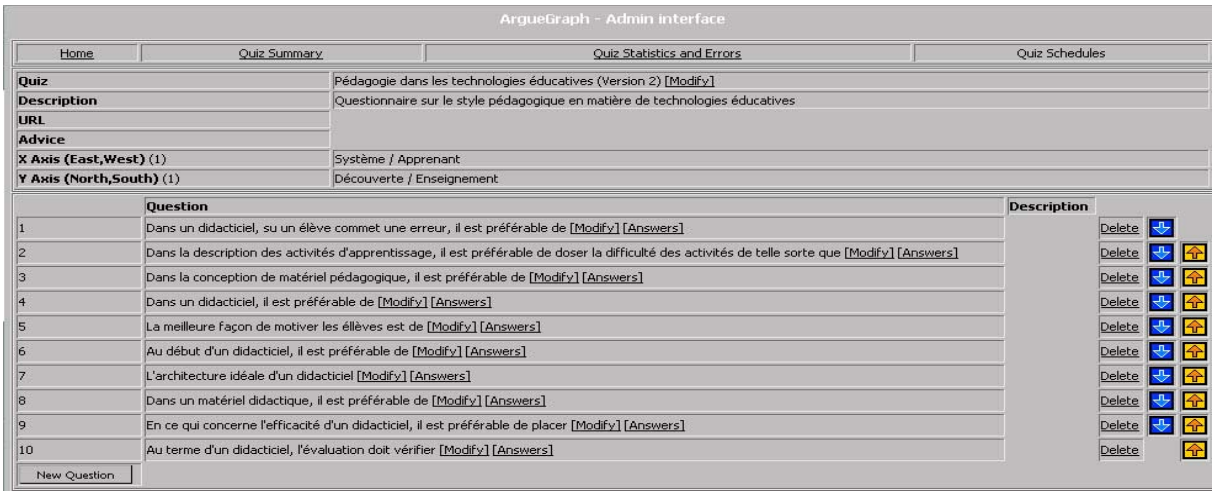


Figure 38 : Sommaire du questionnaire test

3.2) Exécution du scénario Argue Graph

Le déploiement du scénario sur un groupe de personnes est la partie la plus importante du module. Nous avons réuni un groupe de huit collaborateurs TECFA et nous avons désigné une pédagogue (la même que pour les tests d'administration) pour modérer l'expérience, celle-ci faisant aussi partie du groupe de participants. Les participants au scénario étaient disposés dans des locaux différents et n'avaient pas, pour la majorité, la possibilité de communiquer directement. La séance a naturellement commencé par la phase solo. Les utilisateurs ont apprécié la présentation du formulaire qui offre, entre autres, une barre de progression qui permet d'informer chaque participant de son état d'avancement lors de cette première étape.

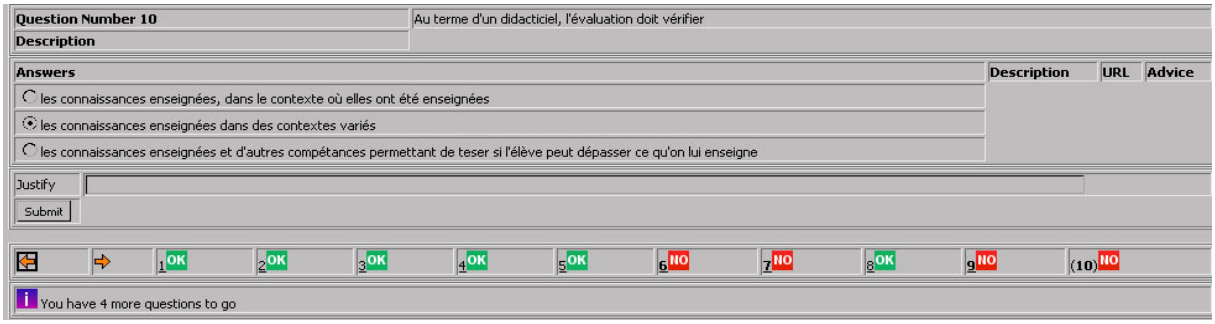
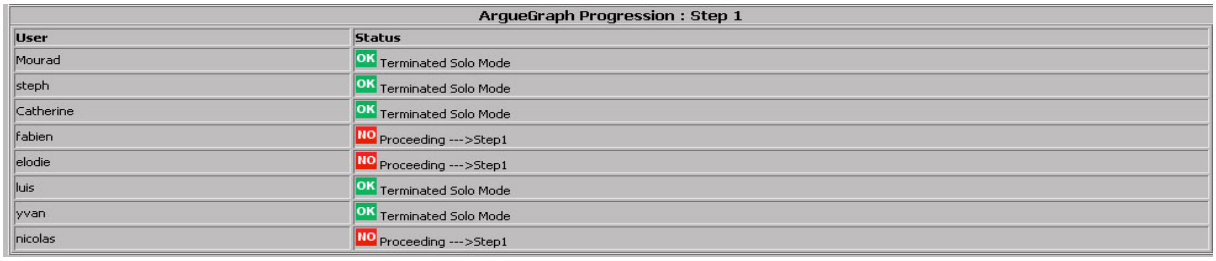


