



ÉCOLE DES MINES D'ALBI
C A R M A U X

ANALYSE ET DÉVELOPPEMENT D'UNE APPLICATION DE SUIVI DU PUBLIC ADULTE EN STAGE DE FORMATION CONTINUE

Rapport de TFE

Ministère de l'Agriculture, de l'Alimentation, de la Pêche et des
Affaires Rurales

Tuteur école :
Paul GABORIT

Tuteur entreprise :
Serge ANDRIEU



Thomas VAN OUDENHOVE

IFI 2003

— Février – Août 2003 —



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

ANALYSE ET DÉVELOPPEMENT D'UNE APPLICATION DE SUIVI DU PUBLIC ADULTE EN STAGE DE FORMATION CONTINUE

Rapport de TFE

Ministère de l'Agriculture, de l'Alimentation, de la Pêche et des
Affaires Rurales

Tuteur école :
Paul GABORIT

Tuteur entreprise :
Serge ANDRIEU



CERIT Toulouse, BP 68
31326 Castanet Cedex

Thomas VAN OUDENHOVE

IFI 2003

— Février – Août 2003 —

Avant-propos

Conventions de notation

- En général, les mots en PETITES CAPITALES font l'objet d'une explicitation dans le glossaire, page 36 ;
- les références à la bibliographie (p. 37) sont faites entre crochets [], par un numéro ;
- les mots en POLICE CAPITALE SANS SERIF désignent des objets relatifs à la base de données : base de données elle-même, tables, ... ;
- en *caractères penchés*, un mot est relatif à un objet POWERBUILDER ;
- dans un encadré, le mot est relatif à un bouton, cliquable par l'utilisateur ;
- le code PowerScript sera présenté comme montré ci-dessous : encadré, les commentaires (ligne commençant par //) en bleu, mots-clé en rouge et types de données en violet, avec une police de type `typewriter`.

```
// Un exemple de code PowerScript
this.script("retourne un string")

return "c'est vrai"
```

Remerciements

Je tiens en premier lieu à remercier mon tuteur, Serge ANDRIEU, mon chef de projet, Louis CASTELLAN et le chef du service qui m'a accueilli, Pierre LAMOTHE, pour m'avoir permis de réaliser ce stage, et pour toute l'aide technique et fonctionnelle qu'ils m'ont apportée.

Je voudrais ensuite remercier mon collègue de bureau, Stéphane DHUR, ainsi que David BLANC, pour toute leur aide sur l'environnement de développement POWERBUILDER et sur le *framework* ODIN, sans oublier Maud LAMARQUE, en particulier sur le partage d'objets.

Mes remerciements vont aussi à Thierry DESCONS et Frédéric BRONNERT, qui m'ont aidé pour la modélisation et la création de la base de données sur des points techniques.

Enfin, je souhaite remercier tout le service des Examens au CERIT, pour leur soutien et leur bonne humeur : Isabelle, Cédric, David, Henri-Pierre, Matthieu, Éric, Philippe, Marie-Laure, Thierry et Geneviève, ainsi que les sus-nommés.

J'ai failli oublier mon tuteur école, Paul GABORIT, que je remercie pour l'aide qu'il m'a accordée, en particulier en ce qui concerne le rapport et la soutenance.

Résumé

L'enquête sur la Formation Professionnelle Continue permet, chaque année, d'avoir un aperçu statistique de toutes les formations dispensées à des adultes dans les centres de formation dépendant du MAAPAR. Actuellement effectuée grâce à l'ancien logiciel (FPC), une nouvelle application FPC doit permettre de répondre aux nouveaux besoins en permettant l'utilisation de nouvelles technologies, l'interaction avec d'autres applications ou bases de données en exploitation, l'historisation des données, . . .

L'utilisation des nouvelles technologies de l'information et de communication (accès *Web*, développement en technologie objet, . . .) permettent d'éviter la circulation de nombreux papiers. Les interactions sont assurées dans la base de données par des vues vers la base de référence de la DGER, et dans l'application par l'utilisation d'objets de l'application de référence, GENOME, ce qui assure la cohérence des données et des objets. Enfin, l'historisation des données est assurée par la base de données, permettant ainsi la comparaison des données sur plusieurs années, impossible avec l'existant.

Pour mener à bien ce projet, l'activité a été planifiée au cours des six mois de stage à trois reprises : au début du stage (mi-février), début avril et début juin. Ces affinages de plannings m'ont permis d'analyser les échecs sur les prévisions de temps (développement) mais aussi les réussites (analyse).

Table des matières

Avant-propos	i
Résumé	ii
Table des matières	iii
Table des figures	vi
I Rapport d'action	1
1 Objet	2
1.1 L'enquête FPC	2
1.2 Objectifs du projet FPC2	2
2 Conclusion	3
2.1 L'analyse	3
2.2 Le développement	3
2.3 Le respect du planning	3
3 Justification	4
3.1 Comprendre le besoin du client	4
3.2 Rédaction des spécifications	4
3.3 Modélisation	5
3.4 Développement	5
4 Rappel de la démarche	6
4.1 Les phases du projet	6
4.2 Analyse des écarts	6
5 Suites à donner	7
5.1 Finalisation de l'application	7
5.2 Prise en compte de données individuelles	7
II Rapport d'études	8
6 Introduction	9
6.1 Présentation du Ministère de l'agriculture	9
6.2 Le projet <i>Safran</i>	9
6.3 Le Travail de Fin d'Études – le projet FPC2	9

6.3.1	L'enquête sur la Formation Professionnelle Continue	9
6.3.2	Le logiciel existant	10
6.3.3	La nouvelle application	11
6.4	Organisation du rapport	12
7	Spécifications de l'application	13
7.1	Découpage en fonctionnalités	13
7.2	Droits, utilisateurs et sécurité	13
7.3	Spécifications techniques	14
7.3.1	Saisie par Internet	14
7.3.2	Importation des données de CIGALE	15
7.3.3	Consultation et vérification des données	16
7.3.4	Présentation des résultats	17
7.3.5	Historisation	17
7.4	Choix d'une architecture	18
7.4.1	Mode client / serveur	18
7.4.2	Mode <i>Full Web</i>	18
7.4.3	Éléments pour le choix	19
8	Modélisation des données et procédures	20
8.1	Élaboration du Modèle Conceptuel des Données	20
8.2	Passage au Modèle Physique, puis génération des scripts	21
8.3	Schématisation de procédures	22
8.3.1	Importation des données de CIGALE	22
8.3.2	Procédure complète – de la saisie à la consultation	23
9	Développement	25
9.1	Présentation de l'environnement de développement	25
9.1.1	PowerBuilder	25
9.1.2	ODIN, <i>framework</i> du Ministère de l'Agriculture	25
9.1.3	Le partage de code	26
9.2	Avantages et inconvénients de ces outils	27
9.2.1	PowerBuilder	27
9.2.2	ODIN	27
9.2.3	Le partage de code	27
9.3	Les principales difficultés rencontrées	28
9.3.1	Réalisation d'un assistant	28
9.3.2	Rapatriement	29
10	La gestion de projet	30
10.1	Découpage du projet en tâches	30
10.2	GANTT prévisionnels et mises à jour	30
10.2.1	Prévisions de la première semaine	30
10.2.2	Diagramme remis à jour en milieu de stage	32
10.2.3	Prévisionnel pour la fin du stage	32
10.3	Analyse des écarts	33
11	Conclusion	35
11.1	Avancement du projet	35
11.2	Mes objectifs pour le stage	35

