



ECOLE DES MINES D'ALBI
C A R M A U X

Projet 5 : Application d'emplois du temps pour l'EMAC

Rapport de projet

Maître d'ouvrage :
Franck VIGNOT

Maîtrise d'œuvre :
Thomas GROSSIN
Xavier MOGHRABI

IFI 2003
Option GSI

-Janvier 2003-



ECOLE DES MINES D'ALBI
C A R M A U X

Projet 5 : Application d'emplois du temps pour l'EMAC

Rapport de projet

Maître d'ouvrage :
Franck VIGNOT

Maîtrise d'œuvre :
Thomas GROSSIN
Xavier MOGHRABI

IFI 2003
Option GSI

-Janvier 2003-

Sommaire

<i>Sommaire</i>	4
<i>Sommaire</i>	4
<i>Introduction</i>	7
<i>Introduction</i>	7
<i>Gestion de projet d'emploi du temps</i>	8
<i>Gestion de projet d'emploi du temps</i>	8
<i>1 Application de consultation Intranet</i>	8
1.1 Définition des fonctions	8
1.2 Modification de la page de saisie	8
1.2.1 Reprogrammation de la page.....	8
1.2.2 Utilisation des scripts Javascript.....	8
1.2.3 Intégration LDAP.....	8
1.3 Modification de l'affichage	8
1.3.1 Prise en main du module SVG.....	8
1.3.2 Création d'un affichage en SVG.....	8
1.4 Réunion et validation	9
1.4.1 Intégration des sous-programmes.....	9
1.4.2 Test et correction des bogues.....	9
1.4.3 Validation du résultat.....	9
1.4.4 Prise en compte des critiques.....	9
1.4.5 Test et correction des bogues.....	9
<i>2 Sélection de progiciels d'emploi du temps</i>	9
2.1 Recherche de produits	9
2.1.1 Recherche sur la base CXP.....	9
2.1.2 Recherche sur Internet.....	9
2.1.3 Demande de documentation.....	9
2.2 Analyse des produits	10
2.2.1 Analyse des documentations.....	10
2.2.2 Réalisation d'une check liste.....	10
2.2.3 Sélection des produits adéquats.....	10
2.2.4 Renseignement de la check liste.....	10
<i>3 Administratif</i>	10
3.1 Rédaction des spécifications	10
3.2 Rédaction compte rendu	10
3.3 Soutenance	10
<i>4 Gestion des risques</i>	10
<i>Recherche de progiciel de création et de gestion d'emploi du temps</i>	12
<i>Recherche de progiciel de création et de gestion d'emploi du temps</i>	12
<i>1 Création de la check liste</i>	12
<i>5 Réponse à la check liste</i>	12

6	<i>Check liste pour le choix d'un progiciel d'emploi du temps</i>	13
6.1	Caractéristiques commerciales	13
6.1.1	Identifiant logiciel.....	13
6.1.2	Commentaires, demandes du cahier des charges.....	13
6.1.3	Généralité.....	13
6.1.3.1	Société.....	13
6.1.3.2	Commercialisation.....	14
6.1.4	Prestations.....	14
6.1.4.1	Documentation.....	14
6.1.4.2	Contrat de droit d'usage.....	15
6.1.4.3	Formation.....	15
6.1.4.4	Garantie.....	15
6.1.4.5	Maintenance du logiciel.....	16
6.1.5	Coûts du logiciel et des prestations (Prix HT).....	17
6.2	Caractéristiques techniques	17
6.2.1	Développement du produit.....	17
6.2.2	Spécification de fonctionnement.....	18
6.2.3	Maintenance technique.....	19
6.2.4	Administration et supervision applicative.....	19
6.2.5	Intégration et communication applicative.....	20
6.2.6	Gestion de l'aide en ligne.....	21
6.3	Caractéristiques fonctionnelles	22
6.3.1	Caractéristiques générales du système d'emploi du temps.....	22
6.3.1.1	Réalisations des emplois du temps.....	22
6.3.1.2	Statistiques d'utilisation des ressources.....	23
6.3.1.3	Consultation des emplois du temps.....	24
6.3.1.4	Gestion des données d'entrée/sortie.....	24
6.3.2	Les données de bases.....	24
6.3.2.1	Les ressources (les champs non renseignés sont pris en compte par le progiciel).....	24
6.3.3	Caractéristiques non fonctionnelles.....	26
6.3.3.1	Interface Homme Machine.....	26
7	<i>Conseil sur les logiciels</i>	27
	<i>Développement de l'application de consultation des emplois du temps</i>	28
	<i>Développement de l'application de consultation des emplois du temps</i>	28
2	<i>Spécifications de la page de saisie des critères</i>	28
7.1	Spécifications fonctionnelles.....	28
7.2	Spécifications logiques.....	29
8	<i>La réalisation</i>	30
8.1	Fonctionnement général du programme.....	30
8.2	Description de la page de saisie des critères.....	30
8.2.1	Le choix de la période et de la promotion se fait par boutons radios.....	30
8.2.2	La sélection des critères de la matière, de la salle et de l'enseignant.....	31
8.2.2.1	L'utilisation de scripts Javascript.....	32
8.2.2.2	L'utilisation de l'annuaire centralisé de l'école.....	33
8.2.2.3	La vérification sur la saisie du formulaire.....	34
9	<i>Spécifications de la page de consultation</i>	34
9.1	Spécifications fonctionnelles.....	35
9.2	Spécifications logiques.....	36
10	<i>La réalisation</i>	37
10.1	Le sous programme AfficherResultat.....	37

10.2 Le sous-programme Graph.....	38
10.3 Le sous programme Detail.....	40
<i>Conclusion.....</i>	<i>41</i>
<i>Conclusion.....</i>	<i>41</i>
<i>3 Conclusion sur la recherche de logiciel :.....</i>	<i>41</i>
<i>11 Conclusion sur l'application de consultation.....</i>	<i>41</i>
<i>12 Conclusion en terme d'apport personnel.....</i>	<i>42</i>
<i>Bibliographie.....</i>	<i>43</i>
<i>Bibliographie.....</i>	<i>43</i>
<i>4 SVG.....</i>	<i>43</i>
<i>13 Javascript.....</i>	<i>43</i>
<i>14 Perl.....</i>	<i>43</i>
<i>15 HTML.....</i>	<i>43</i>
<i>16 CXP.....</i>	<i>43</i>
<i>17 Information sur les progiciels.....</i>	<i>43</i>
<i>Annexes.....</i>	<i>44</i>
<i>Annexes.....</i>	<i>44</i>

INTRODUCTION

Le projet d'application d'emplois du temps a pour but :

- de développer une nouvelle application de consultation des emplois du temps sur Intranet,
- de conseiller le responsable du service scolarité dans l'acquisition d'un progiciel de création et de gestion des emplois du temps.

Ainsi la réalisation consiste à mener deux sous projets différents.

La maîtrise d'œuvre est déléguée à deux élèves en option de Gestion des Systèmes d'Information. La maîtrise d'ouvrage est assurée par M. Vignot, responsable du service scolarité, et M. Otton, responsable du service informatique. Ils apportent ainsi leurs compétences techniques pour le développement. Ils servent pour le deuxième sujet de conseillers pour rechercher une solution adaptée aux besoins spécifiques des concepteurs des emplois du temps. De plus, ils assurent la gestion et le suivi du projet pour son bon déroulement.

L'application de consultation répondra à un souci d'esthétique. Le résultat de présentation devra s'approcher des plannings imprimés à partir du logiciel d'emploi du temps GET utilisé pour l'instant à l'école.

Ce dernier logiciel devra être remplacé à terme par un nouveau système. La recherche devra donc montrer l'ensemble des progiciels existants satisfaisants les critères définis dans le cahier des charges.

Nous commencerons donc le récit du projet par la gestion de projet adoptée. Nous relaterons ainsi l'ensemble des tâches réalisées, la répartition du travail et les risques du projet. Nous enchaînerons par les détails du développement de la consultation en spécifiant les choix adoptés. Nous finirons sur la recherche des progiciels d'emploi du temps en expliquant notre démarche, les résultats obtenus et les conseils de choix.

Gestion de projet d'emploi du temps

La gestion est illustrée par un diagramme de Gant donné en annexe.

1 Application de consultation Intranet

1.1 Définition des fonctions

C'est la première tâche de cette partie du projet. Nous l'avons réalisée en collaboration étroite avec les maîtres d'ouvrages. Cette partie concerne simplement la validation du cahier de charges.

A l'issu des définitions, nous avons identifié deux parties distinctes dans l'élaboration de l'application de consultation. Xavier s'est occupé de la partie modification de la page de saisie et Thomas de la modification de l'affichage. Ces tâches ont été exécutées en parallèles.

1.2 Modification de la page de saisie

1.2.1 Reprogrammation de la page

La page de saisie des critères a été redéveloppée en ajoutant de nouvelles fonctionnalités et la navigabilité a été améliorée. Elle a permis de redéfinir les paramètres qui sont envoyés du formulaire à la page de traitement des données. Un travail a été fait pour prendre les données de la base et les retraiter afin d'obtenir des listes de choix.

1.2.2 Utilisation des scripts Javascript

Cette partie a duré 2 jours puisqu'il a fallu apprendre comment s'utilise ce langage de script. Les fonctions ont permis d'ajouter du dynamisme sur la page de saisie. Une première version de la page a été développée en utilisant des fenêtres ouvrantes mais elle a été abandonnée car l'ouverture prenait environ 2 secondes en raison de la consultation de la base de données. Ainsi une deuxième version a été développée sans fenêtre ouvrante.

