

TECHNOWEB
Moyens mis en œuvre par le groupe
de coordination et d'uniformisation

FALLET Laurent
JOUANNO Guillaume
MALLET Grégory

21 mai 2003



Table des matières

1	Notre rôle dans le projet SILAT	3
1.1	Uniformiser	3
1.2	Coordonner	3
2	Uniformiser : moyens mis en œuvre	4
2.1	Spécifications rédigées en L ^A T _E X	4
2.2	Consignes de codage	4
2.3	Cohérence du site	4
2.4	Rapports de spécifications	4
3	Coordonner : moyens mis en œuvre	5
3.1	Le CVS	5
3.2	Génération automatique de documentation	5
3.3	Suivi des groupes	5
3.4	Devenir l'interlocuteur privilégié	5
4	Synthèse du site	6
4.1	Maquette	6
4.2	Intégration	6
4.3	Erreurs à ne pas commettre	6
4.3.1	Ne pas être un guide	6
4.3.2	Mal définir le rôle de chacun	6
5	Annexes	7
5.1	Exemple de modèle de spécifications	7
5.2	Consignes de codage	10
5.3	Maquette	10
5.4	Comment documenter les fichiers php ?	11
5.5	API générée	11

Table des figures

1	Maquette de la première page	8
2	Maquette de la seconde page	9
3	Sommaire des consignes de codage	10
4	Interface du site de l'Engref	10
5	Aide à la documentation de l'API	11
6	Exemple d'API que l'on peut obtenir avec PHPDoc	11

1 Notre rôle dans le projet SILAT

Début février, les 6 groupes de TT ont choisis un rôle dans le projet SILAT. Celui du TT3 est de veiller à la mise en place d'outils identiques pour chaque groupe, et de rendre possible l'intégration finale des éléments réalisés par chaque groupe.

Voici le découpage :

- TT 1 : Gestion des logs et des comptes
- TT 2 : Fonctionnalités de l'éditeur : création d'exercices
- TT 3 : Uniformisation et coordination
- TT 4 : Fonctionnalités de l'éditeur : traitement et visualisation
- TT 5 : Carte dynamique de position
- TT 6 : Fonctionnalités du lecteur

1.1 Uniformiser

L'uniformisation des documents permet à chaque membre du projet de pouvoir comprendre le travail réalisé par les autres groupes, ou au moins d'en avoir une vue d'ensemble. Les documents produits ont souvent pour destinataire le client (ici le client et le professeur en charge du projet). Ceux-ci, produits par différents groupes, doivent avoir la même structure, le même format, pour permettre une lecture plus facile, donc une meilleure compréhension. Les documents ne doivent pas non plus donner l'impression d'un désordre mais que l'ensemble de l'équipe travaille ensemble sur le même projet.

1.2 Coordonner

Il est nécessaire que chaque groupe avance au fur et à mesure. Sans cela, les groupes ne peuvent traiter des problèmes qui apparaissent au fur et à mesure, et doivent tout faire à la dernière minute. De plus, l'intégration finale de toutes les parties dans le site ne doit pas être réalisée à la fin. Il faut pouvoir assembler le site morceau par morceau ; un groupe nous fournira la première fois une version sommaire de la partie dont il a la charge, avec les fonctionnalités minimales. Ensuite il agrandira ses fonctionnalités dans les prochaines versions. De notre côté, cela permet d'avoir le temps de réaliser l'intégration (qui n'est pas immédiate) et de vérifier au fur et à mesure l'avancement du projet. Nous pouvons ainsi plus facilement, grâce à un point de vue plus global, détecter les erreurs ou incohérences de conception (du point de vue de l'utilisateur). Cela est moins aisé lorsque l'on est en charge de la programmation de la dite partie.

Cette démarche par itération nous permettra de voir les problèmes arriver et de trouver des solutions.

2 Uniformiser : moyens mis en œuvre

2.1 Spécifications rédigées en L^AT_EX

Les spécifications sont réalisées par tous les groupes ; elles doivent donc être au même format. A la première échéance, les formats rendus n'étaient pas tous les mêmes, malgré qu'un modèle en L^AT_EX avait été fourni. Pour la suite du projet, nous veillerons à ce que chaque document ait un format prédéfini (une DTD en quelque sorte).

Pourquoi L^AT_EX ? Ce langage permet d'avoir, peu importe la station (Windows ou Linux), un document formaté conformément à un modèle pré-établi. De plus de nombreux programmeurs recommandent ce langage, dont le format de sortie peut être varié mais avec une présentation toujours soignée.

Voir en annexe un exemple du résultat de ce que l'on obtient grâce au modèle créé.