Glossaire	36
Bibliographie	37
Références bibliographiques	37
Références Internet	37
III Annexes	38
A Organigrammes	39
B <i>Fiche objet</i> de l'objet session	41
B.1 Description générale	41
B.2 Description détaillée	41
B.2.1 Arbre et liste	41
B.2.2 Habilitations	41
B.2.3 Fenêtre de propriétés	42
B.2.4 Assistant	44
B.2.5 Règles de gestion	44
C Modèles Conceptuels de Données	46
D Modèle Physique des données	47
E Scripts de création de la base de données	48
F Conventions de notations en développement	49
G Extraits de code	50
G.1 Initialisation de l'affichage	50
G.2 Correction de l'affichage	50
G.3 Enchaînement des onglets	51
G.4 Rapatriement	51
H Diagrammes de GANTT	53
I À propos de ce document	55

Table des figures

6.1	Synoptique de la procédure actuelle	10
6.2	Synoptique de la nouvelle procédure	11
7.1	Spécifications techniques de l'onglet <i>Saisie des données</i>	15
7.2	Processus de transfert de données depuis CIGALE	17
7.3	Architecture de l'application en mode client/serveur	18
7.4	Architecture de l'application en mode <i>Total Web</i>	19
8.1	Schéma des flux – intégration de données provenant de CIGALE	22
8.2	Graphe des flux – enquête FPC	24
A.1	Organigramme du MAAPAR	40
A.2	Organigramme du CERIT	40
B.1	Enchaînement des écrans de l'assistant « propriétés d'une session »	45
	MCD, première version	46
	MCD, version n° 4	46
	MCD, version n° 8	46
	MCD, dernière version	46
	MPD, version définitive	47
H.1	Diagramme de GANTT initial (au 17 février 2003)	53
H.2	Diagramme de GANTT intermédiaire (au 03 avril 2003)	54
H.3	Dernier diagramme de GANTT (au 02 juin 2003)	54

Première partie

Rapport d'action



Chapitre 1

Objet

1.1 L'enquête FPC

Cette enquête est menée tous les ans par la DGER au travers des SRFD auprès des centres de formation pour adultes. Elle porte sur des données agrégées par session (donc par groupe de stagiaires). Elle remplit plusieurs objectifs pour la DGER :

- évaluer l'activité des centres dans le domaine de la formation continue ;
- disposer d'un outil de suivi de la carte des formations ;
- adapter les ressources aux besoins (humains et financiers).

Ces objectifs sont déjà remplis par un logiciel. Cependant, ce logiciel est obsolète : il n'est plus maintenu, est développé sous DOS, et n'utilise pas les technologies de communication récentes (bases de données en réseau, ...).

1.2 Objectifs du projet FPC2

Étant donné ce logiciel vieillissant (développé en 1990, aucune évolution depuis 1993), il devenait nécessaire de développer une nouvelle application. Cette application doit permettre à tous les utilisateurs de se connecter à la même base de données, et éviter aux SRFD de saisir les données pour les centres (cf. section 6.1, page 10).

De plus, une fonctionnalité attendue du nouveau logiciel est de pouvoir comparer les données d'une année sur l'autre, ce qui était impossible avec l'existant.

Enfin, le développement d'une nouvelle application permet de contacter les utilisateurs pour redéfinir leurs besoins. En effet, les besoins des utilisateurs ont évolué en dix ans et l'existant ne correspondait plus tout à fait à leurs attentes. En particulier, il a été possible de remettre à jour la liste des éditions utilisées.

Dans le cadre de mon stage, l'objectif visé est la réalisation d'un prototype. Ce stage sera sanctionné par une présentation de l'application réalisée le 31 juillet 2003 à la DGER, à Paris, et la suite à donner au projet sera décidée lors de cette réunion avec le comité de suivi.

Plus généralement, le projet FPC2 comprend des objectifs qui n'étaient pas encore bien définis à mon arrivée. Le premier objectif porte sur l'enquête sur l'année en cours, qui sera réalisée début 2004. Un autre stagiaire prendra donc la relève pour finaliser le prototype réalisé, de façon à pouvoir traiter l'enquête sur l'année 2003 avec la nouvelle application. Le deuxième objectif est à long terme, il consiste à prendre en compte des données individuelles sur les stagiaires, afin de pouvoir effectuer des traitements plus précis. Cependant ce deuxième objectif demande plus de ressources, d'après l'expérience du CERIT sur les élèves en formation initiale.

Chapitre 2

Conclusion

2.1 L'analyse

L'analyse a été menée à sa fin. Le CERIT dispose d'un document récapitulant les spécifications techniques de l'application [6]. Cependant, l'historisation est prise en compte par la modélisation et l'architecture a été dictée par les compétences du terrain. Il manque toutefois une partie à l'analyse : tout ce qui concerne l'exploitation des résultats, qui ne rentre pas vraiment dans le cadre du prototype.

2.2 Le développement

Le développement a connu des hauts et des bas, mais l'application est en passe de fonctionner relativement correctement pour le module de saisie et vérification des données. Le passage en mode *Web* devrait être imminent, et le prototype devrait pouvoir être présenté à la DGER le 31 juillet 2003. Étant donné que l'application est un prototype, seulement quelques éditions seront présentées, mais l'application fonctionnera.

2.3 Le respect du planning

Certaines tâches prévues en début de planning ont disparu et d'autres n'avaient pas été planifiées. Cependant, sur les macro-tâches, le planning a été plutôt bien respecté en ce qui concerne l'analyse et la modélisation. Cependant, la prévision des temps de développement a été tout à fait arbitraire au départ par méconnaissance de l'environnement de développement.

Chapitre 3

Justification

Réaliser une application qui correspond aux souhaits de la maîtrise d'ouvrage est un travail très difficile. Cet ouvrage se décompose en plusieurs étapes :

1. bien comprendre le besoin du client ;
2. rédiger les spécifications générales puis détaillées qui permettent de répondre à ce besoin ;
3. modéliser les données avec une méthode adaptée, et éventuellement les traitements ;
4. développer l'application.

Idéalement, ces phases sont censées se succéder mais dans la pratique, on est souvent obligé de revenir en arrière pour avancer correctement.