Figure 39 : Répondre en mode solo

Du côté administrateur, la modératrice, qui a aussi répondu au questionnaire, a eu la possibilité de s'informer de l'état d'avancement de la phase solo à travers un tableau qui déterminait pour chaque participant si, oui ou non, il a terminé de répondre aux questions.



ArgueGraph Progression : Step 1

User	Status
Mourad	OK Terminated Solo Mode
steph	OK Terminated Solo Mode
Catherine	OK Terminated Solo Mode
fabien	NO Proceeding --->Step1
elodie	NO Proceeding --->Step1
luis	OK Terminated Solo Mode
yvan	OK Terminated Solo Mode
nicolas	NO Proceeding --->Step1

Figure 40 : Progression de la phase solo

Une fois que tout le monde a répondu à toutes les questions, les résultats se sont automatiquement affichés chez la modératrice sous la forme d'un graphe et d'une liste de paire de participants. L'algorithme que nous avons expliqué à la page 53 a donné une assez bonne répartition du groupe. La modératrice pouvait néanmoins répartir le groupe manuellement et en changer la configuration. Estimant la répartition générée par l'algorithme convenable, la modératrice a validé et publié les résultats qui se sont affichés chez les participants. Pour se préparer à la phase duo, un salon de chat réservé à chaque paire leur a permis de décider celui qui allait répondre pour le compte de la paire pour la suite du scénario.