1.2.3 Intégration LDAP

L'annuaire centralisé a permis de sélectionner des choix par défaut (nom de l'enseignement ou promotion de l'élève) suivant l'utilisateur. Cette partie a nécessité l'intégration d'une fonction pour récupérer les données et l'ajout dans le programme de code pour proposer ces choix par défaut.

1.3 Modification de l'affichage

1.3.1 Prise en main du module SVG

Les capacités de SVG nous étaient totalement inconnues. Il a donc fallu comprendre les bases de son fonctionnement. Ceci a été réalisé en lisant la documentation sur Internet et en suivant un tutorial. Nous avons réalisé des exemples simples pour la pratique. Les sites utilisés sont reportés en bibliographie.

1.3.2 Création d'un affichage en SVG

Une fois SVG suffisamment maîtrisé, nous avons programmé l'affichage en tant que tel. La génération du SVG est réalisée dynamiquement en Perl. Il a donc fallu mélanger ces deux technologies.

1.4 Réunion et validation

1.4.1 Intégration des sous-programmes

A partir de cette tâche nous reprenons le travail en commun.

Ce travail a pour but d'accorder les deux parties de nos sources. Cela a été grandement facilité par le dialogue constant qu'il y avait entre nous deux et l'anticipation de nos besoins réciproques. De plus, certains problèmes de programmation ont été résolus par une entraide mutuelle.

1.4.2 Test et correction des bogues

Nous avons effectué une série de test pour valider la programmation, nous avons essayé chacun des cas de sélection. Chaque bogue repéré a été immédiatement corrigé.

1.4.3 Validation du résultat

Les résultats obtenus ont été présentés aux Maîtres d'ouvrages.

1.4.4 Prise en compte des critiques

Les limites dénoncées de notre programme ont été corrigées.

1.4.5 Test et correction des bogues

Nous avons de nouveau réalisé une série de test.

Les modifications ont été montrées aux Maîtres d'ouvrages. Les prises en compte des remarques ont été jugées satisfaisantes.

D'autres pistes d'améliorations, non définies dans le cahier des charges, ont été avancées lors de cette démonstration. Leur importance relative étant faible. Le temps imparti à la programmation terminé, nous avons décidé de n'en pas tenir compte afin de ne pas porter préjudice à l'autre partie du projet.

2 Sélection de progiciels d'emploi du temps

C'est la deuxième partie du projet. En terme chronologique, nous avons commencé par la recherche de produits pour pouvoir rapidement demander de la documentation. Nous avions prévu de réaliser la programmation en attendant l'arrivée de celle-ci.

2.1 Recherche de produits

2.1.1 Recherche sur la base CXP

L'utilisation de la base CXP nous avait été conseillée.

2.1.2 Recherche sur Internet

La base CXP s'étant révélée décevante pour ce type de logiciel. Nous avons cherché d'autres moyens de trouver des produits. Internet paraissait le moyen le plus efficace et le plus rapide.

2.1.3 Demande de documentation

Sur chacun des logiciels trouvés, nous avons demandé une documentation par téléphone ou par mail.

2.2 Analyse des produits

2.2.1 Analyse des documentations

Nous espérons que la documentation serait suffisamment détaillée pour réaliser une bonne évaluation des progiciels du marché. Cela n'a pas été le cas. Nous avons donc décidé de réaliser une grille de questions (une check liste) que nous pourrions remplir par entretien téléphonique.

2.2.2 Réalisation d'une check liste

Nous nous sommes aidés d'une check liste réalisée par le CXP sur les ERP. La check liste reprend les éléments du cahier des charges et des éléments d'information et de qualité sur les logiciels.

2.2.3 Sélection des produits adéquats

Nous avons restreint notre recherche aux seuls progiciels qui répondaient à la contrainte majeure de notre cahier des charges : le caractère multiposte.

2.2.4 Renseignement de la check liste

Nous nous sommes partagés les produits sélectionnés et nous avons réalisé de longues interviews téléphoniques pour renseigner cette check liste. Certaines personnes se sont montrées largement moins coopératives (celles d'Index Education).

3 Administratif

3.1 Rédaction des spécifications

Nous avons rédigé les spécifications des programmes pour commencer la rédaction du rapport. Elles sont la mise au propre des définitions réalisées en première tâche et des remarques suggérées par les Maîtres d'ouvrages.

3.2 Rédaction compte rendu

Pour faire un retour d'expérience, nous avons rédigé un compte rendu faisant part de notre organisation de projet, des spécifications détaillées et de nos conclusions.

3.3 Soutenance

La soutenance permettra de présenter l'ensemble de nos travaux aux maîtres d'ouvrage.

4 Gestion des risques

Risques	Mesures pour la prévention du risque
Projet inachevé dans le temps	<ul style="list-style-type: none">- Simplifier le développement en fixant des objectifs concrets- Se faire aider par les experts- Faire du travail en dehors des heures projet- Bien se répartir les tâches
Le programme ne répond que partiellement aux attentes	<ul style="list-style-type: none">- Définition des objectifs avec le maître d'ouvrage- Bien définir le cahier des charges en le faisant valider et s'y référer- Faire des tests afin de corriger les éventuels bogues- Valider les réalisations auprès du maître d'ouvrage

L'équipe a des difficultés techniques et bloque sur le développement	<ul style="list-style-type: none">- Utiliser des ouvrages techniques d'informatique- S'aider des experts ou de personnes compétentes en informatique- Rechercher d'autres moyens de programmation
Le site Internet n'est pas utilisé par les émaciens	<ul style="list-style-type: none">- Créer un site de qualité (convivial, précis et esthétique)
Manque d'informations sur les progiciels de gestion d'emplois du temps	<ul style="list-style-type: none">- Recherche des progiciels et demande de documentations effectués en priorité- Multiplication des sources d'information- Importante prise de contact téléphonique

Recherche de progiciel de création et de gestion d'emploi du temps

1 Création de la check liste

Nous avons étudié le Pack Expert du CXP concernant l'analyse des ERP. La check liste est un outil d'expression de besoins. Elle consiste à élaborer une liste de critères pertinents en vue de choisir un progiciel.

Certains critères assez généraux sont ainsi issus de la check liste proposée par le CXP dans l'ouvrage précité. Cependant nous l'avons adaptée à notre contexte de choix d'un progiciel de création et de gestion d'emploi du temps. Ainsi pour déterminer les questions propres à nos besoins, nous avons repris les caractéristiques du progiciel idéal décrites dans le cahier des charges.

Les critères de la liste se découpent en trois domaines :

- les **critères commerciaux** recensent les points clefs à observer en ce qui concerne la diffusion du progiciel, les prix, les prestations offertes par le fournisseur, le contrat, le nombre de références, la formation des utilisateurs et la documentation fournie
- les **critères techniques** concernent particulièrement l'environnement matériel du progiciel, à savoir le type de machine, le système d'exploitation nécessaire, le mode de stockage des données supporté et l'architecture
- les **critères fonctionnels** consistent la partie essentielle de la check liste. Elle correspond aux grandes fonctions que nous avons définies dans le cahier des charges.

Cette check liste est donc adaptée aux besoins exprimés par les utilisateurs. Nous avons décidé de ne pas réaliser une pondération pour attribuer une note globale aux progiciels mais de qualifier les critères en déterminant ceux essentiels.

5 Réponse à la check liste

Nous avons recherché l'ensemble des progiciels d'emploi du temps. Puis nous avons sélectionné ceux qui répondaient à un critère de déterminant : le progiciel doit obligatoirement être multiposte. Ainsi le nombre de progiciels est redescendu à 4 soit :

- Emphi d'Auriga,
- Celcat de Celcat,
- Hyperplanning d'Index Education,
- Heliset d'Helvetius.

Nous avons contacté les acteurs pour lesquels nous n'avions que peu d'informations sur leur produit pour demander de la documentation. Cette étape a été réalisée dès le début du projet, le temps que la documentation arrive. Nous avons repris le projet une fois l'application de consultation sur Intranet développée. Nous avons commencé à remplir la grille de sélection à partir des documents reçus et des informations sur leur site Internet. Ensuite nous avons pu compléter une majeure partie de la grille grâce à un contact téléphonique avec chacun des acteurs.