3.1 Comprendre le besoin du client

C'est le point de départ du maître d'œuvre. Dans le cas du projet FPC 2, ce besoin n'était pas exprimé sous la forme d'un cahier des charges, et il a donc fallu parcourir attentivement les comptes-rendus de réunion et les notes de service du maître d'ouvrage pour dégager le besoin du *client*. Ce besoin, bien qu'étant exprimé de manière implicite dans les documents, était relativement clair et précis : un logiciel existant, FPC, remplit la majorité des fonctionnalités attendues par le client. Cependant, ce logiciel a été développé avec des technologies abandonnées, et n'est donc plus maintenu.

Ainsi, le principal besoin (implicite) de la maîtrise d'ouvrage est de disposer d'un logiciel « maintenable ». Évidemment, quelques fonctionnalités supplémentaires sont demandées : prise en compte d'une historisation (pour pouvoir comparer les données de plusieurs enquêtes, ce qui était impossible), mise à jour de la liste des éditions demandées, possibilité d'exporter les données au format CSV pour d'autres traitements, . . . De plus, les autres fonctionnalités attendues sont le report de la saisie sur les centres, l'interface avec les applications de gestion existantes et la cohérence avec le système d'information de la DGER pour éventuellement recouper les données avec des sources externes.

3.2 Rédaction des spécifications

Lorsque les spécifications sont bien rédigées, le travail du développeur en est d'autant plus simplifié. La conception (immatérielle) de l'application se fait donc à ce moment-là. Ainsi, cette application a été divisée en deux modules (plus une fonction).

Le premier de ces modules concerne la saisie et la vérification des informations par session. Ce module se divise en trois *onglets*, permettant chacun d'afficher les données pour chaque type d'utilisateur (cf. section 7.2, page 13). Ensuite, une gestion des droits appropriée permet de limiter les opérations possibles pour un utilisateur à celle autorisées selon le tableau 7.1, page 14.

Le deuxième module permet l'édition de résultats par les SRFD et la DGER. Ce module comprend deux onglets, un pour les SRFD et un pour la DGER. Les données seront ensuite présentées à l'utilisateur sous la forme définie par la remise à jour de la liste des éditions, et les SRFD auront accès aux seules données correspondant à leur région.

Enfin, la fonction qui concerne l'importation de données de CIGALE sera accessible seulement au CERIT.

3.3 Modélisation

La méthode de modélisation utilisée au CERIT est basée sur la méthode MERISE¹, ce qui permet de générer automatiquement la base de données. Cette méthode, utilisée avec des outils performants (PowerAMC par exemple), permet, en créant seulement un Modèle Conceptuel de Données, de générer automatiquement le Modèle Physique des Données, puis les scripts de création de la base de données.

L'inconvénient que l'on peut trouver à MERISE dans le cadre du CERIT, c'est qu'il s'agit d'une méthode de modélisation qui ne permet pas de faire de la modélisation objet. Les outils de développement étant basés sur des concepts d'objets, une modélisation UML serait peut-être plus adaptée (mais plus exigeante, en termes de formation du personnel, ...).

Au CERIT, les traitements à effectuer font très rarement l'objet d'une modélisation. Les concepts donnés par la méthode MERISE pour les modéliser sont généralement beaucoup moins appliqués que pour les données. Ainsi, en restant en amont du MCT, quelques procédures ont tout de même été modélisées par des schémas et graphes de flux, de façon à mieux comprendre le fonctionnement de l'enquête.

3.4 Développement

L'outil utilisé au CERIT pour développer des applications est POWERBUILDER, environnement de développement sur lequel a été développé un *framework*, qui permet de normaliser les applications (conventions de nommage, applications développées sous la forme d'arbres et de fenêtres de propriété, ...).

Ce *framework*, ODIN, propose de nombreux objets ancêtres pour faciliter le développement ; cependant, même si la plupart des tâches sont facilitées, certaines ne sont pas évidentes, malgré la documentation et la présence de scripts exemples dans les ancêtres. Ainsi, réaliser un assistant, pour la saisie de sessions de formation, est l'une des tâches les plus hardues à réaliser en utilisant le *framework*.

La réutilisation du code a aussi posé certains problèmes. Il a fallu refaire en milieu de développement un certain nombre d'objets en utilisant les objets de référence. De même pour la base de données, qu'il a fallu recréer en utilisant des vues sur la base de référence : SCHEMAS.

¹Pour plus d'informations, cf.[7]

Chapitre 4

Rappel de la démarche

4.1 Les phases du projet

Pour réaliser ce projet de développement d'un prototype, il a au préalable été divisé en trois grandes phases :

1. analyse du besoin, rédaction des spécifications techniques détaillées ;
2. modélisation des données ;
3. développement du prototype.

Ces trois phases avaient été prévues de cette façon : environ huit semaines pour l'analyse, puis quatre ou cinq semaines pour la modélisation, et le reste pour la réalisation (soit environ dix semaines). En réalité, les prévisions pour la première phase se sont révélées assez justes. En revanche, la deuxième phase a été plus courte que prévue (deux à trois semaines) mais la modélisation a eu quelques modifications à subir après le début du développement. Pour la troisième phase, elle a donc commencé plus tôt, mais le découpage de cette macro-tâche en tâches plus fines a été trop approximatif.

4.2 Analyse des écarts

Les écarts sont dus à la méconnaissance de l'environnement de développement et à la « naïveté ». En effet, le temps prévu pour la phase d'analyse était à quelques jours près le bon. Pour la modélisation, le temps prévu était aussi assez proche du temps passé réellement, mais celui-ci a été décomposé : la modélisation elle-même et les corrections apportées en cours de développement. La modélisation a pris deux ou trois semaines seulement avant de commencer le développement, mais elle a dû être retouchée ensuite, donc le temps total était, là aussi, assez proche. Cependant, la répartition du temps prévu était assez mauvaise, sûrement par idéalisme : on finit la modélisation et on se penche ensuite sur le développement. Malheureusement, la réalité est plus mitigée : il faut parfois savoir revenir sur quelque chose que l'on croyait fini. Enfin, concernant le développement, la méconnaissance absolue du langage de développement, du *framework* et des normes utilisées au CERIT expliquent pour beaucoup les différences entre temps prévus et passés.

Chapitre 5

Suites à donner

5.1 Finalisation de l'application

Une suite évidente au projet que j'ai traité est bien sûr la réalisation d'une véritable application, et son déploiement sur tout le territoire : installation des clients lourds pour la DGER et les SRFD et communication des *login* et mots de passe pour les centres de formation, sans oublier la mise en place des serveurs nationaux (de données, *Web* et applicatif).