2.2 Consignes de codage

Le rôle premier de ces consignes est de résoudre les questions récurrentes dans toute programmation telles que les conventions de noms ou bien l'indentation. Ces recommandations rendent le code plus lisible et permettent donc une plus grande compréhension entre les différents groupes. Le deuxième objectif de ces consignes fut de fixer la manière de traiter certains problèmes (inclusion de fichiers, structures de contrôle, ...). Encore une fois, ceci augmente la compréhensibilité du code, puisque le lecteur n'a plus à se demander pourquoi telle solution a été adoptée plutôt qu'une autre. Enfin, ces consignes nous ont permis d'insérer quelques petits conseils, souvent inspirés par le bon sens, mais auxquels on ne pense pas toujours lorsque l'on est plongé dans le codage.

Pour la réalisation elle-même de ces consignes, nous nous sommes inspirés de toutes les sources que nous avons pu trouver sur le net (celles citées à la fin du document ne sont que les plus utilisées), en essayant de faire un ensemble le plus cohérent possible. La structure du document est donc restée très proche du document de base (<http://pear.php.net/manual/fr/standards.php>) même si de nombreuses rubriques et remarques ont été ajoutées.

Il est à noter que, dans le cadre d'un projet professionnel, une section sur l'optimisation du code aurait pu être ajoutée mais notre projet ayant également une dimension pédagogique. La lisibilité du code a donc été préférée aux quelques millisecondes qui auraient pu être ainsi gagnées.

Voir en annexe les sections des consignes de codage.

2.3 Cohérence du site

Notre mission est aussi d'assurer la cohérence de l'ensemble du site : l'IHM étant entièrement prise en charge par notre groupe, nous informons les autres groupes de ce que nous attendons d'eux afin de pouvoir intégrer directement leur travail au reste du site. Cela concerne la mise en page mais aussi et surtout la navigation.

2.4 Rapports de spécifications

Les rapports de spécifications sont joints à la suite de ce document. Ils sont classés par ordre de TT (groupe de travail) comme suit :

- TT 1 : Login et comptes (administrateur, prof, étudiant)
- TT 2 : Création d'un exercice par l'éditeur
- TT 4 : Traitement et visualisation (delta, suivi du parcours de l'étudiant)
- TT 5 : Carte dynamique de position
- TT 6 : Navigation dans l'exercice par le lecteur

3 Coordonner : moyens mis en œuvre

3.1 Le CVS

Le CVS est un outil très pratique pour le versionning tant au niveau des documents de spécifications (partie BD et TW conception) mais aussi pour le codage. En effet, il est nécessaire de pouvoir revenir à une version antérieure d'un fichier lorsque la version actuelle ne fonctionne pas. On dispose de sauvegardes de chaque instant du projet, nous mettant à l'abri des erreurs de programmation ou des incidents matériels. Une copie du projet est également chez chaque programmeur, lui permettant de tester l'ensemble du site.

Le CVS était déjà installé et paramétré par le groupe chargé de la collaboration globale en Base de Données. Nous avons donc mis un répertoire en plus chez chaque groupe pour qu'il y dépose ses fichiers. L'avantage majeur des fichiers non compilés (mode texte, c'est à dire html, php, inc) est que lorsque CVS fait une révision, il conserve la modification du fichier et les commentaires associés. Nous pourrions savoir qui a modifié le fichier et pour quelle raison.

3.2 Génération automatique de documentation

Nous allons coupler CVS avec PHPDoc, un outil de génération de documentation de classes PHP. Une page sur le site web de la machine Silat permettra de déclencher la génération de cette API, qui sera ensuite accessible directement sur le même serveur. La génération de la documentation prend de quelques secondes à 1 ou 2 minutes pour un projet contenant beaucoup de classes avec de nombreuses fonctions.

Les groupes qui codent pourront d'une part coder à plusieurs dans le même groupe, et d'autre part accéder aux fonctions de classes codées par d'autres groupes, ce qui permettra de ne pas réécrire les mêmes fonctions plusieurs fois. Dans le cas où plusieurs accès simultanés à cette page bloquent le serveur ou nuisent à la génération de la doc, il sera mis en place un script qui génèrera cette api toutes les heures.

Voir en annexe (page 11) un exemple de documentation générée.

3.3 Suivi des groupes

Le suivi a été instauré à partir du moment où les groupes se sont mis à coder. Auparavant nous n'étions pas toujours au courant de leur avancement, nous savions seulement si la tâche globale avait été terminée ou non.

Dorénavant nous demandons à chaque groupe de se fixer des objectifs pour la quinzaine à venir. Il doit également signaler si il a rempli ses objectifs pour la quinzaine passé, et se justifier si ce n'est pas le cas. Cela permettra de juger de l'avancement du projet plus précisément que dans le passé.