6 Check liste pour le choix d'un progiciel d'emploi du temps

6.1 Caractéristiques commerciales					
6.1.1 Identifiant logiciel					6.1.2 Commentaires, demandes du cahier des charges
Quel est le nom du logiciel ? Quel est la date de la dernière version ?	Emphi 15.02 2000	Celcat, emplois du temps 2001	Heliset	HyperPlanning 3.0 24 avril 2002	
6.1.3 Généralité					
6.1.3.1 Société					
Nom de la société Adresse Téléphone Contact Fonction du contact	AURIGA 9 cité Trévis 75009 PARIS 01 46 24 86 00 Florence CLAIR Ingénieur commercial	Celcat 22, 23 Mercia Village Torwood Close Westwood Business Park Coventry CV4 8HX Grande-Bretagne (0044)24 76 42 09 93	Helvetius 15, avenue Augustine 92250 La Garenne Colombes 0147690163 Philippe RENARD Responsable commercial	INDEX EDUCATION BP 201 13 322 MARSEILLE Cedex 16 04 9615 21 70 Bernard BASTIDE Service assistance technique	

Nationalité Date de création	Française 16/09/1988	Britannique 1979	Française 6/10/1988	Française 01/02/1992	De préférence française, pour des raisons de proximité.
Effectifs Chiffre d'affaires	Moins de 10 0,9 M€ à 1M€	20 Non communiqué	20 1707 K€ (en 2001, en augmentation)	20 à 30 2,1 M€ - 2,2 M€	
6.1.3.2 Commercialisation					
Nombre de références clients	450	17 multipostes 100 monopostes (en France)	62	150	Important pour montrer la pérennité de l'entreprise.
Références en école d'ingénieur - nom - Adresse - contact	Le groupe ISEN ISEN de Toulon Toulon Petit PERET	Université du Mirail de Toulouse	Ecole des Mines de Douai, Nantes... (la version installée n'est pas la dernière)		
6.1.4 Prestations					
6.1.4.1 Documentation					
La documentation de présentation est-elle en français ?	Oui	Oui	Oui	Oui	Indispensable
Le mode d'emploi est-il en français ?	Oui	Oui	Oui	Oui	
Quel est le support de la documentation ?	Manuel d'utilisation papier	Manuel d'utilisation papier.	Manuel d'utilisation papier.	Manuel d'utilisation papier Version informatique	

6.1.4.2 Contrat de droit d'usage					
Quels sont les types de contrats pratiqués ? - Concession des droits d'usage ; - Location ; - Autres.	Oui Non	Oui Oui, par an avec le nombre de licences illimité	Oui Non	Oui Oui	
6.1.4.3 Formation					
Des formations sont-elles proposées ?	Oui	Oui	Oui	Oui	De préférence inférieur à 1 jours.
Nature de la formation proposée.	Utilisation du logiciel, saisie des données et exploitation	Stage	Cours	Stage	
Durée de la formation.	2 jours conseillés voire 0.5 jour pour l'administration	De 1 à 3 jours	Étalé sur 3 semaines.	1 à 2 jours (1 jour = 6h)	
Lieu de la formation.	Au choix	Au choix	Sur place	Au choix	
Nombre de participants.	8 maximum	10 maximum	5/6 selon les fonctions	6 maximum	
6.1.4.4 Garantie					
Nombre de mois de garantie	6 mois	1 mois	3 mois (souvent offerte jusqu'au 31 décembre)	1 an d'assistance	

La garantie couvre-t-elle :					
- La correction des bogues ;	Oui	Oui	Oui	Oui	
- La correction des défauts de conformité du logiciel aux spécifications annoncées ;	Oui	Oui	Oui	Oui	
- La livraison de nouvelles versions ;	Non (fait partie de la maintenance)	Non	Oui	Oui	
- Autres.					
6.1.4.5 Maintenance du logiciel					
Les sources sont-elles déposées à l'APP ?	Non	Non, mais clause de transfert des sources en cas de faillite	Oui		
Ailleurs ?	Chez leur avocat				
Que couvre le contrat de suivi du progiciel ?					
- La livraison de nouvelles versions ?	Oui Mise à jour des produits	Non			
- L'assistance téléphonique ?	Oui	Oui			
- L'assistance sur site (dans quels délais) ?	Non	Oui			
- La correction de bogues ?	Oui				
- L'évolution du système d'exploitation ?	Logiciel suit les évolutions de l'OS de Microsoft				
- La mise à jour de la documentation.	Oui	Non			
- La télémaintenance des programmes.	Non, mais réflexion				

6.1.5 Coûts du logiciel et des prestations (Prix HT)

Coût de concession des droits d'usages.					< 15 000 € pour 5 ans.
- Pour un utilisateur.	2500 €	2666 €	Hélistet tout compris :	Pour 2 : 5335,72 €	
- Par utilisateur supplémentaire.	400 €	(administrateur)	environ 15000 €	1067,14 €	
- Pour 5 utilisateurs	soit 4100 €	1397 €	Hélisa tout compris :	soit 8637,14 €	
	sinon 4500 € nb	soit 8254€ (sans	environ 40000 €		
	utilisateurs illimité	réduction Ministère			
		de la recherche)			
Coût d'un exemplaire supplémentaire de documentation.				68,60 €	
Coût annuel de maintenance en pourcentage du coût initial de concession du droit d'usage.	23 %	Maintenance : 229 € / personne		17%	
		Soit 15 %			
Coût annuel de location du logiciel.		Sur devis en fonction de la taille de l'établissement.		3048,97 (5 utilisateurs)	
Coût journalier d'un formateur.	998 €		1000 €		
- Chez le client.	frais de déplacement/hébergement	1401 €	Frais de déplacement	1295,82 €	
- Sur le site du fournisseur.		373 €		990,92 €	
Coût des frais de port.		30 €		8 €	

6.2 Caractéristiques techniques

6.2.1 Développement du produit

Quels sont les langages de développement du progiciel ?	Delphi	C ++	C, C ++, PHP, javascript	Delphi	
---------------------------------------------------------	--------	------	--------------------------	--------	--

6.2.2 Spécification de fonctionnement

<p>Mode de fonctionnement du progiciel :</p> <ul style="list-style-type: none"> - Application centrale ; - Client – serveur de données ; - Client – serveurs multi-niveaux ; - Web 	Oui	Oui	Oui	Oui	
<p>Quels sont les briques logicielles nécessaires au fonctionnement du progiciel :</p> <ul style="list-style-type: none"> - SGBDR ; - Serveur HTTP ; - Autres. 	Dbase	Paradoxe 4	MySQL, Oracle ou ... Apache	Format propriétaire	
<p>Quels sont les systèmes d'exploitation supportés par la partie cliente ?</p> <p>Quelles sont les spécifications matérielles nécessaires pour les postes clients ?</p>	Windows NT4, 2000 Pro, XP PC : Pentium II 300 Mhz RAM : 64 MO	Windows (toutes depuis 95) Linux (non garantie)	Soit Windows Soit un navigateur Web	Windows : 95/98/NT4/2000 PC : CPU 500 Mhz RAM : 128 MO DD : 40 Mo Ecran 17'' 1024*768	Macintosh ou Windows
<p>Quels sont les systèmes d'exploitation supportés par la partie serveur ?</p> <p>Quelles sont les spécifications matérielles nécessaires pour le poste serveur ?</p>	Windows NT server, 2000 server Réseau 100 Mbits RAM : 128 MO DD : 50 à 400 Mo	Windows	Tous tant qu'il supporte le SGBDR et le serveur Apache	Windows 98, NT, 2000, XP RAM : 128 MO DD : 10 à 100 Mo	

Quelles sont les structures de stockage des données ?	SGBD	Serveur de fichier	SGBD	SGBD	
Ces systèmes sont-ils propriétaires ?	Oui	Oui		Oui	
6.2.3 Maintenance technique					
Est-il possible de modifier les structures de données proposées (ajout de tables, de champs...)	Non, mais champs paramétrables	Non, mais l'architecture des tables est donnée.	Oui, mais perte de garantie...	Non	
Lors d'un changement de version, le non-écrasement des données personnelles est-il assuré ?	Oui	Oui	Oui	Oui	
Quels sont les méthodes de sauvegarde proposées ?	Possibilité d'enregistrement	Automatique	Multiples	Possibilité d'enregistrement	Indispensable
6.2.4 Administration et supervision applicative					
Le logiciel propose-t-il le support de services d'annuaires compatible LDAP ?	Non	Non	Non	Non	Souhaitable

L'accès au progiciel est-il protégé de façon applicative au niveau de :					
- Ses données ;	Oui	Oui	Oui		
- Ses fonctions ;	Oui	Oui	Oui	Oui	
- Son administration ?	Oui	Oui	Oui	Oui	
Le progiciel gère-t-il la notion de profil utilisateur permettant de définir des autorisations distinctes ?	Oui	Oui	Oui	Oui	Indispensable
Si oui :					
- Quels sont les profils standards ?		Néant			
- Est-il possible d'en ajouter ?	Oui autant que de licences	Oui		Oui	
6.2.5 Intégration et communication applicative					
Le progiciel fait-il partie d'une gamme ou d'une architecture de produits ?	Oui	Non	Oui	Non	Intéressant pour les possibilités d'extension
Quels sont les autres produits ?	ensemble de logiciels pour la gestion de l'enseignement		Tous les modules Hélisha		
Comment les produits de la gamme communiquent-ils entre eux ?	Partage base de données		Partage base de données		

Le progiciel offre-t-il des fonctions d'import/export avec des applications tierces ? - Avec un format d'échange propriétaires ; - Un format d'échange normalisé (EDI) ; - Un format d'échange standard (ASCII) ; - Autres ?	Oui Oui au format Dbase (.dbs) Oui	Oui, par un séparateur de virgule	Oui Oui	Oui Oui	Indispensable
Le logiciel supporte-t-il des protocoles de messageries : - SMTP ; - POP3.	Oui	Oui, il possède un utilitaire qui fait l'interface avec des services mails.	Oui	Non En recherche	Souhaitable
6.2.6 Gestion de l'aide en ligne					
Le progiciel est-il en français ?	Oui	Oui	Oui	Oui	Indispensable
Le progiciel dispose-t-il d'une aide en ligne ? - Par fonction ; - Fenêtre ; ? - Zone ?	Non			Non	

6.3 Caractéristiques fonctionnelles

6.3.1 Caractéristiques générales du système d'emploi du temps

6.3.1.1 Réalisations des emplois du temps

Peut-on saisir et décrire l'ensemble des contraintes :					
- Séquences cours/TD ;	Non	Non	Oui	Non	Souhaitable
- Volumes horaires ;	Oui	Oui	Oui	Oui	Indispensable
- Prérequis ;	Oui	Non	Oui	Non	Souhaitable
- Disponibilités ;	Oui	Oui	Oui	Oui	Indispensable
- Mono-utilisation des ressources ?	Oui	Oui	Oui	Oui	Indispensable
Peut-on saisir et décrire l'ensemble des ressources :					
- Salles ;	Oui	Oui	Oui	Oui	Indispensable
- Intervenants ;	Oui	Oui	Oui	Oui	Indispensable
- Elèves ?	Oui	Oui	Oui	Oui	Indispensable
Peut-on visualiser le planning sur l'ensemble de l'année scolaire ?	Oui période modifiable	Oui	Non	Oui	Souhaitable