Pour finaliser cette application, les principales tâches à accomplir seront les suivantes :

- réalisation de toutes les éditions demandées par la maîtrise d'ouvrage ;
- dimensionner les serveurs pour une implantation (peut-on utiliser des machines existantes ? doit-on en acquérir de nouvelles ?) ;
- vérifier la sécurité du système, bien que l'application ne soit pas sensible (vérifier qu'il n'est pas possible de passer des trous de sécurité grâce à cette application pour « attaquer » d'autres systèmes plus sensibles) ;
- installation des postes clients ;
- réalisation de tests un peu plus poussés.

5.2 Prise en compte de données individuelles

Mon sujet de stage aurait pu être modifié pour devenir : « Rédaction des spécifications pour le suivi statistique individuel des stagiaires adultes ». Cependant, cette perspective a été repoussée, en particulier pour la raison suivante : la prise en compte de données individuelles n'était pas essentielle pour le service rendu, et la CNIL¹ n'aurait sans doute pas accordé l'autorisation nécessaire. De plus, beaucoup de centres de formation ne sont pas équipés de systèmes d'information leur permettant d'automatiser la saisie de chacun de leurs stagiaires, et la saisie de plusieurs centaines de stagiaires représente une tâche trop fastidieuse pour la plupart des centres.

Pour éviter des problèmes avec la CNIL, on pourrait cependant collecter des données individuelles anonymes, ce qui ne résout pas le problème de la saisie pour les centres non équipés de CIGALE.

Ainsi, un stagiaire, issu d'IUP MIAGE, sera recruté courant juin, de façon à étudier le problème en ma présence. Sa tâche sera ensuite de finaliser l'application, puis d'éventuellement étudier des possibilités pour réaliser cette remontée de données individuelles.

¹Commission Nationale Informatique et Libertés

Deuxième partie

Rapport d'études



Chapitre 6

Introduction

6.1 Présentation du Ministère de l’agriculture

Les deux directions du Ministère qui ont interagi dans le cadre de mon stage sont la DGER et la DGA. Officiellement, j’étais rattaché à la DGER, qui est la maîtrise d’ouvrage du projet FPC 2. En réalité, je travaillais pour la maîtrise d’œuvre, le CERIT, qui dépend de la DGA. Des organigrammes du Ministère sont détaillés en annexe A, page 39. Enfin, la maîtrise d’œuvre associée, le CNERTA, est basé à Dijon et réalise plus spécifiquement des applications de gestion locales (pour les centres de formation). Ses principaux produits sont CIGALE et LIBELLULE.

Au sein du CERIT, le département ECF (Examens, Concours et Formation) s’occupe particulièrement de développer des applications de gestion destinées à la DGER. Leurs réalisations principales sont : GENOME, le référentiel de données sur les formations et les centres de formation et Indexa (Indexa2 en cours de développement), logiciel de gestion des examens de l’enseignement agricole.

6.2 Le projet *Safran*

Ce projet s’occupe de réaliser des applications pour le suivi des apprenants. On le nomme aussi SPF (Suivi des Populations en Formation). Il a été initié en 1998, et son principal objet est d’organiser la circulation d’informations sur les publics en formation entre les niveaux locaux (interaction avec LIBELLULE ou CIGALE), régionaux (SRFD) et l’échelon national (la DGER). La principale difficulté consiste à remonter des données des logiciels gérés par le CNERTA, essentiellement parce que cette étape implique une étroite collaboration entre les deux entités de la maîtrise d’œuvre.

6.3 Le Travail de Fin d’Études – le projet FPC2

En deux mots, la partie du projet FPC 2 à laquelle j’ai participé consiste à réaliser un prototype d’application permettant le suivi des populations adultes en stage de formation continue dans des établissements dépendants du MAAPAR.

6.3.1 L’enquête sur la Formation Professionnelle Continue

L’enquête sur les populations stagiaires adultes du Ministère de l’agriculture se fait tous les ans. Les informations demandées aux centres de formation sont déjà agrégées (pas

d'informations individuelles). Par exemple, les centres doivent faire remonter les informations suivantes : nombre et nature des formations, nombre de participants selon le sexe, la catégorie socio-professionnelle, l'âge et l'origine géographique, ainsi que l'origine de leurs financements (organismes publics, entreprises, ...).

Tous les ans, la DGER, relayée par les SRFD, demande aux centres de formation de remplir une enquête sur la formation professionnelle continue. Les données concernant les stagiaires sont agrégées au niveau des centres de formation, puis l'enquête est remplie. Ces données servent ensuite à la DGER pour éditer un recueil des résultats statistiques de l'enquête. Les SRFD peuvent aussi utiliser leurs résultats régionaux pour faire leurs propres statistiques.

Ces résultats permettent plusieurs analyses : le financement des sessions de formation, pour pouvoir prospecter de nouvelles sources de financement, ou relancer les anciennes. Le deuxième intérêt des résultats est la répartition des stagiaires (et heures stagiaires) par région et par centre pour ensuite déterminer les centres de formation qui ont besoin de plus de postes de formateurs, ainsi que ceux qui ont (éventuellement) trop de formateurs. Enfin, on peut aussi analyser la carte des formations (répartition géographique, ...) et son évolution dans le temps.

6.3.2 Le logiciel existant

Le ministère est déjà équipé d'un logiciel depuis 1990, mais celui-ci est obsolète :

- il n'est plus maintenu ;
- les SRFD et la DGER ne se connectent pas à la même base de données, et ne voient donc pas forcément les mêmes données ;
- le développement de cette application a été effectué sous DOS, ce qui induit une ergonomie limitée par rapport aux nouvelles technologies disponibles ;
- ce logiciel n'évite pas la circulation de nombreux documents papier entre les différents services.

On peut voir ci-dessous, sur le schéma 6.1, page 10, le synoptique de la procédure actuellement utilisée avec le logiciel FPC existant.

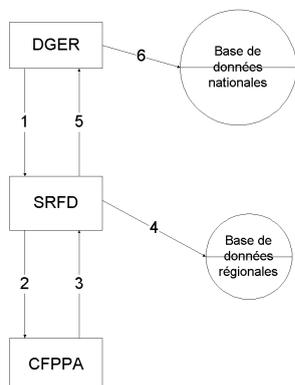


FIG. 6.1 – Synoptique de la procédure actuelle

1. La DGER envoie (chaque année) à chaque SRFD le logiciel FPC, et une structure de base de données [1].
2. Le SRFD envoie à chacun des centres de formation une enquête papier [2],
3. que celui-ci lui retourne remplie [3].
4. Le SRFD saisit les résultats de l'enquête dans sa base de données [4],
5. puis la retourne à la DGER une fois que tous les établissements ont répondu [5].
6. Enfin, la DGER rassemble les informations issues de chaque base régionale dans une base de données nationales [6], ce qui lui permet d'éditer les résultats de l'enquête.

6.3.3 La nouvelle application

La nouvelle application doit permettre de passer à une procédure comme schématisée sur la figure 6.2, page 11.