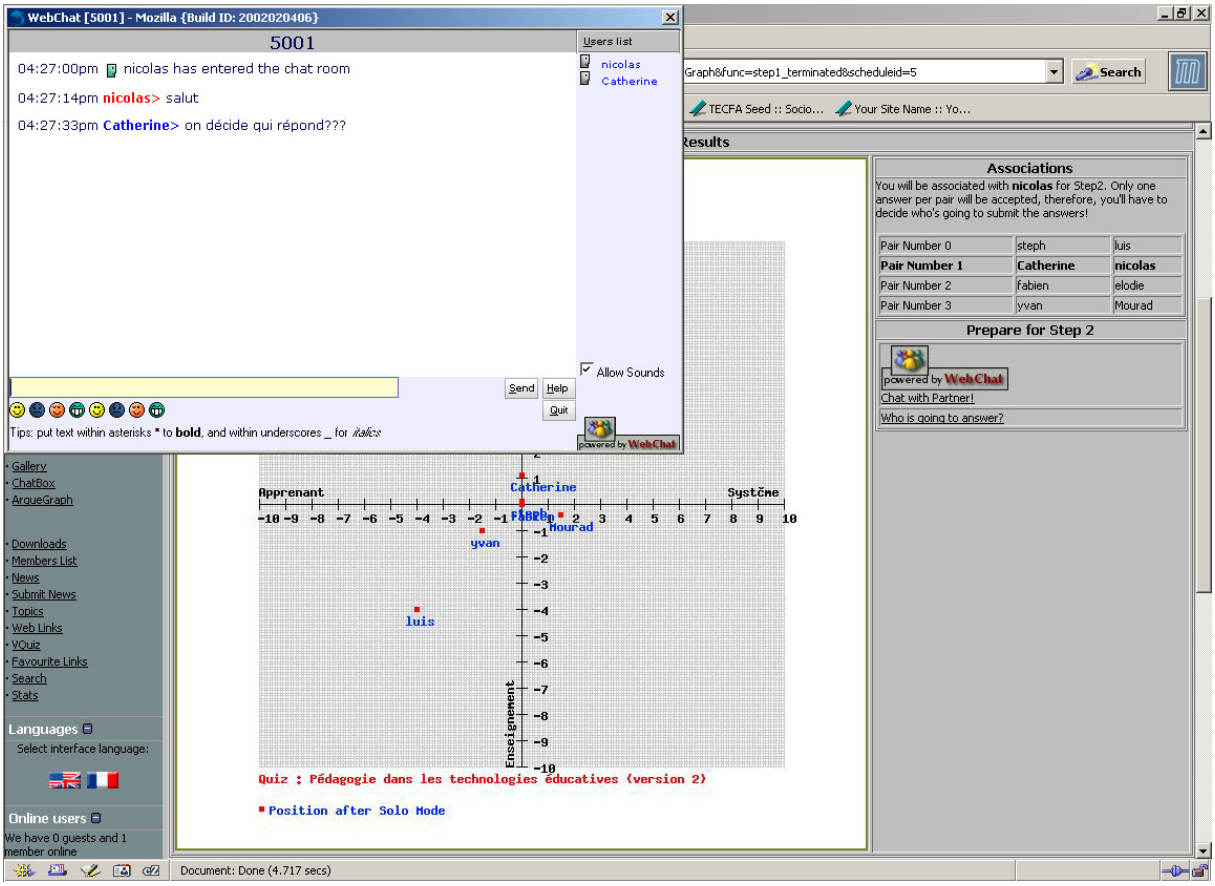


Figure 41 : Résultats de la phase solo et préparation de la phase duo

En décidant l'envoi de la phase duo, les utilisateurs ont pu commencé à répondre au questionnaire de nouveau. Le chat les a motivé pour communiquer et s'échanger des idées.

Une fois la phase achevée et les résultats publiés, les participants ont donc pu constater l'évolution de leur opinions à travers le graphe duo et se situer dans les statistiques qui leurs sont fournies avec.

Le déploiement du script côté participant s'est assez bien déroulé, le mode de fonctionnement étant facile à saisir. Par contre, du côté de la modératrice, une assistance de ma part a été nécessaire au moment de la transition entre les deux phases.

4) Evaluation du travail

Les tests que nous avons entrepris nous ont permis d'évaluer le degré de correspondance du module aux attentes de départ. Nous énumérons dans cette partie les différents accomplissements que nous avons atteints et les améliorations que nous pourrions envisager pour une version future du programme.

4.1) Accomplissements

L'interface utilisateur correspond généralement aux attentes de départ. Son utilisation par les participants est aisée et ne nécessite aucune mise en condition préalable. De plus, des fonctionnalités ont été rajoutés pour accroître la convivialité du programme. Les barres de progression et la présentation des graphes ont nettement amélioré l'aspect visuel des interfaces.

Un point important concernant les problèmes de navigation dans les applications web a été soulevé par une collaboratrice TECFA. Nous savons que les navigateurs Internet offrent des fonctionnalités de retour en arrière qui permettent au navigateur de visualiser des pages web déjà visitées. L'utilisation de ces pages, surtout dans le cas de formulaires, peut produire des incohérences tant sur le plan du déroulement normal de l'application que sur les résultats qu'elle fournit. Pour y remédier, nous avons prévu un contrôleur d'état d'avancement des séances Argue Graph qui se base sur les codes que nous avons attribués à chaque phase du script (voir Tableau 1: Les différents états d'une séance Argue Graph page 27). A chaque fois qu'un utilisateur déclenche un événement dans le système (poster des formulaires, rafraîchir une page expirée, etc.), le module vérifie la code de la phase en cours. Si l'événement fait partie de la phase, le système enclenche la fonction cible. Dans le cas contraire, le système est redirigé vers une fonction que nous avons nommé « Switchboard » (Tableau de bord) qui va déterminer quelle interface doit être affichée.