Le progiciel propose-t-il des aides à la réalisation ?	Oui : vérification de conflits, proposition de ressources disponibles	Oui : vérification de conflit	Oui	Oui	Indispensable
Le progiciel peut-il intégrer des données d'un syllabus.	Non	Non	Oui (avec de nouveaux modules pour être complet, sinon toutes les contraintes sont intégrées)	Non	Souhaitable
A partir des données en entrée, le progiciel peut-il réaliser une première ébauche d'EDT ?	Non	Non	Oui	Non	Souhaitable
Les ressources disponibles sont –elles proposées automatiquement ?	Oui	Oui	Mode de fonctionnement différent.	Oui	
Deux utilisateurs peuvent-ils accéder et modifier un EDT simultanément ?	Oui	Oui	Oui pour ajouter des données Non pour lancer les calculs	Oui	Indispensable
6.3.1.2 Statistiques d'utilisation des ressources					
Est-il possible de comptabiliser les heures d'utilisations des ressources ?	Oui	Oui	Oui	Oui	Indispensable
Est-il possible de pondérer les heures d'utilisation des ressources ?	Oui	Oui	Oui	Oui	Souhaitable
Le progiciel peut-il détecter les ressources inexploitées ?	Oui	Oui	Oui	Oui	

6.3.1.3 Consultation des emplois du temps

Les EDT sont-ils consultables sur un réseau Intranet.	Oui, module séparé, 3050 € consultation dynamique	Oui, avec Web Publisher 1806 €	Oui	Oui, mais statiques	
Le progiciel propose-t-il un envoie automatique des EDT par mail aux enseignants ?	Oui	Oui	Non, mais les fonctions sont disponibles	Non	

6.3.1.4 Gestion des données d'entrée/sortie

Le progiciel peut-il exploiter un annuaire centralisé des enseignants ?	Non	Non	Oui, avec certain module d'Helisa	Non	Indispensable
Peut-on imprimer des documents :					
- Les EDT hebdomadaires ;	Oui	Oui	Oui	Oui	
- Les EDT sur une année ;	Oui	Oui	Non	Oui	
- Les bilans d'utilisations des ressources ;	Oui	Oui	Oui	Oui	
- Autres.				Possibilités de choisir des états	

6.3.2 Les données de bases

6.3.2.1 Les ressources (les champs non renseignés sont pris en compte par le progiciel)

Peut-on ajouter des ressources non prévues par le progiciel ?	Non	Non		Non, possibilité de regroupement	
---------------------------------------------------------------	-----	-----	--	----------------------------------	--

<p>Le logiciel gère-t-il les propriétés des salles suivantes :</p> <ul style="list-style-type: none"> - Nom ; - Localisation ; - Tel ; - Capacité d'enseignement ; - Capacité d'examen ; - Matériel ; - Groupement de salle ; 	<p>Non A titre indicatif Non</p>	<p>Oui Non</p>	<p>Tous peuvent être pris en compte selon les modules exactement installés.</p>	<p>Oui Non Non Oui</p>	<p>Les données sont non exhaustives...</p>
<p>Le progiciel gère-t-il les propriétés suivantes pour les enseignants :</p> <ul style="list-style-type: none"> - Nom ; - Civilité ; - Code ; - Prénom ; - Bureau ; - Département ; - Tel bureau ; - Tel domicile ; - Fax ; - E-mail ; - Adresse ; - Remarques 	<p>2 2 2 + autres</p>	<p></p>	<p></p>	<p>Oui global</p>	<p></p>
<p>Le progiciel gère-t-il les propriétés suivantes pour les élèves :</p> <ul style="list-style-type: none"> - Nom ; - Année - Type de formation ; - Système de crédit dans le suivi des cours ; - Prénom ; 	<p>Oui Module Phœnix à part</p>	<p>Oui : nom ;prénom Non Oui, mais voir nom</p>	<p></p>	<p></p>	<p></p>

6.3.3 Caractéristiques non fonctionnelles

6.3.3.1 Interface Homme Machine

Le progiciel est-il en mode fenêtré ?	Oui	Oui	Oui	Oui	Indispensable
Les déplacements du cours sur l'emploi du temps peuvent-ils se faire par déplacement de la souris ?	Non, double click changement des horaires	Oui	Non, c'est le logiciel qui place tout.	Oui	Souhaitable
Les consultations d'emploi du temps sont-elles esthétiques et claires ? Utilisent-elles de la couleur ?	Oui Oui	Oui Oui	Oui Oui	Oui Oui	Non vérifiable par téléphone
Peut-on modifier les échelles de temps ?	Oui	Oui	Non	Oui	
Les propositions du progiciel sont-elles bloquantes ?	Selon le paramétrage de l'administrateur.	Selon le paramétrage de l'administrateur.	Oui, le progiciel indique les contraintes qui le bloquent	Selon le créateur	Non souhaitable
Les violations de contraintes sont-elles signalées par des alertes ?	Oui	Oui	Oui	Oui	

7 Conseil sur les logiciels

A partir de la check liste, nous avons pu éclaircir notre point de vue sur les caractéristiques des logiciels. Il nous apparaît que deux types de progiciels coexistent. Emphi et Heliset font partie d'un ensemble de progiciels de gestion de l'enseignement. Ils permettent donc des possibilités extensions et sont plus d'actualités si la politique de l'école est d'intégrer une gestion assistée par ordinateur. Ces deux logiciels sont pour le moment les seuls à prendre en compte (avec l'ajout de certains modules...) la directive européenne de suivi des étudiants par système de crédit.

Heliset nécessite une mention particulière. Les frontières entre Heliset et Helisa, la version complète, sont assez floues et la politique de l'éditeur est bien de se servir d'Heliset comme tremplin pour l'ensemble des ses produits. Heliset est cependant de loin le logiciel le plus complet.

Aucun des logiciels étudiés ne répond parfaitement au cahier des charges. A partir de notre étude le maître d'ouvrage devra donc choisir les orientations qui lui paraissent les mieux adaptées.

Développement de l'application de consultation des emplois du temps

2 Spécifications de la page de saisie des critères

L'application doit permettre la consultation des emplois du temps sur le réseau intranet de l'école. L'utilisateur devra saisir des critères pour délimiter sa recherche. Ensuite un emploi du temps graphique se dessinera en prenant compte les critères sélectionnés.

7.1 Spécifications fonctionnelles

La page de saisie doit permettre à l'utilisateur d'affiner sa requête sur cinq champs.

Tout d'abord, le premier critère concerne la période de sélection de l'emploi du temps. Ce choix doit obligatoirement être renseigné par l'utilisateur. Par défaut, le programme sélectionne la semaine courante. L'utilisateur aura le choix entre cinq périodes :

- la semaine courante
- la semaine suivante
- dans deux semaines
- le mois courant
- le mois prochain

Cette structure devra permettre à l'utilisateur de sélectionner la période plus facilement que s'il devait rentrer des dates de bornage. Après un rapide sondage de certains utilisateurs des précédentes applications, la plupart consulte l'emploi du temps de la semaine courante.

Ensuite, le second critère de sélection correspond à la promotion des élèves. L'utilisateur pourra sélectionner une seule des promotions disponibles ou toutes. Le programme devra tenter de reconnaître, si l'utilisateur est un élève, sa promotion et la proposer par défaut. Si l'utilisateur n'est pas un élève, par défaut, le programme devra sélectionner toutes les promotions.

Le troisième critère est la sélection du nom de la matière. L'utilisateur ne pourra sélectionner qu'une seule matière dans un menu déroulant contenant l'ensemble des matières. Pour améliorer l'ergonomie de la page de saisie un champ texte permettra à l'utilisateur de positionner le choix de la liste sur la matière commençant par le texte écrit.

Le quatrième critère de sélection concerne le choix de la salle. Le menu se présente de la même manière que le précédent. Ce critère de sélection propose ainsi de visualiser l'occupation des salles.

Le cinquième critère permet de sélectionner le nom d'un intervenant ou professeur donnant un cours. Ce menu est défini de la même manière que les deux critères précédents. Si un professeur utilise l'application, celle-ci proposera par défaut son nom dans la sélection. Ce critère permet à un professeur de consulter les cours qu'il doit donner.

Le menu de sélection proposera un bouton pour effacer les champs sélectionnés et un second pour valider les choix de l'utilisateur. En cliquant sur ce dernier bouton, l'application affichera l'emploi du temps en fonction des données sélectionnées.

Pour la consultation d'un emploi du temps, il sera imposé à l'utilisateur de saisir au moins l'un des 4 derniers critères précités. Dans le cas contraire, l'application reviendra sur la page de sélection et un message d'erreur s'affichera.

7.2 Spécifications logiques

Le menu de sélection des critères sera formé d'un formulaire en HTML. La sélection de la période doit laisser cinq choix prédéfinis. Il convient alors d'utiliser un ensemble de boutons radios pour que l'utilisateur ne puisse saisir qu'un seul des choix. Le bouton radio correspondant à la semaine courante sera coché par défaut.