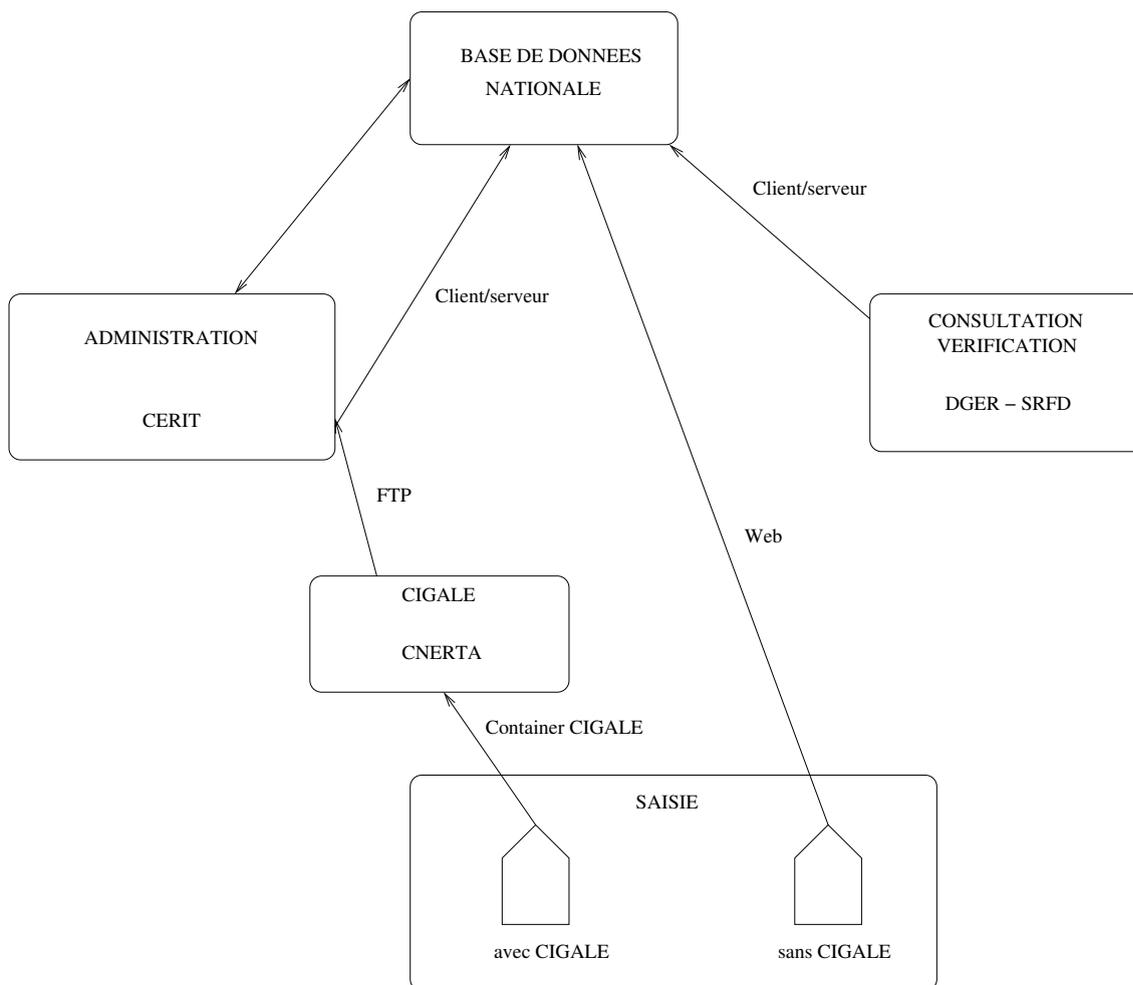


FIG. 6.2 – Synoptique de la nouvelle procédure

La première évolution que l'on peut constater est le passage à une seule base de données pour toutes les entités. Ceci permet d'éviter des incohérences si les données ne sont pas mises à jour en même temps dans toutes les bases. Ensuite, on fait une distinction entre les CFPPA équipés de CIGALE et les autres. En effet, un module de saisie *Web* permettra à tous les centres de saisir ou de vérifier des informations sur Internet, par contre, ceux équipés de CIGALE n'auront presque plus de travail en ce qui concerne l'enquête FPC.

Les services régionaux et la direction générale ont un droit de consultation des données brutes (éventuellement rectification en cas d'urgence), et peuvent consulter les résultats statistiques.

Plus précisément, l'application développée doit permettre de :

- Assurer la cohérence des données entre le référentiel GENOME de la DGER et toutes les autres applications utilisées pour gérer les nomenclatures ;
- Saisir les informations nécessaires par Internet (ou Intranet) pour les centres de formation n'étant pas équipés de CIGALE ;

- Prévoir une procédure d'intégration d'un fichier d'export de données de l'application CIGALE, pour les établissements équipés ;
- Prévoir un droit d'accès et de modification (vérification) des données par les SRFD ;
- Prévoir une consultation des données et des résultats par les membres de la DGER ;
- Éditer les résultats statistiques issus des informations saisies ;
- Pouvoir exporter les informations de la base de données au format CSV, pour pouvoir effectuer d'autres traitements.

6.4 Organisation du rapport

Analyse et spécifications Une fois l'acclimatation au projet maîtrisée (compréhension du besoin, ...), il a fallu analyser le besoin et rédiger les spécifications détaillées ; cette phase est détaillée dans le chapitre 7, page 13.

Modélisation et développement Une fois la phase d'analyse terminée, il faut modéliser la base de données et réaliser un prototype d'application. Ces deux tâches sont détaillées dans les chapitres 8, page 20 et 9, page 25.

La gestion de projet Elle était assurée en collaboration avec mon chef de projet, Louis CASTELLAN. La phase d'analyse a été un peu plus courte que prévu, cependant, nous avons souvent parlé du respect des prévisions lors du développement. En effet, certaines phases ont demandé beaucoup plus de temps que prévu, et d'autres un peu moins. Ce sujet est développé dans le chapitre 10, page 30.

Chapitre 7

Spécifications de l'application

7.1 Découpage en fonctionnalités

Pour réaliser les spécifications de l'application, celle-ci a été découpée en plusieurs *modules* :

- Saisie par Internet ;
- Importation des données de CIGALE ;
- Consultation et vérification des données ;
- Présentation des résultats ;
- Exportation des données ;
- Historisation ;
- Droits et sécurité.

Dans les sections suivantes, chacune de ces parties sera détaillée à un niveau différent selon les modules.