L'interface administrateur a aussi bénéficié de l'ajout de fonctionnalités qui rendent la gestion du module plus aisée. Pour la création des graphes, une nouvelle fonctionnalité a attiré l'attention des testeurs et des observateurs parmi lesquels Mr Patrick Jermann, l'un des deux concepteurs du scénario Argue Graph. Il s'agit du gestionnaire des fichiers-questionnaires XML.

D'autres fonctionnalités, comme celle de pouvoir créer et sauvegarder des groupes d'utilisateurs, celle de pouvoir suivre l'état d'avancement des phases ou bien celle de pouvoir avoir une fonction d'agrandissement des graphes ont été bien accueillies puisqu'elles facilitaient à des degrés différents l'administration et la modération du système.

Enfin, d'un point vue technique, la nouvelle version de l' « ArgueGraph » est plus malléable et documentée que les deux versions précédentes.

Il sera donc facile de comprendre le code source du programme et de le changer si besoin en est, dans le cadre de la réutilisation.

4.2) Améliorations et perspectives

Comme nous l'avons cité lors de la phase des tests, un manuel d'utilisation demeure un élément crucial pour faciliter la compréhension des fonctionnalités de gestion et de modération du système. Vu le nombre important d'options et de services qu'offre l'interface administration, un manuel permettrait aux utilisateurs du module de ne pas passer à côté de fonctionnalités qui existent et qui seraient susceptibles de simplifier leurs tâches.

Sur le plan de l'interactivité, nous pourrions envisager un outil qui permette au modérateur de communiquer avec les participants à travers l'envoi de messages groupés ou individuels (une variante de la fonction « net send » par exemple). Cela pourrait être utile pour, par exemple, aviser les utilisateurs des limitations en temps de chaque phase ou pour les informer de la publication des résultats.

D'un point de vue technique, des problèmes de navigateurs ont été mis en évidence lors de la phase de tests. L'aspect des interfaces et leur comportement changent selon le navigateur. A TECFA nous avons surtout utilisé les navigateur Internet Explorer et Mozilla. Lorsqu'un lien s'ouvre sur une nouvelle page avec Mozilla, la page source ne change pas alors que pour Internet Explorer, le navigateur est automatiquement et étrangement redirigé vers la homepage du portail. Ce problème mineur est lié à la fonction « myURL() » que nous avons ajouté aux fonctions de la classe « pnHTML ». Cette fonction, qui permet d'ouvrir un lien sur une nouvelle fenêtre, contient du JavaScript. Il faudra sans doute collaborer avec la communauté PostNuke pour trouver une solution meilleure à ce problème.

Pour conclure, nous pourrions apporter des changements au module « WebChat » pour limiter son utilisation au seul module « ArgueGraph ».

Discussion est conclusion

Les portails communautaires fleurissent de partout sur la toile dont un grand nombre offre des contenus intéressants qui accrochent les internautes en les intégrant dans des communautés de personnes d'intérêt de plus en plus larges.

Nous ne doutons pas qu'en combinant scénarios d'apprentissage socioconstructivistes avec plates-formes de collaboration telles que les portails communautaires, une nouvelle dimension pourrait être apportée à l'éducation se basant sur les nouvelles technologies de l'information. Actuellement, TECFA engage ses étudiants dans des scénarios d'apprentissage par projet basés sur des modules PostNuke. L'expérience est concluante et les résultats sont satisfaisants tant du point de vue des apprenants que celui des enseignants. Néanmoins, la formule ne prend toujours pas dans le reste du monde de l'éducation, exceptions faites des pays scandinaves qui ont une marge d'avance dans ce domaine et où les enseignements se font désormais avec un support Internet (en ligne ou en classe) et en collaborant. Nous pouvons expliquer ceci par le fait que l'on n'a pas assez de recul par rapport à ces technologies ce qui ne contribue pas à éliminer la réticence des enseignants à adopter ces nouvelles méthodologies d'apprentissage. La complexité des systèmes d'information peut aussi accentuer ce sentiment chez les enseignants. En effet, nous avons remarqué que des portails Java se basant sur les « portlets » ont du mal à conquérir ce marché. Ceci est tout à fait normal vu que ce sont des systèmes gourmands en ressources et particulièrement compliqués à utiliser pour des non informaticiens (donc les enseignants et les pédagogues) .