La sélection de la promotion sera également composée de boutons radios. Un bouton supplémentaire sera rajouté pour pouvoir sélectionner toutes les promotions. Le programme sélectionnera le groupe d'appartenance (soit le champ « ou ») de l'utilisateur dans l'annuaire LDAP à partir de son login. Le résultat est sous la forme : « Centre/ Service/ Promo ». Le service permettra de déterminer si l'utilisateur est un élève, dans ce cas le bouton radio de sa promotion sera coché. Si l'utilisateur n'est pas un élève, le bouton radio représentant toutes les promotions sera coché. Un cas d'exception est à noter si l'utilisateur est en option de dernière année, il n'est pas possible de sélectionner celle-ci car cette donnée n'est pas renseignée dans l'annuaire. Dans ce cas, le bouton de toutes les promotions sera coché.

La sélection des critères de la matière, de la salle et du nom de l'intervenant se fera à l'aide d'une liste déroulante. Une valeur supplémentaire sera rajoutée à la liste pour ne pas faire de sélection sur ce critère. Une case contenant du texte permettra à l'utilisateur de positionner le sélectionneur de la liste sur la première valeur commençant par les caractères alphanumériques saisis. Dans le cas où aucun champ ne commencerait par le texte saisi, le sélectionneur se positionnera sur la valeur « Aucune sélection ». Lorsque l'utilisateur sélectionne une valeur dans la liste, celle-ci est ramenée automatiquement dans la case précitée de saisie. Cette fonctionnalité est offerte par des fonctions de script. Le nom de l'intervenant est par défaut celui contenu dans l'annuaire LDAP. Cette donnée correspond au champ nommé « nomget » (où GET est le logiciel d'emploi du temps actuellement utilisé) dans cet annuaire. Le nom de l'utilisateur sera par défaut sélectionné dans la liste et complétée dans la case. Si l'utilisateur est un élève ou ne donne pas de cours, cette donnée n'est pas remplie. Ainsi, par défaut, la sélection sera positionnée sur « Aucune sélection » et la case sera vide.

Le formulaire sera composé de deux boutons, un pour effacer les données sélectionnées et un pour valider. Le programme vérifiera si au moins l'un des quatre derniers critères est sélectionné. Dans ce cas, l'affichage de l'emploi du temps sera lancé. Dans le cas contraire, le message « Sélectionnez au moins un des critères » apparaîtra et affichera la page de saisie.

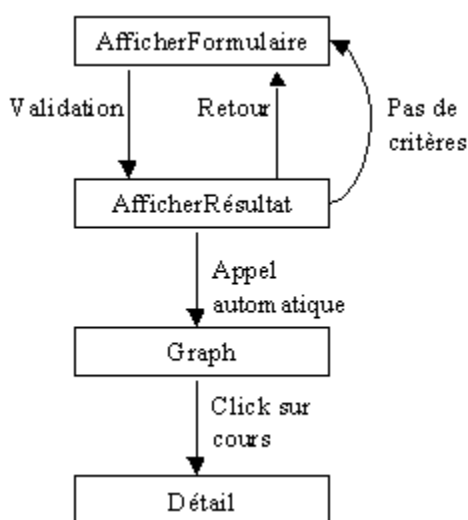
8 La réalisation

8.1 Fonctionnement général du programme

Le programme contient quatre grandes sous fonctions qui sont appelées pour permettre la navigation. Celles-ci sont :

- *AfficherFormulaire* : menu pour afficher le formulaire de saisie,
- *AfficherResultats* : menu qui inclut le résultat de *Graph*,
- *Graph* : Fonction qui construit l'emploi du temps graphique,
- *Detail* : Menu qui affiche le détail de chaque cours.

Figure 1 : Enchaînement des écrans



La sous fonction *AfficherRésultat* n'affiche pas directement l'emploi du temps mais fait appel dans un objet qui représente le résultat de la fonction *Graph*. Cette méthode a été adoptée car un objet graphique doit obligatoirement être appelé par une page HTML pour pouvoir à la fois afficher du texte et des images.

8.2 Description de la page de saisie des critères

La page doit évoluer vers une interface homme machine plus conviviale pour l'utilisateur. Pour cela, il a été décidé de pouvoir sélectionner un critère en un seul click. Les boutons « Filtrer sur » ou « Filtrage large » ont donc été enlevés. Cette interface a été choisie afin de permettre un affichage graphique de l'emploi du temps correctement. En effet, si les critères de sélection sont trop larges, les données à l'écran sont trop nombreuses. Des problèmes de visibilité à l'affichage se posent. Nous avons choisi des critères stricts en tenant compte des cas d'utilisation du logiciel. Un élève consulte habituellement l'emploi du temps de sa promotion pour la semaine en cours ou suivante. Un professeur consulte les cours qu'il doit assurer pour les mêmes périodes. Ces deux types d'utilisation sont largement les plus répandus.

8.2.1 Le choix de la période et de la promotion se fait par boutons radios

Un formulaire a donc été créé pour passer l'ensemble des critères de recherche. Nous avons décidé que la sélection des critères de la période et de la promotion se fasse par boutons

radios. L'utilisateur pourra donc ne sélectionner qu'une seule valeur. L'ensemble des promotions disponibles est issu de la table *XTSEANCES* de la base de données Intranet. Pour sélectionner les valeurs la requête SQL utilisée est : *SELECT DISTINCT PROMOTION FROM XTSEANCES ORDER BY PROMOTION*

Pour créer chaque bouton, il suffit de boucler sur chaque ligne retournée par l'ordre SQL et de créer à chaque fois un bouton radio. Une case permettant de sélectionner toutes les promotions a été rajoutée afin de consulter les emplois du temps des professeurs ou les occupations des salles. Le code Perl qui génère du HTML pour créer un bouton radio concernant la promotion est le suivant :

```
<LABEL FOR="$Valeur"><INPUT ID="$Valeur" TYPE="Radio" Name="Liste$Variable"
Value="$Valeur">$Valeur</LABEL>
```

où : *\$Variable* = 'PROMOTION' et *\$Valeur* représente une valeur retournée par la requête.

La balise LABEL permet de relier le texte avec le bouton. Ainsi le bouton radio peut être coché en cliquant indifféremment sur le bouton ou sur le texte.

Figure 2 : Utilisation de boutons radios dans le formulaire de saisie

CONSULTATION	CRITERES				
PERIODE	<input checked="" type="radio"/> Cette semaine	<input type="radio"/> La semaine prochaine	<input type="radio"/> Dans 2 semaines	<input type="radio"/> Ce mois	<input type="radio"/> Le mois prochain
PROMOTION	<input type="radio"/> FCD-D 2004 <input type="radio"/> FCD-P 2004	<input checked="" type="radio"/> IFI1 2006 <input type="radio"/> IFI2 2005 <input type="radio"/> IFI3 2004	<input type="radio"/> Opt BIO 2003 <input type="radio"/> Opt ECO 2003 <input type="radio"/> Opt GE 2003 <input type="radio"/> Opt GI 2003 <input type="radio"/> Opt GP 2003 <input type="radio"/> Opt GSI 2003 <input type="radio"/> Opt MX 2003	<input type="radio"/> Reservations <input type="radio"/> Aucune	

8.2.2 La sélection des critères de la matière, de la salle et de l'enseignant

La sélection des critères de la matière, de la salle de cours et de l'enseignant se fait dans une liste déroulante. Une cellule de texte a été rajoutée pour faciliter la navigation dans ces longues listes. Ainsi dès qu'il tape un mot, la liste se positionne sur la première valeur commençant par ce mot.

Ce choix a été adopté alors qu'initialement il avait été prévu que l'utilisateur commence à taper un mot puis sélectionne la valeur souhaitée dans une liste. Celle-ci apparaissait dans une nouvelle fenêtre et affichait les valeurs commençant par ce mot. Enfin, la valeur de la liste remplace le mot entré par l'utilisateur dans la fenêtre initiale. Cette méthode a été développée. Cependant comme la liste des valeurs provient de la base de données et nécessite des traitements, le temps d'ouverture de la nouvelle fenêtre prenait environ 2 secondes. Le maître d'ouvrage a préféré s'orienter dans une solution plus rapide.

Celle choisie demande plus de temps à l'ouverture de la page de saisie mais une fois chargée, la navigation est immédiate.

Figure 3 : Choix de la matière, du lieu et de l'enseignant

MATIERE	<input type="text" value="BASES DE DONNEES"/>	<ul style="list-style-type: none">AUTOMATIQUE TP1BASES DE DONNEESBIOTECH. TPBIOTECHNOLOGIE
LIEU	<input type="text"/>	<ul style="list-style-type: none">Aucune sélection0F10F20F3
ENSEIGNANTS	<input type="text" value="ott"/>	<ul style="list-style-type: none">OGE F.ORTEU C.ORTEU J.OTTON E.

8.2.2.1 L'utilisation de scripts Javascript

Le Javascript est un langage simple de programmation qui dynamise les pages Internet en y incorporant des scripts directement interprétés par le navigateur. Ces scripts aident à la navigation et ajoutent de petites animations. Une majorité de sites Internet incorporent d'ores et déjà des scripts dans leurs pages et plus de 90 % des internautes possèdent un navigateur qui interprète le Javascript. De plus, ce langage ne nécessite pas de logiciel supplémentaire à télécharger.