Le *framework* ODIN impose des normes pour les spécifications. Chaque application est développée sous forme de modules, qui affichent des données. Ces données sont présentées sous forme d'arbres ; un même module peut comporter plusieurs onglets, qui sont des façons différentes de voir les mêmes données, regroupées selon différents critères. Pour en savoir plus sur les outils de développement, reportez-vous à la section 9.1, page 25

7.2 Droits, utilisateurs et sécurité

En effet, ce module est de toute première importance, puisqu'il est prévu que chaque utilisateur (CFPPA, SRFD, DGER) se connecte avec un *login* et un mot de passe spécifique, qui permet de l'identifier. Ainsi, chaque type d'utilisateur n'aura pas le même écran de démarrage et n'aura pas accès aux mêmes fonctionnalités. On peut résumer dans le tableau 7.1, page 14 les différentes fonctions pour chaque utilisateur.

Pour les CFPPA, un seul onglet d'un seul module sera disponible : celui de saisie de session. L'arbre montrant les informations sera déjà filtré pour le centre se connectant, ainsi, lorsqu'un centre saisira une nouvelle session, elle sera toujours rattachée à son identifiant.

Les SRFD doivent pouvoir vérifier les informations rentrées par chaque centre ; ils ont donc accès à un arbre comportant les mêmes données que les CFPPA, mais celui-ci comprend un niveau de plus : le niveau *centre*.

De même, la DGER, en cas de problème sur un certain résultat, doit pouvoir vérifier les données qui y ont abouti. Ainsi, on rajoute dans l'arbre deux niveaux supplémentaires :

	Saisie	Vérification saisie	Vérification données - Validation	Consultation résultats	Exportation CIGALE	Importation CIGALE
CFPPA	■	■				
SRFD			■	Régionaux		
DGER				■		
CNERTA					■	
CERIT						■

TAB. 7.1 – Droits des utilisateurs par service

les niveaux *région* et *département*. On obtient alors la structure d'arbre ci-dessous :
 Liste des enquêtes

– Enquête du 01/01/2004

□ Midi-Pyrénées

◇ Haute-Garonne

★ CFPPA Auzeville

● Session 001

● Session 002

● ∴

★ CFPPA Auterive

★ ∴

◇ Tarn

◇ ∴

□ Nord-Pas de Calais

□ ∴

– Enquête du 01/01/2005

– ∴

Pour obtenir la structure de l'arbre pour les SRFD, il faut y enlever le niveau *région* ; pour les centres, il faut aussi oter les niveaux *département* et *centre*

De plus, la DGER aura accès au module *éditions*, qui comportera l'agencement (agrégation) de ces données selon certains critères. On peut par exemple citer la *répartition des financements par origine et par région*.

Dans un second temps, il sera avantageux de prévoir un module d'édition des résultats spécifique aux SRFD, de façon à ce qu'ils puissent disposer d'informations (en particulier sur le financement et l'activité des centres) agrégées seulement pour les centres de la région.

7.3 Spécifications techniques

7.3.1 Saisie par Internet

Tous les CFPPA n'étant pas équipés de CIGALE, il faut prévoir un module de saisie des données en mode *Web*, pour les établissements non équipés. Ce module doit aussi intégrer, pour chaque centre de formation, la vérification des données qu'il a saisies.

Le Modèle Conceptuel d'Interface résume de manière graphique l'arbre conçu pour la saisie par internet. La profondeur d'arbre a été réduite à son minimum pour éviter aux

utilisateurs d'avoir à naviguer sur trop de pages pour saisir leurs informations. Le Modèle Conceptuel d'Interface est visible sur la figure 7.1, page 15.

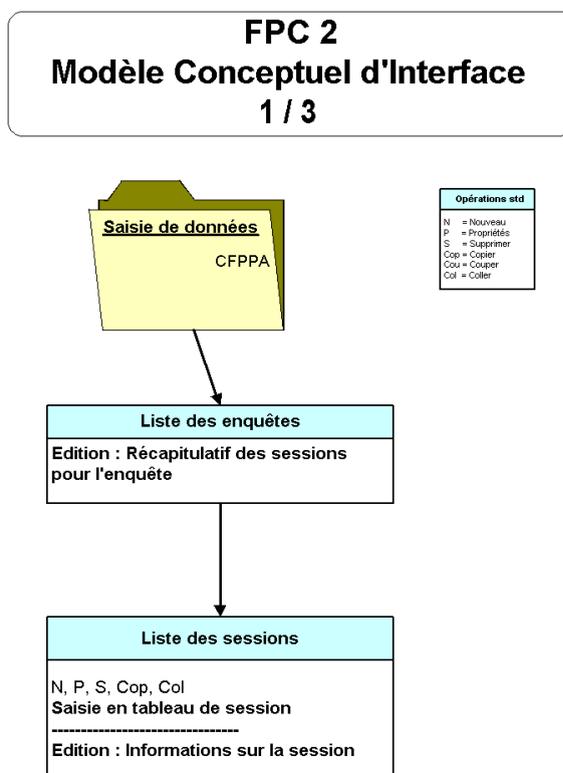


FIG. 7.1 – Spécifications techniques de l'onglet *Saisie des données*

Les spécifications de chaque objet sont ensuite renseignées dans des *fiches objet*¹. On compte deux objets principaux pour le module de saisie : *enquete* et *session*. La *fiche objet* de *session* est présentée en quasi-intégralité en annexe B, page 41, et les autres *fiches objet* sont détaillées dans le rapport de spécifications détaillées [6]. Une *fiche objet* contient les informations minimales suivantes :

1. informations pour la fenêtre *tree-view list-view* ;
2. habilitations ;
3. description graphique (simulation de captures d'écrans, ...) et description des colonnes ;
4. assistant : cinématique de changement d'onglet, ...
5. règles de gestion ;

on peut ensuite y ajouter des compléments d'information si nécessaire : filtres, éditions, opérations spécifiques, ...

7.3.2 Importation des données de CIGALE

En revanche, les établissements ayant fait l'effort d'intégrer CIGALE dans leur système d'information doivent disposer d'une fonction leur facilitant le travail. En effet, toutes les informations nécessaires à l'enquête FPC sont au préalable saisies dans CIGALE. Ainsi, il

¹La structure de la *fiche objet* est définie par le département ECF.

faut prévoir une fonction permettant d'agrèger ces données, puis de les exporter dans la base de données FPC.

Cette application étant réalisée par le CNERTA, il faut prévoir de travailler avec eux pour fixer plusieurs points :

1. le format d'échange de données ;
2. le *medium* d'échange de données.

Le format d'échange de données

Au stade des spécifications, nous avons envisagé plusieurs solutions pour cet échange, en particulier deux. La première consiste en un fichier texte, comme c'est le cas actuellement pour les données individuelles sur la formation initiale. La deuxième solution consiste à utiliser un fichier XML², qui permettrait de reproduire la structure de la base de données. Cette solution aurait l'avantage de pouvoir détecter certaines erreurs de transmission.

Après moults échanges avec le CNERTA (réunion, mails, téléphone), ils nous ont dit ne pas avoir le temps, pour l'enquête sur l'année 2003, de développer une solution de type XML. Ainsi, le transfert de données se fera par des fichiers textes.