3.4 Devenir l'interlocuteur privilégié

Cela signifie que nous serons le contact entre les groupes et le professeur en charge de l'encadrement du projet (partie Technoweb, c'est à dire intégration en PHP). Nous nous devons d'être le plus possible au courant des problèmes de chaque groupe (si ils ne sont pas internes). Cependant si les précédents points ont été respectés les problèmes ne devront pas être trop nombreux. Ce sera un indicateur des outils mis en place.

En français courant, les documents seront remis au TT3 qui se chargera de les transmettre au professeur superviseur du projet. La vérification des documents pourra ainsi être faite par le TT précédemment cité.

4 Synthèse du site

4.1 Maquette

La maquette est élaborée avec un souci constant : l'ergonomie. L'utilisateur ne doit jamais être perdu, l'accès aux documents doit pouvoir se faire le plus rapidement possible. Chaque lien doit être précis et permettre à l'utilisateur de savoir exactement l'information qu'il va trouver derrière.

Pour que l'interface soit la plus conviviale et la plus confortable possible, la présentation est aérée, les différentes zones du site sont clairement délimitées par des formes aux contours fermés et des couleurs distinctes pour chaque élément.

4.2 Intégration

L'intégration est le groupement des 5 parties des groupes ayant collaboré au projet. Si la répartition du travail de chaque groupe est bien faite, la quantité de pages non réalisées devrait être presque nulle. Dans le cas contraire, il nous incombera de réaliser ces pages, soit en les déléguant aux groupes, soit en les faisant nous même.

C'est pourquoi il est important de commencer cette intégration au plus tôt. Cette phase recoupe celle des tests.

4.3 Erreurs à ne pas commettre

Ci-dessous la leçon des erreurs que nous avons commises.

4.3.1 Ne pas être un guide

Dès le début nous devons être plus "directeur". Il faut être plus ferme sur la nécessité d'uniformisation, ce qui est toujours un peu contraignant pour les groupes qui ne rédigent qu'un seul document. Il faut montrer et expliquer pourquoi ce choix est le meilleur, et pour que cette décision soit mieux acceptée fournir le modèle au minimum une semaine avant la deadline.

4.3.2 Mal définir le rôle de chacun

Les rôles ont été mal découpés au niveau de la charge de travail. L'appréciation de la charge de travail au début d'un projet est chose difficile. De plus les compétences de chacun sont les mêmes ; il n'est pas possible d'attribuer une partie à un groupe plus qu'un autre, cela s'est établi selon les goûts de chacun. Par exemple, la création des interfaces de l'éditeur et de la carte de position sont 2 parties qui demandent du travail. La première tâches est scindable en 2 alors que la seconde l'est difficilement. Voilà pourquoi avoir l'expérience et l'habitude de programmer permet de mieux évaluer la quantité de travail à fournir et le temps à y consacrer.

Les zones d'ombre sont aussi un problème qui survient en cours de codage. Nous appelons zone d'ombre les pages que nous avons oublié, ou qui ont été mal définies par le client. Par exemple la page nommée "restitution"¹ est très obscure, peu détaillée. Une page aurait du être prévue pour que les élèves puissent donner leur avis sur l'exercice. Nous avons maintenant figé le schéma de la base de donnée, donc nous ne pourrons plus stocker ces informations à l'intérieur. Si nous réalisons la page, les informations seraient stockées anarchiquement, ce qui est contraire aux principes de programmation que l'on nous enseigne.

¹section 4.1.2, page 16 du rapport de Cédric MAHE

5 Annexes

5.1 Exemple de modèle de spécifications

FALLET Laurent JOUANNO Guillaume MALLET Grégory

TT X - TechnoWeb - Fonction du groupe

21 mai 2003

Rapport fonctionnalités - Version : 1.0

Partie : étudiant, prof, login...

Ici vous pouvez insérer un commentaire de ce que vous avez fait. N'hésitez pas à décrire votre rôle, si cela aide à la compréhension de votre travail.

Table des figures



FIG. 1 – Maquette de la première page

Fonctionnalités associées à la page :

- obtenirBidule
E :
S : (login)[..]
Récupère la liste de tous les étudiants

- obtenirTruc
E : login
S : (numero_exercice, version)[..]
Récupère la liste de tous les exercices



FIG. 2 – Maquette de la seconde page

Fonctionnalités associées à la page :

- obtenirBidule
E :
S : (login)[..]
Récupère le bidule

- obtenirTruc
E : login
S : (numero_exercice, version)[..]
Récupère la liste de tous les trucs

5.2 Consignes de codage

Ces dernières sont disponibles sur la page web <http://asi.insa-rouen.fr/~lfallet/technoweb/consignesdecodage.html>. Egalement disponible sur CVS.