Les instructions Javascript doivent être utilisées avec la balise `<SCRIPT type='text/javascript'>`. Il est possible avec cette balise d'afficher du code HTML si et seulement si le navigateur prend en compte ce langage. Pour cela, on doit utiliser la méthode `write()` de l'objet `document` pour écrire dans la fenêtre du document. Avec une balise `NOSCRIPT`, il est possible d'afficher du code HTML si le navigateur ne prend pas en compte ce langage. Ainsi l'application développée sera consultable sur les navigateurs n'intégrant pas Javascript. Les fonctions de navigation en Javascript ne seront alors pas disponibles.

La fonction nommée *Reporter* permet de ramener le champ sélectionné d'une liste dans un champ texte. Elle prend deux valeurs en entrée, le nom de la liste dont la valeur provient et le nom du texte de sortie. Le langage permet d'atteindre des éléments de la page HTML et chaque élément possède des propriétés et méthodes. L'élément liste possède des options correspondant à chaque ligne et la propriété `selectedIndex` sélectionne le numéro de la valeur sélectionnée. La propriété `value` ramène la valeur de l'option sélectionnée. Ensuite il suffit juste d'affecter cette valeur à l'élément sortie dans le formulaire dont le nom est « origine ». La fonction est détaillée ci-après.

```
function Reporter(liste,sortie){  
    var choix=liste.options[liste.options.selectedIndex].value;  
    origine.elements[sortie].value=choix;  
}
```

Pour utiliser la fonction, il suffit de l'affecter à un événement dans la balise de la liste comme ci-dessous.

```
<SELECT NAME="Liste$Variable" SIZE=4 onClick="Reporter(this,'VAL_RECH_$Variable')">
```

Dans ce cas, la fonction est appelée lorsque l'utilisateur clique sur un élément de la liste grâce à l'événement `onClick`.

Une seconde fonctionnalité permet de sélectionner la première valeur de la liste commençant par la chaîne de caractères entrée par l'utilisateur. Une première fonction nommée *verifTouche* affecte à la variable globale au script *touche* le code décimal de la touche appuyée. Cette variable prend la valeur d'une propriété de l'élément *event* concernant un événement utilisateur. Cette propriété distingue suivant le navigateur : Internet Explorer (*keyCode*) ou Netscape, Mozilla et iCab (*wich*). La fonction est facilement compréhensible mais la différenciation du navigateur montre certaines limites de Javascript. En effet, un certain nombre d'éléments est propre au navigateur.

La fonction *verifTexte* sélectionne la première valeur de la liste commençant par la chaîne de caractères entrée par l'utilisateur. La variable *touche* est alors utilisée pour lancer le traitement si la touche entrée n'a pas été appuyée. Les valeurs en entrée sont le nom de la liste et celui du champ texte. Le principe de l'algorithme consiste à boucler sur le nombre d'éléments de la liste et de comparer la chaîne de caractère avec chacune des valeurs de la liste scindée à la longueur de la chaîne. Si ces deux valeurs sont égales, la liste est positionnée sur la précédente valeur et le traitement s'arrête. Si cette condition d'égalité n'est pas vérifiée, la liste est positionnée sur la première valeur, soit '*Aucune sélection*'.

Pour utiliser ces deux fonctions, on a mis dans une case du formulaire de type texte trois événements :

- *onBlur* : lance la fonction *verifTexte* lorsque le champ est quitté
- *onkeydown* : lance la fonction *verifTouche* en appuyant sur une touche
- *onkeyup* : lance la fonction *verifTexte* en relâchant la touche

```
<SCRIPT LANGUAGE='javascript'>
    document.write("<INPUT TYPE='TEXT' SIZE=30 $ValeurDefaut NAME=VAL_RECH_$Variable
onBlur=\\\" verifTexte('Liste$Variable','VAL_RECH_$Variable');\\\" onkeydown=\\\"document.onKeyDown =
verifTouche(event);\\\" onkeyup=\\\"if (touche) verifTexte('Liste$Variable','VAL_RECH_$Variable');\\\">
</INPUT>")
</SCRIPT>
```

Une fonction permettant de réaliser une mise à zéro du formulaire a été définie en Javascript. Elle propose en plus l'effacement du nom de l'enseignant, écrit par défaut si un professeur se connecte, et le placement de la liste sur la première valeur correspondant à '*Aucune sélection*'.

Le code est le suivant :

```
function MAZ(nomform){
    nomform.reset();
    nomform.VAL_RECH_ENSEIGNANTS.value = "";
    nomform.ListeENSEIGNANTS.selectedIndex = 0;
}
```

Pour appeler cette fonction, on utilise l'événement *onClick* sur une balise *INPUT* de type *button* au lieu d'un *reset*. Dans une balise *NOSCRIPT*, on place un bouton *reset* pour les navigateurs n'utilisant pas le langage.

8.2.2.2 L'utilisation de l'annuaire centralisé de l'école

Le programme recherche dans l'annuaire LDAP, le champ « centre/service/promotion » et le nom de GET. Le premier champ sert à cocher une promotion par défaut et le second à renseigner automatiquement le nom du professeur. Pour ajouter cette fonctionnalité, il faut utiliser le module Net:LDAP de Perl.

Le sous-programme créé est simple. Tout d'abord, un objet LDAP est défini puis se connecte à l'annuaire sur Intranet grâce à la méthode *bind()*. Pour réaliser une recherche, le programme utilise la méthode *search()* avec comme *base* : '*dc=enstimac dc=fr*' et comme *filter*

'*uid=login*'. Le login est récupéré dans une variable d'environnement du serveur. Toutes les informations provenant de LDAP sont récupérées dans un tableau de hachage global au programme nommé *%userLDAP*. Ainsi ces données pourront être utilisées dans l'ensemble du programme. Les clefs de ce tableau correspondent au nom des champs de l'annuaire, soit :

- *uid* : identifiant de l'utilisateur soit le login,
- *ou* : groupe de l'utilisateur,
- *nomget* : nom de l'utilisateur dans le logiciel GET.

Pour récupérer les valeurs de l'annuaire et les affecter au tableau, il faut utiliser la méthode *get_value* sur chacune des valeurs de la recherche. Le tableau *@InfoLDAP* déclaré en global au début contient l'ensemble des champs à récupérer. Ainsi pour rajouter un champ, il suffira d'ajouter une valeur à ce tableau.

La valeur contenue dans *\$userLDAP{ou}* est une chaîne de caractères de la forme « centre/service/promotion ». Pour un élève en IFI 2003, cette valeur sera : « /DE/ELV/IFI2003 ». Ainsi ce champ nous indique si l'utilisateur est un élève. En utilisant, fonction *split()*, il est possible de découper la précédente chaîne de caractères selon les /. Si le 3^e élément du tableau (le premier est vide) est bien 'ELV', deux variables représentant le cycle (IFI, IFC...) et l'année sont alors créées et affectées en utilisant la fonction *substring()* pour scinder le dernier caractère du tableau précité. Pour cocher la promotion par défaut, on compare les 3 premiers caractères du champ promotion issu de la base de données avec le cycle et les 4 derniers avec l'année. Il a été décidé d'utiliser les expressions régulières pour réaliser cette comparaison.

Cependant le script ne fonctionne que sur les individus en formation initiale et dans les 3 premières années. En effet, il n'est pas possible pour les élèves en quatrième année de présélectionner leur option puisque cette information n'est pas renseignée dans LDAP. De même, les élèves en formation continue par distance de 2004 ne sont pas répertoriés dans LDAP.

Le case texte représentant le nom de l'enseignant contient par défaut la valeur *\$userLDAP{nomget}*. Dans le cas d'un étudiant ou en général d'un utilisateur n'assurant pas de cours, cette donnée est vierge. Ensuite, le programme sélectionne dans la liste des enseignants provenant de la base de données, la valeur correspondant à l'enseignant. Pour cela, à chaque ajout d'une donnée dans la liste, il compare ces deux chaînes de caractères précitées et si elles sont identiques sélectionne alors la ligne par la commande HTML suivante :

```
<OPTION SELECTED VALUE="$donnee">$donnee</OPTION>
```

Si le nom de GET est vierge, alors la première ligne '*Aucune sélection*' est saisie.

8.2.2.3 La vérification sur la saisie du formulaire

Une fonction nommée *Vérification* sert à vérifier si au moins l'un des critères suivant : promotion, matière, lieu et enseignant, est renseigné. Les valeurs qui sont réutilisées lors de la validation du formulaire sont, pour la promotion, la valeur cochée et pour les autres critères, la valeur sélectionnée dans la liste.

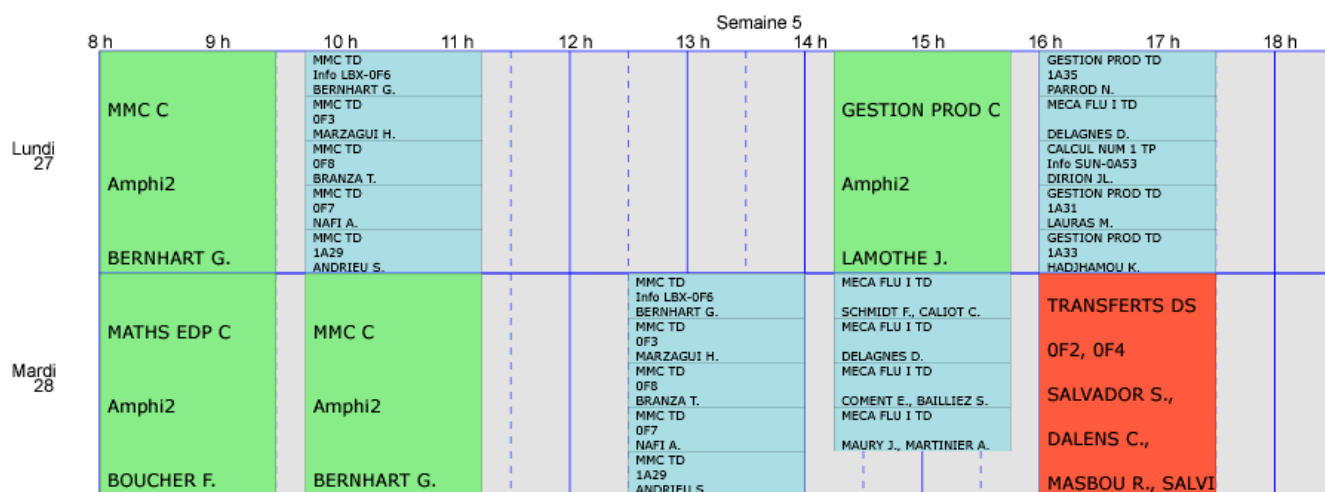
Le sous programme compte le nombre de critères contenus dans *@VariableObligatoire*, c'est-à-dire les 4 critères précités, qui sont vides. Si ce nombre atteint le nombre d'éléments du précédent tableau, un message d'erreur est retourné. Le tableau de hachage *%Erreur* défini au début contient l'alerte à afficher.