Le *medium* de transfert

Le processus utilisé est le même que celui qui avait déjà été mis en place par le CNERTA et le CERIT pour les données concernant les élèves en formation initiale. Ainsi, on peut voir sur le schéma 7.2, page 17 un aperçu de la procédure de fonctionnement.

Les CFPPA équipés de CIGALE saisissent, tout au long de l'année, leurs données sur cette application. Le CNERTA dispose d'une fonction de transfert vers un *container* – base de données nationale collectant toutes les données des centres. Ensuite, une fonction leur permet d'agrèger les données par session, de façon à nous transmettre une archive contenant trois fichiers :

- un fichier .LIV, qui contient des informations sur les fichiers contenus dans l'archive et la génération de celle-ci ;
- un premier fichier texte (.TXT) contenant les informations sur la session elle-même : dates, formation dispensée, . . . ;
- un deuxième fichier texte contenant les informations agrégées sur les stagiaires, le financement de la session, . . .

Une fois ce fichier transféré par FTP au CERIT, une fonction de l'application FPC permettra d'intégrer ces données à la base de données FPC. On ne pourra plus alors les différencier de celles saisies par le module *Web*.

Il reste aussi à prévoir des conventions de nommage spécifiques à cette remontée, qui seront sûrement calquées sur les précédentes expériences du CERIT en la matière.

7.3.3 Consultation et vérification des données

La consultation et vérification des données est basée sur les mêmes objets non visuels que la saisie. En effet, les données sont exactement les mêmes, présentées avec un peu plus de détails (par région, département, CFPPA, cf. section 7.2, page 13).

²Pour en savoir plus sur XML, cf. [9], [8] et [10].

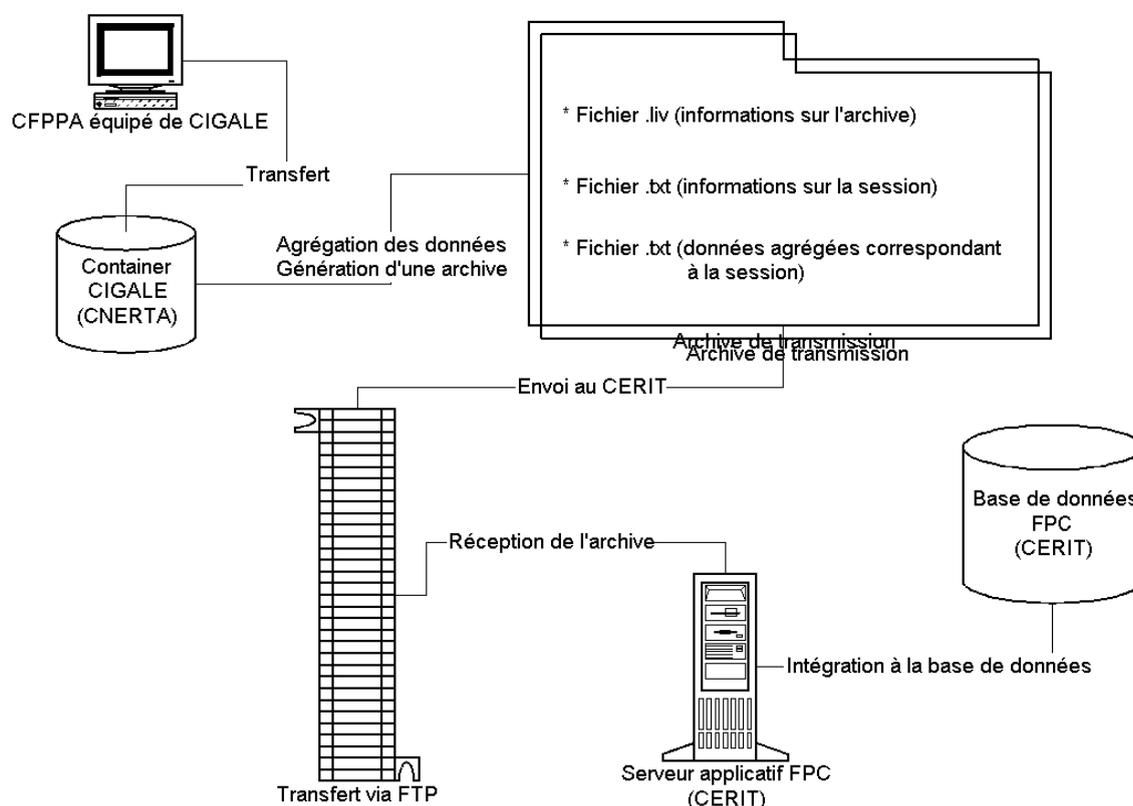


FIG. 7.2 – Processus de transfert de données depuis CIGALE

7.3.4 Présentation des résultats

Le module *Éditions* a été prévu pour ça : présenter à l'utilisateur des données exploitables de différentes façons :

- à l'écran, pour visualisation, et éventuellement impression ;
- en export, de préférence au format CSV pour pouvoir ensuite les mettre en forme dans un tableur (MS EXCEL) et les intégrer lors de la rédaction du rapport annuel sur les formations continues (« La formation professionnelle continue : résultats statistiques », cf. [2]).

7.3.5 Historisation

L'historisation a été prévue dans la modélisation des données. En effet, une des demandes de la maîtrise d'ouvrage était de pouvoir comparer les données sur plusieurs années (ou enquêtes, si plusieurs enquêtes sont effectuées dans une année). De l'autre côté, les responsables de la maîtrise d'œuvre souhaitent, en règle générale, ne pas perdre les données d'une année sur l'autre, et demandent ainsi de conserver toutes les données.

La modélisation envisagée (cf. chapitre 8, page 20) permet de faire d'une pierre deux coups : l'entité **ENQUETE** permet de faire des comparatifs entre plusieurs enquêtes et de conserver les données par session.

7.4 Choix d'une architecture

7.4.1 Mode client / serveur

Certaines spécifications dictent des choix en ce qui concerne l'architecture de l'application. Ainsi, la figure 7.3, page 18 représente une architecture possible, en mode client/serveur majoritaire.