Du côté des apprenants le processus d'adoption est par contre plus aisé. En effet, les enfants semblent avoir moins de problèmes que les adultes à s'adapter aux nouvelles technologies éducatives, car ils s'adaptent très facilement aux graphismes et aux abstractions conceptuelles et ont souvent une bien meilleure expérience dans la découverte et l'exercice des fonctionnalités des interfaces (Rousseau, 1999), ce qui laisse présager un avenir possible pour ce type d'outils dans l'éducation.

Il nous paraît donc primordial de trouver un compris entre scénarios pédagogiques et complexité des nouvelles technologies de l'information. Nous avons pu démontrer à travers ce projet que l'on pouvait à partir d'outils légers et simples (PostNuke, PhP, MySQL et XML) concevoir des systèmes éducatifs pouvant s'intégrer parfaitement dans l'esprit de l'apprentissage socioconstructiviste tout en restant faciles à manipuler pour les novices et les plus réticents des enseignants. Il s'agit maintenant de promouvoir ce genre d'outils et de les diffuser…

Annexes

Annexe 1 : Le module « Favourite Links »

En guise de familiarisation avec les portails PostNuke, j'ai été chargé de développer un module qui permet à l'administrateur de choisir jusqu'à dix liens de la liste des URL du module « Web Links » et de les afficher dans un bloc indépendant sur un portail PostNuke.

Le principe de ce module-bloc que nous avons appelé « FavLinks » est simple. Il s'agit de développer une interface administrateur qui regroupe par catégories et sous-catégories de liens (la forme d'un arbre) tous les URL d'un portail PostNuke. Il a donc fallu créer des fonctions qui permettent de construire un arbre à partir de la hiérarchisation de ces liens et de l'afficher sur une interface qui se présente sous la forme d'un formulaire. L'administrateur n'aura qu'à cocher les cases correspondant à ses liens favoris. Une seule table est prévu pour ce module dans la base de données. Cette table va contenir le code des liens sélectionnés par l'administrateur. Les autres caractéristiques des URL sont puisés des tables du module « Web Links ». Une fois les liens sélectionnés, ils apparaissent dans un bloc que l'on pourra placer parmi les autres blocs d'un portail PostNuke.

La programmation du bloc est plus simple que la programmation de l'interface administrateur. En effet, dans le cas de « Favourite Links » , une seule fonction est appelée pour l'affichage qui est block_FavLinks_block(). Cependant avant d'entamer le développement des fonctionnalités de cette fonction, une en-tête nécessaire à sa reconnaissance par le système doit être rajoutée. Le morceau de code suivant explicite cette en-tête qui permet de au portail d'accéder à la fonction principale du bloc.

```
$blocks_modules['FavLinks'] = array(
   'func_display' => 'blocks_FavLinks_block',
   'text_type' => 'FavLinks',
   'text_type_long' => "Favourite Links",
   'allow_multiple' => false,
   'form_content' => false,
   'form_refresh' => false,
   'show_preview' => false
);
```

La fonctionnalité du bloc doit être sous la forme d'un fichier qui a pour nom le titre du module. Dans notre cas, c'est le fichier « FavLinks.php ». Ce fichier doit être placé dans un sous-répertoire « pnblocks » du répertoire « FavLinks ».

Nous avons empaqueté ce module avec un mode d'emploi sous la forme d'un fichier compressé ZIP et nous l'avons soumis à la communauté PostNuke à travers le site www.postnuke.com. Un comité de validation a vérifié le bon fonctionnement du module et l'a publié sur le site afin qu'il puisse être téléchargé. Le lien de téléchargement pointe vers la rubrique « Downloads » du portail TECFA Seed. A cette date 193 téléchargements ont été enregistrés et plusieurs sites utilisent ce module. Plusieurs utilisateurs ont donné des feed-back et suggéré des améliorations renforçant cet aspect de collaboration. La figure 6 à la page 17 illustre le bloc « FavLinks » tel qu'on peut le voir sur le portail TECFA Seed (en haut à droite).
La prochaine étape devra donc être la validation du module « ArgueGraph » par la communauté PostNuke.