9 Spécifications de la page de consultation

A partir des critères saisis par l'utilisateur et de la base de données des cours, l'application génère une image attrayante des cours.

9.1 Spécifications fonctionnelles

L'apparence de l'emploi du temps doit ressembler à la présentation imprimée. Le principe sera donc conservé : chaque jour sera présenté par une ligne, chaque heure par une colonne. Nous présenterons uniquement les jours travaillés et les heures de 8h00 à 18h30 (les heures par des traits pleins, les demi-heures par des traits pointillés). Les informations sur le jour seront affichées en marge à gauche, les heures seront écrites au début de chaque semaine. Les cours seront figurés par des rectangles dont les cotés droit et gauche indiqueront respectivement les heures de début et de fin du cours. Pour les cours qui ne concernent qu'une partie de la promotion, le rectangle présentera une hauteur proportionnelle à la fraction d'élèves considérée. Ainsi un groupe de TD (et il y en a 5 pour les IFI2 2005) représentera 1/5 de la hauteur de la ligne du jour et son placement se fera en fonction du numéro du groupe : le groupe 3 se retrouvera au milieu, tandis que le groupe 5 sera tout en bas.



A l'intérieur des rectangles seront écrites les informations sur les cours. Celles-ci dépendront des critères sélectionnés : seront affichés les paramètres variables (ainsi si vous avez demandé l'emploi du temps de la promotion IFI2 2006, le professeur, la salle et la matière s'afficheront, au contraire si la sélection a été faite sur la matière, c'est la promotion qui s'affichera). Pour une consultation dont les critères sont autres qu'une promotion ou une matière, le découpage par groupe n'est plus graphique (une salle ou un professeur sont occupés entièrement même par une fraction de promotion...), le numéro du groupe sera alors affiché dans le texte du rectangle.

Pour une meilleure visibilité lors de la consultation, les différents types de cours seront singularisés par une couleur. Ce code des couleurs sera rappelé par une légende en tout début de document :

LEGENDE : Visite (orange) TD (light blue) Cours (green) Projet (yellow) TP (purple) Divers (pink) DS (red)

La première ligne du document sera une phrase récapitulative des éléments sélectionnés sur la feuille de saisie. Cette ligne permettra de toujours savoir quel est l'emploi du temps consulté :

- Emploi du temps de la semaine courante pour la promotion IFI2 2005 .

L'affichage dans chacun des rectangles de cours ne pourra, pour des raisons de lisibilité, montrer l'intégralité des informations. Pour permettre néanmoins un accès à celles-ci, chacun des rectangles des cours sera un lien qui permettra l'affichage de toutes les informations :

Champs	Valeurs
SEMAINE	5
PERIODE	23
DATE_SEANCE	2003-01-28
PROMOTION	IFI2 2005
OPTION1	
GROUPE	2/5
CODE	MMC TD
SEQUENCE	14
TEXTE	
DUREE	1:30
TYPE	TD
DEBUT	12h30
FIN	14h00
LIEU	OF3
ENSEIGNANTS	MARZAGUI H.
STATUT_ENSEIGNANT	Interne
THEME	Genie mecanique
MATIERE	MMC TD

Pour permettre une navigation plus facile, des boutons de navigations entre les semaines seront incorporés directement dans la page d'affichage. Il sera ainsi possible de changer de semaine sans revenir à la page de saisie. Un lien vers cette page, pour modifier de manières plus importantes les critères, sera également proposé :

[Retour à la sélection](#)



La taille de l'affichage des cours s'adaptera automatiquement à la taille de la fenêtre du navigateur. L'affichage sera programmé pour montrer automatiquement 4 jours en même temps, l'affichage des autres jours sera possible grâce au défilement de la fenêtre. Cette taille est un compromis entre visibilité et lisibilité.

9.2 Spécifications logiques

Pour réaliser l'ensemble des choix graphiques exigés, nous avons choisi d'utiliser un format de dessin vectoriel. Un document de ce type est composé d'objets, tels que des cercles, des traits et des polygones. Ces éléments sont définis par leurs propriétés, ce qui implique que quelque soit ensuite l'agrandissement ou la déformation, ils paraîtront correctement. Ce n'est pas le cas du dessin *bitmap* puisqu'il travaille sur des pixels et l'agrandissement de l'image augmente leur taille. Chaque objet constitue un élément identifié pouvant être sélectionné, déplacé et modifié indépendamment des autres. Les palettes de fond et de trait permettent respectivement de remplir les objets d'une couleur ou d'un dégradé et de définir la couleur, le motif et l'épaisseur de leur contour.

Ces spécificités prennent en compte de manière simple les demandes. Chacun des rectangles des cours devient un objet indépendant, sur lequel il est possible de créer un lien (pour donner l'accès à toutes les informations) et dont l'ensemble des aspects peut être défini (pour choisir la couleur, la taille...).

Pour créer ce dessin vectoriel nous avons choisi d'utiliser le module SVG de Perl. En effet, SVG est un format de dessin vectoriel gratuit et libre.

Le dessin est inclus dans une page HTML. Les boutons de navigation entre les semaines et le lien vers la page de saisie sont réalisés dans ce langage qui est lui-même généré dynamiquement par un script CGI en Perl.

Les informations nécessaires à la fabrication de l'emploi du temps sont recherchées grâce à l'appel d'une table (XTSEANCES) de la base de données.

10 La réalisation

La programmation de l'affichage de l'image SVG se compose de deux parties principales. La première partie s'occupe de l'environnement HTML et prépare l'affichage de l'emploi du temps au moyen d'une balise « *embed* ».

La deuxième partie, lancée à partir de la balise « *embed* » gère l'ensemble de la création de l'image.

A ces deux parties principales s'ajoutent deux fonctions secondaires, l'une qui construit les boutons de navigation entre les périodes, l'autre qui s'occupe de l'affichage des informations complètes d'un cours.

Ce découpage est imposé par les différentes fonctions logiques assurées par chaque fonction. Une première ébauche du programme ne requerrait qu'un seul sous programme, mais l'image SVG était écrite dans un fichier. Pour permettre la consultation par plusieurs utilisateurs simultanément, le deuxième sous programme a été réalisé. Il élimine la nécessité du fichier.

10.1 Le sous programme *AfficherResultat*

La première tâche exécutée par cette fonction est l'appel à *Verification()*. Cette fonction est détaillée précédemment. Elle se contente de vérifier que les paramètres saisis sont conformes à la réalisation d'un emploi du temps.

La deuxième action est l'appel de la fonction *BoutonChangementSemaine*. Celle-ci a pour but de générer les boutons de navigation entre périodes, d'écrire le lien de retour vers la page de saisie et d'écrire la phrase récapitulative des choix réalisés par l'utilisateur.

Sa première action est de récupérer, avec la fonction *param*, le nom de la période en cours de consultation. On construit alors le nom des boutons qui permettront de changer de période. La construction de ces noms est indispensable, nous ne pouvons afficher directement les noms que nous leur avons affectés. En effet, ils sont tous précédés d'une lettre pour pouvoir avoir un classement logique dans les boutons de la page de saisie.

Ces noms sont exploités dans deux formulaires. Les formulaires renvoient en mode POST toutes les données qui avaient été données par la feuille de saisie en remplaçant la valeur de *ListeSEMAINE* par celle de la période souhaitée, puis réactivent l'ensemble des fonctions d'affichages.

Le lien vers la page de saisie est construit simplement avec une balise `<a>`.

La phrase récapitulative est construite avant l'affichage du formulaire, chacune des valeurs que l'on peut transmettre est interrogée. Toutes celles définies sont écrites avec une petite présentation.

La fonction *AfficherResultat* reprend la main. Elle construit la partie « where » de la requête. Elle est réalisée de manière classique par concaténation. On profite de la boucle pour préparer l'appel de la fonction *Graph*. L'appel de cette fonction est préparé au moyen des fonctions du module URI.

La requête est alors envoyée. Elle est exploitée pour compter le nombre de jours et de semaines qu'inclus la demande de l'utilisateur. Ce comptage est indispensable pour l'ouverture d'une image de taille suffisante.

La fonction se conclut par l'écriture de la balise « embed » :

```
print qq(<p><embed height="$a%" width="100%" src=$uri><p>);
```

ou celle de l'avertissement qu'aucun emploi du temps n'est à afficher.

La balise « embed » est ainsi formée pour assurer un bon affichage de l'emploi du temps. Le \$a est calculé afin de n'afficher que 4 jours en même temps.