Consignes de Codage V0.4

Sommaire

[Consignes générales](#)

[Packages](#)

[Normalisation](#)

[Tags PHP](#)

[Structures de Contrôles](#)

[Les "Magic Numbers"](#)

[Appels de Fonction](#)

[Définitions des Fonctions](#)

[Commentaires](#)

[Inclure du Code](#)

[Indentation](#)

[Commentaires d'En-tête](#)

[Exemple d'URLs](#)

[Conventions de Nom](#)

[Derniers conseils importants](#)

[Sources](#)

FIG. 3 – Sommaire des consignes de codage

5.3 Maquette

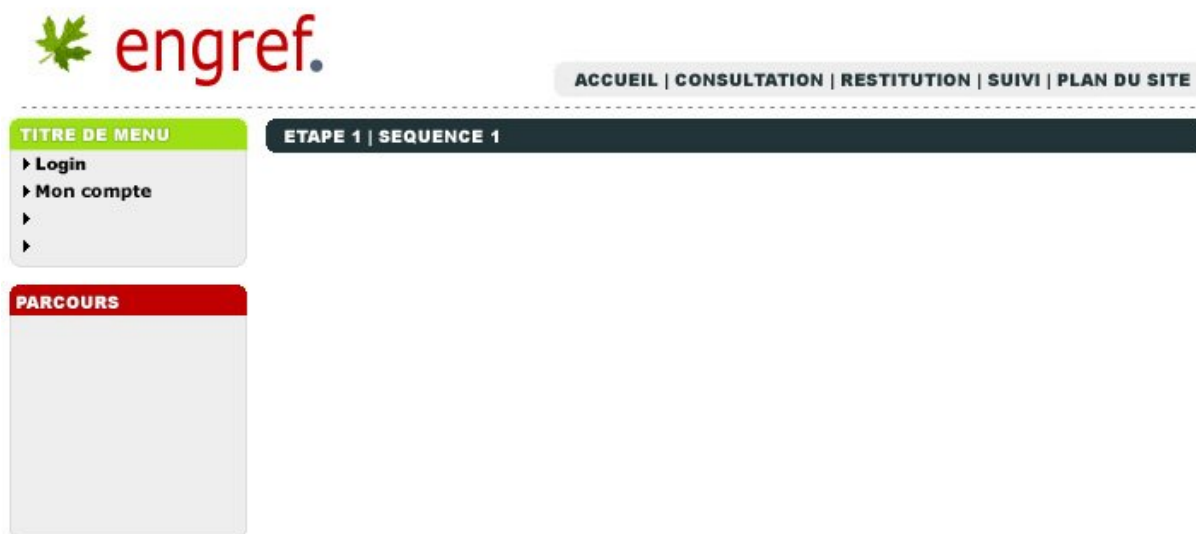


FIG. 4 – Interface du site de l'Engref

5.4 Comment documenter les fichiers php ?

La documentation pour commenter les fichiers contenant les classes est disponible sur cette page <http://asi.insa-rouen.fr/~lfallet/technoweb/commentairesphp.html>. Egalement disponible sur CVS.

HOW TO commenter ses fichiers PHP... avec PHPDoc!

[Documenter ses fichiers](#) -- La structure des commentaires dans un document

[@author](#) -- décrit l'auteur

[@const](#) -- décrit les constantes créées avec la commande `define()`

[@deprecated](#) -- signale les éléments obsolètes de l'API

[@global](#) -- documente les variables globales utilisées

[@package](#) -- regroupe les classes en packages

[@param](#) -- décrit les paramètres d'une fonction

[@see](#) -- marque les références à d'autres éléments ou méthodes

[@since](#) -- indique la date à laquelle l'élément a été introduit

[@static](#) -- indique que la fonction est static

[@var](#) -- décrit les variables de la classe

Sont indiqués en gras les éléments dont vous devez absolument vous servir...
La syntaxe est propre à chaque marqueur. Cela n'est pas à connaître par coeur
mais vous devez vous y référer à chaque fois que vous codez.

FIG. 5 – Aide à la documentation de l'API

5.5 API générée

The screenshot shows a PHPDoc-generated API page. On the left is a sidebar with a navigation menu. The main content area displays the class `SecondClasse` which inherits from `PrimClasse`. It includes a code snippet for the class definition and a description of the class. The sidebar menu includes links for 'Packageindex', 'Classtrees', 'Modulegroups', 'Elementlist', 'Report', 'XML', and 'Files'. The main content area also has a top navigation bar with the same links. The class `SecondClasse` is described as a derived class of `PrimClasse`. The code snippet shows `public class SecondClasse extends PrimClasse`. The authors are listed as Laurent FALLET with an email address. The version is 0.1. At the bottom, there is a section for 'Methods inherited from PrimClasse' with a link to 'primclasse'.

FIG. 6 – Exemple d'API que l'on peut obtenir avec PHPDoc