Annexe 2 : Le fichier « myClass.php »

```
<?php

class myclass extends pnHTML{
 function myURL($url, $text)
    {
       if (empty($url))
            return;
       $output = "<a href=\"".$_SERVER['PHPSELF']."\" onclick=\"javascript:window.open('$url','_blank')\">";

       if (!empty($text)) {
          if ($this->GetInputMode() == _PNH_PARSEINPUT)
            $text = pnVarPrepForDisplay($text) ;
          $output .= $text;
       }
       $output .= '</a>' ;
       if ($this->GetOutputMode() == _PNH_RETURNOUTPUT)
            return $output;
       else
          $this->output .= $output;
}

  function myIMG($url, $text) {
    if (empty ($url)) return;
    $output = "<img src='$url' alt='$text' >";
    if ($this->GetOutputMode() == _PNH_RETURNOUTPUT) {
      return $output;
    } else {
      $this->output .= $output;
    }
  }

   function myImgURL($url,$text,$alt) {
    if (empty ($url))
            return;
 $output = "<a title='$alt' href='$url'><img SRC='$text' ALT='$alt'>";
    if (!empty($text)) {
      if ($this->GetInputMode() == _PNH_PARSEINPUT) {
            $text = pnVarPrepForDisplay($text);
      }
    }
    $output .= '</a>';

    if ($this->GetOutputMode() == _PNH_RETURNOUTPUT) {
      return $output;
    } else {
      $this->output .= $output;
    }
  }
}

?>
```

La section suivante illustre le contenu du fichier « myClass.php » qui contient des fonctions que nous avons estimés nécessaires et qui manquent à l'API « pnHTML » (voir page 44). Ces fonctions sont des méthode d'une nouvelle classe « myClass » qui étend la classe « pnHTML ».

- La fonction « myURL » permet d'ouvrir un lien dans une nouvelle page.
- La fonction « myImg » permet d'afficher des images.
- La fonction « myImgURL » est une combinaison des deux fonctions précédentes. Elle permet d'afficher des liens image.

Ces fonctions ont été utilisées dans le module "ArgueGraph", dans le module-bloc "FavLinks" (voir Annexe 1 page 64) et le module "ePBL" développé par une collaboratrice TECFA pour l'apprentissage par projet.

La fonction "myURL" présente tout de même quelques problèmes lorsqu'elle la navigateur utilisé est Internet Explorer de Microsoft. Nous avons constaté que lorsque l'URL est lancée, la page d'origine est systématiquement redirigée vers la homepage du portail. Il serait donc intéressant de collaborer avec la communauté PostNuke pour trouver une solution à ce problème et intégrer ces fonctionnalités dans l'API « pnHTML ».

Annexe 3 : Analyseur XML

Nous présentons dans cet annexe le code source complet de l'analyseur de fichiers XML que nous pensons intéressant. Cela complètera la description de l'analyse de fichiers XML que nous avons abordé au chapitre 4 de ce document (page 52).

```
 function parse($filepath) {
  global $obj;
  $output= new pnHTML() ;
  $error=0;

  $xml_parser = xml_parser_create();
  xml_set_element_handler($xml_parser, "startElement", "endElement");
  xml_set_character_data_handler($xml_parser, "characterData");
  if (!($fp = fopen($filepath, "r"))) {
   $error=1;
  }

  while ($data = fread($fp, 4096)) {
   if (!xml_parse($xml_parser, $data, feof($fp))) {
          $error=1;
          }
  }
  xml_parser_free($xml_parser);
  return $error;
 }

function startElement($parser, $name, $attrs) {
 global $obj;

 // If var already defined, make array
 eval('$test=isset('.$obj->tree.'->'.$name.');');
 if ($test) {
  eval('$tmp='.$obj->tree.'->'.$name.';');
  eval('$arr=is_array('.$obj->tree.'->'.$name.');');
  if (!$arr) {
   eval('unset('.$obj->tree.'->'.$name.');');
   eval($obj->tree.'->'.$name.'[0]=$tmp;');
   $cnt = 1;
  }
  else {
   eval('$cnt=count('.$obj->tree.'->'.$name.');');
  }

  $obj->tree .= '->'.$name."[$cnt]";
 }
 else {
  $obj->tree .= '->'.$name;
 }
 if (count($attrs)) {
  eval($obj->tree.'->attr=$attrs;');

 }
}
```

```
function endElement($parser, $name) {
  global $obj;
  // Strip off last ->
  for($a=strlen($obj->tree);$a>0;$a--) {
    if (substr($obj->tree, $a, 2) == '->') {
      $obj->tree = substr($obj->tree, 0, $a);
      break;
              }
  }
}

function characterData($parser, $data) {
  global $obj;
  $data=addslashes($data);
  eval($obj->tree.'->data=\".$data.'\';');
  $data=stripslashes($data);
}
```