La variable \$uri contient l'appelle à la fonction *Graph*. En mode GET sont passées toutes les variables nécessaires à la réalisation de l'image, ce sont les mêmes que celles transmises à *AfficherResultat*.

10.2 Le sous-programme Graph.

Voici les actions qu'il réalise :

- Connexion à la base de données :

```
my $dbh = EMAC::ConnexionOracle();
```
- Définition des styles pour le dessin de l'emploi du temps. Exemple des rectangles des cours :

```
my %cours_style = (  
    'fill'=>'white',  
    'stroke'=>'black',  
    'stroke-width'=>'0.2',  
    'fill-opacity'=>1,);
```

- Définition des variables de présentation. Ce sont ces variables qu'il faudra changer pour faire varier la taille de l'image (d'autres variables sont définies de manière globale au tout début du script).
- La réalisation de la requête est presque la même que celle de la fonction précédente. On exploite les réponses ligne par ligne. Chacune de ces lignes représente un cours.
 - L'affichage se réalise de la façon suivante. Chaque cours est analysé pour savoir s'il appartient à la même semaine et au même jour que le cours précédent. Si ce n'est pas le cas, un nouveau cadre est réalisé. L'écriture d'un nouveau cadre produit un décalage vertical d'une hauteur fixe.

Les cadres des jours et les lignes verticales des heures sont réalisés grâce à des méthodes du module SVG de Perl, à savoir « rect » et « line » :

Exemple de « line » :

```
$svg->line(x1=>$longEDT/$nbligne*$i+$rectorgx,  
          y1=>$coordy,x2=>$longEDT/$nbligne*$i+$rectorgx,  
          y2=>$hautjour+$coordy,  
          style=>{%line_style});
```

- Chaque cours est dessiné sous la forme d'un rectangle (toujours avec la méthode « rect »), le texte est écrit avec la méthode « text ».

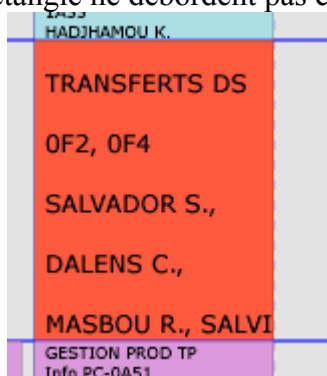
•
Traitement des textes :

L'écriture des textes à l'intérieur des rectangles a été un des points les plus problématiques. Tout d'abord, la place à l'intérieur est limitée, nous avons donc sélectionné les textes importants à afficher : ceux qui ne sont pas sélectionnés par l'utilisateur.

Ensuite la taille des rectangles est très variable en fonction des groupes. La police des caractères utilisée est donc de plus en plus petite. Nous avons fixé la taille de la police pour qu'elle soit la plus grande possible dans les petits rectangles, puis elle grandit progressivement.

Malgré tout, les champs peuvent dépasser des cases. C'est le cas en particulier pour les devoirs où de très nombreux noms d'enseignants sont cités. Quand c'est possible nous avons découpé ce champ en plusieurs lignes.

Nous avons réalisé une fonction de « clipping » sur SVG de manière à ce que les textes de taille supérieure à celle de leur rectangle ne débordent pas et restent masqués :



Le nom des enseignants, ou d'autres variables comme le numéro de séquence d'un TD, ne sont donc consultables à coup sûr que grâce à la fonction *Détail* :

Champs	Valeurs
SEMAINE	5
PERIODE	23
DATE_SEANCE	2003-01-28
PROMOTION	IFI2 2005
OPTION1	
GROUPE	
CODE	Transferts DS
SEQUENCE	2
TEXTE	
DUREE	1:30
TYPE	DS
DEBUT	16h00
FIN	17h30
LIEU	OF2, OF4
ENSEIGNANTS	SALVADOR S., DALENS C., MASBOU R., SALVIGNOL P., MOLEINS G.
STATUT_ENSEIGNANT	Interne
THEME	Genie procedes
MATIERE	TRANSFERTS DS

Nombre de groupes :

La base de données renseigne pour les groupes de TD de la manière suivante : 2/5, soit le deuxième sur les cinq existants.

Mais les cours électifs sont détaillés CE et les cours modulaires CM. Le nombre total de ces cours n'est donc pas renseigné alors que les informations CE et CM sont redondantes.

L'information a été fixée dans les sources.

10.3 Le sous programme Detail

Le sous programme *Détail* est assez simpliste.

Au moyen d'une requête sur la base de données, nous écrivons dans un tableau toutes les informations sur un cours.

Le lien qui commande l'ouverture de cette fenêtre est réalisé pour chacun des blocs de cours grâce à la méthode « anchor » de SVG. Le lien qui lui est affecté passe en mode GET toutes les données nécessaires pour identifier le cours. Au moyen de ces données, nous sollicitons une nouvelle fois la base de données pour recueillir toutes les informations qu'elle possède sur ce cours. Ces dernières sont alors affichées dans un tableau.

Conclusion

3 Conclusion sur la recherche de logiciel :

La recherche de logiciel est mitigée.

- Nous n'avons trouvé aucun nouveau logiciel sur le marché ; ceux détectés par les maîtres d'ouvrages sont les plus pertinents.
- Aucun ne répond parfaitement au cahier des charges, tous possèdent leur propre lacune.
- Le temps imparti pour le projet ne permettait pas de réaliser des tests. Les capacités d'ergonomie n'ont donc pu être validées.
- Heliset se détache comme étant le progiciel le plus complet. Il est également le plus cher, et de très loin, et le plus lourd à installer. Le choix de ce logiciel impliquerait à terme une migration vers l'ensemble du système Helisa. Ce choix dépasse largement le cadre d'une application de gestion d'emploi du temps.
Néanmoins même Heliset possède certain manquement vis à vis du cahier des charges (lien dynamique sur LDAP, vision à l'année...)
- Les trois autres progiciels nous semblent de capacité équivalente. Le choix d'Emphi, de la société Auriga, semble alors le plus pertinent étant donné ses possibilités d'extension à travers toute la gamme de logiciel de leur constellation.

Nous ne pouvons donc pas conseiller strictement un logiciel plutôt qu'un autre, la check liste que nous avons réalisée résume les capacités de chacun des logiciels et nous paraît être une bonne base pour leur choix.

La suite logique de notre projet serait de choisir un des progiciels et de le tester ou de consulter des références.

11 Conclusion sur l'application de consultation

Le projet est largement abouti au vu du cahier des charges :

- Elle est plus ergonomique, il y a moins de boutons à sélectionner, on peut naviguer directement entre les semaines
- L'affichage est graphique, coloré, proche des impressions papiers montrés en référence
- L'ensemble des données d'un cours est accessible.

Les limites sont :

- la non prise en compte de demandes tardives (notamment celle de M Otton de pouvoir consulter l'occupation de toutes les salles informatiques)
- Le temps de chargement accru
- L'impossibilité de consultation avec Mozilla
- La mise à jours de quelques lacunes des bases de données :
 - La non prise en compte des options en dernière année dans l'annuaire LDAP
 - Le nombre des cours électifs et modulaires n'est pas renseigné dans la base de données et a dû être fixé dans le programme.

12 Conclusion en terme d'apport personnel

Nous nous sommes très bien entendus tout au long du projet, très bonne ambiance et coordination au sein de l'équipe.

Nous avons appris de nouvelles possibilités techniques (Javascript, SVG...) et nous avons consolidé nos acquis de cours.

La gestion du projet dans le temps est une réussite. La programmation a été poussée au-delà du cahier des charges.

Bibliographie

4 SVG

- http://www.euroclid.fr/Cours_SVG/plan.htm : très bon cours en français de SVG
- <http://www.w3.org/TR/SVG/types.html#ColorKeywords> : pour le choix des couleurs...
- <http://www.w3.org/TR/SVG/index.html#minitoc> : ce qu'il y a de plus complet
- <http://www.adobe.com/svg> : pour le viewer SVG
- <http://roasp.com/tutorial/index.shtml> : le meilleur tutorial que j'ai trouvé. Idéale pour la prise en même et très rare source d'exemple de SVG généré par du Perl
- <http://www.roasp.com/man> : documentation du module Perl de SVG (un peu légère toutefois...)

13 Javascript

- <http://selfhtml.selfhtml.com/fr/javascript/index.htm> : documentation sur Javascript référant tous éléments et le détail de compatibilité avec les navigateurs.

14 Perl

- <http://www.enstimac.fr/Perl/> : la documentation Perl en français
- C:\perl\html\index.html : la documentation Perl originale propre au module

15 HTML

- <http://www.enstimac.fr/~gaborit/html/w3c-html-4.0-fr/contents.html> : une documentation HTML référant tous les éléments HTML

16 CXP

- <http://www.exp.fr/adherent> : consultation de la base sur Internet
- Pack expert sur les ERP.

17 Information sur les progiciels

- <http://www.celcat.com> et documentation
 - <http://www.auriga.fr> et documentation de l'ensemble des produits de la constellation
 - <http://www.helvetius.com> et documentation d'Heliset et d'Helisa
 - <http://indexeduc.artinternet.fr/com/index.html> et documentation d'Hyperplanning 3.0
 - <http://www.csi.uvsq.fr/services/gestion/autres/celcat/> : comparatif entre Celcat et Hyperplanning réalisé par le CSI pour l'université de Versailles
- Nous ne référençons pas les sites des pistes que nous avons abandonnées.

Annexes

- Diagramme de Gantt.
- Documentation des progiciels (à part)