Nous explicitons dans cet annexe les deux fonctions qui permettent de détecter les balises de début et de fin. Ces deux fonctions sont « startElement » et « endElement ».
La fonction « characterData » permet de remplir l'arbre « obj » avec le contenu de l'information détectée entre les balises.

Bibliographie

Références

- Bielaczyc, K. & Collins, A. (1999). Learning communities in classrooms: a reconceptualization of educational practice. In Reigeluth, C. (Ed.), Instructional-Design Theories and Models: Vol. 2. A New Paradigm of Instructional Theory, (pp. 269- 292). Mahwah, NJ: Lawrence Erlbaum Associates.

- Bloom, B.S. (1984), The 2 Sigma Problem: The Search for Methods of Group Instruction as Effective as One-to-One Tutoring. Educational Researcher, (pp.3-16).

- Dillenbourg, P. (2002). Over-scripting CSCL: The risks of blending collaborative learning with instructional design, http://tecfa.unige.ch/perso/staf/notari/literature/dillenbour2002-1.doc.

- Dillenbourg, P., Baker, M., Blaye, A. & O'Malley, C.(1995) The evolution of research on collaborative learning. In E. Spada & P.Reiman (Eds) Learning in Humans and Machine: Towards an interdisciplinary learning science. (pp. 198-211) Oxford : Elsevier.

- Frimout, D. & Wynants, M. (2000), Manifeste pour l'utilisation d'ordinateurs et de l'Internet dans l'enseignement, URL :http://www.manifest.be/executing.cfm?langue=FR.

- Gagné, R.M. (1987)(ed.) Instructional Technology : Foundations, Hillsdale :Erlbaum.

- Rousseau, M. (1999), Immersive Interactive Virtual Reality and Informal Education, Foundation of the Hellenic World, URL:http://www.fhw.gr/.

- Schneider, D. (2001),Le rôle de l'Internet dans la formation supérieure: scénarii et technologies, colloque International « Enseignement des Langues et Multimédia », Alger, avril 2001.

- Schneider, D.K, Synteta,V., Frété C., (2002) Community, Content and Collaboration Management Systems in Education: A new chance for socio-constructivist scenarios?. Proceedings of the 3rd Congress on Information and Communication Technologies in Education, Rhodes, September 26th-29th 2002.

- Synteta, P. (2001), EVA_pm: Design and Development of a Scaffolding Environment For Students Projects, Master Thesis, TECFA, FPSE, Univerité de Genève.

- Wilson, B. & Lowry, M., Constructivist Learning on the Web, in Burge, L. (Ed.), Learning Technologies: Reflective and Strategic Thinking. San Francisco: Jossey-Bass, New Directions for Adult and Continuing Education.

Ouvrages scientifiques

- Cocoon: Building Xml Applications, Langham, M.; Ziegeler, C., Macmillan Computer Pub, 450 pages, Juillet 2002.

- Core Mysql Atkinson, L., Prentice Hall Computer Books, 721 pages, Novembre 2001.

- Php 4 Developer's Guide, Schwendiman, B., Osborne McGraw-Hill, 775 pages, Janvier 2001.

- Uml in a Nutshell: A Desktop Quick Reference, Alhir, S. S., Oreilly & Associates, 273 pages, Octobre 1998.

Sites Web

- www.mysql.com

- www.php.net

- www.postnuke.com

- www.toutjavascript.com