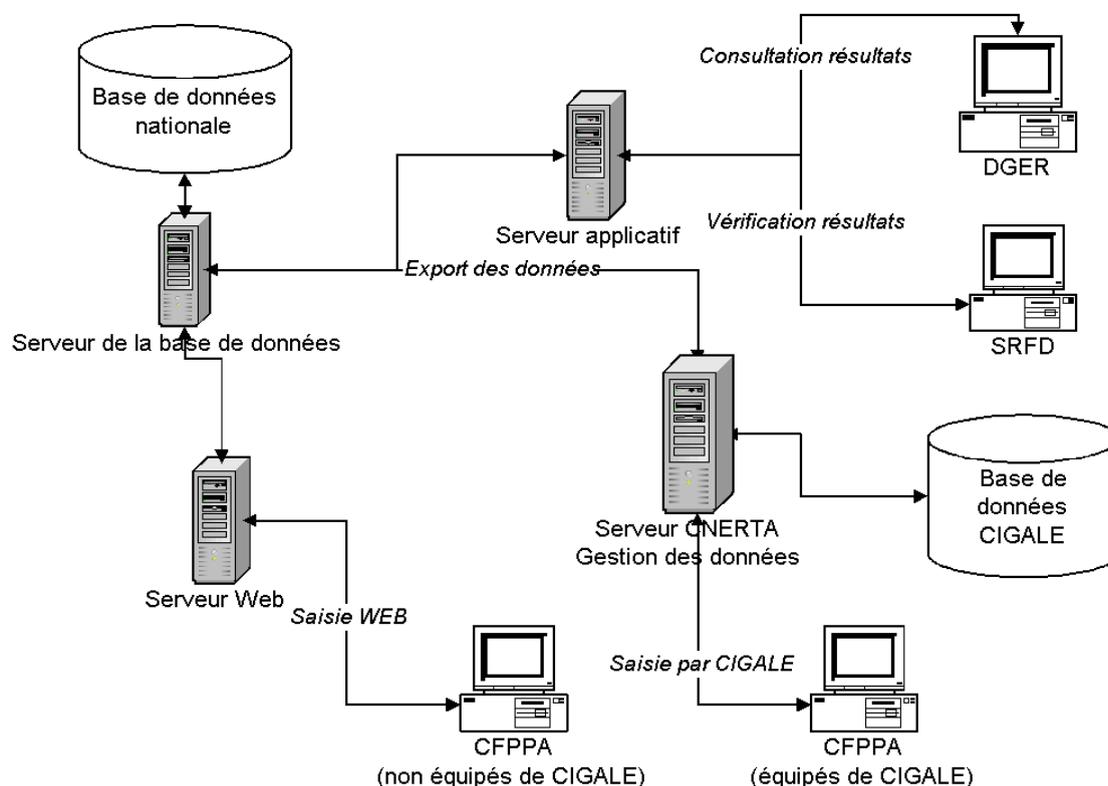


FIG. 7.3 – Architecture de l'application en mode client/serveur

Ce mode de fonctionnement présente un avantage certain au niveau des performances : sûreté de fonctionnement, rapidité, ...

7.4.2 Mode *Full Web*

Cependant, la demande de réaliser un module de saisie des données par *Internet* oblige à développer une partie (au moins) de l'application en mode *Web*. On peut donc penser à une deuxième architecture, montrée par le schéma 7.4, page 19.

Par l'accès à l'application en mode *Internet*, on obtient les avantages suivants :

- accès depuis un poste quelconque ;
- pas d'installation sur les postes clients (vérification de la compatibilité du navigateur et de la présence de la machine virtuelle JAVA) et pas de configuration matérielle particulière ;
- cohérence de l'application car il faudra obligatoirement réaliser un accès en mode Web pour la saisie des données des établissements non équipés de CIGALE.

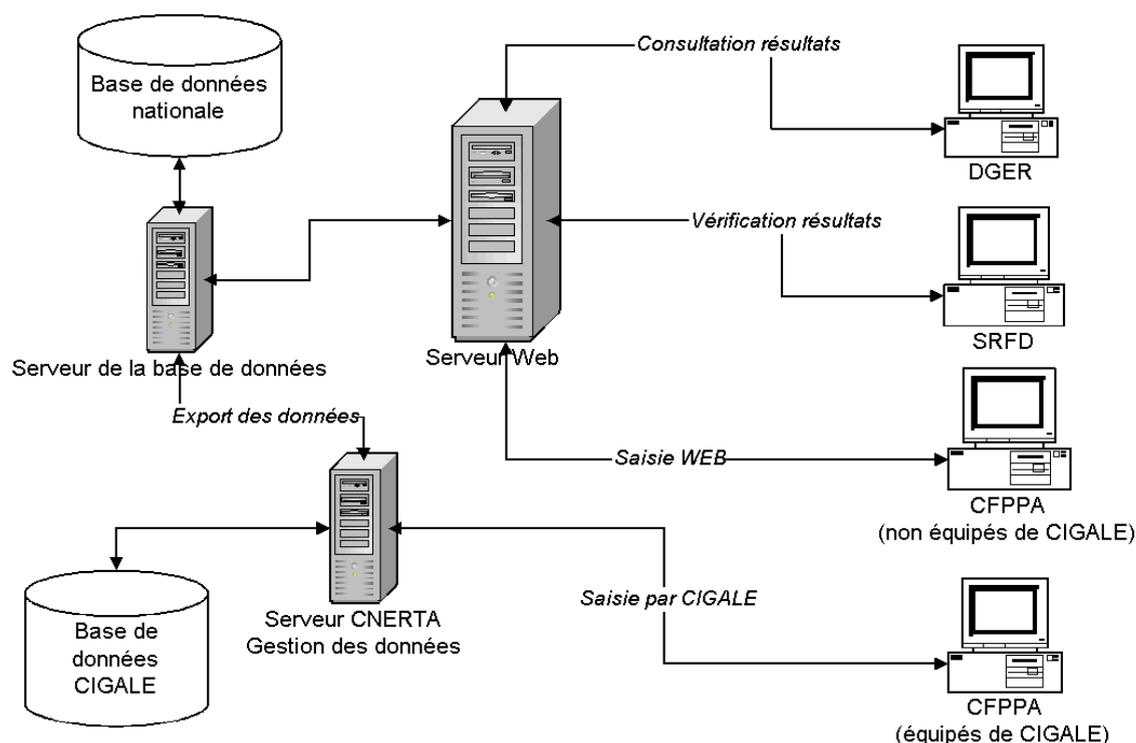


FIG. 7.4 – Architecture de l'application en mode *Total Web*

7.4.3 Éléments pour le choix

A priori, les responsables du projet au CERIT préféreraient utiliser une technique client/serveur, en particulier parce-que les développeurs maîtrisent tous cette technique, alors que le développement *Web* n'est pas totalement maîtrisé.

Cependant, étant donné qu'il faudra réaliser un module pour la saisie *Web*, et donc configurer un serveur pour l'application, avec des accès à la base de données, ... Il serait peut-être plus intéressant de développer l'application en mode *Full Web*, ce qui éviterait une installation de postes clients dans les SRFD et ne demanderait pas beaucoup plus de développements (fonction de vérification par les CFPPA et les SRFD factorisées).

Le choix final a cependant été dicté par les compétences du *terrain*, c'est à dire que l'application réalisée fonctionnera en mode client-serveur pour les SRFD et la DGER. Un dernier point pris en compte pour ce choix est la connexion réseau parfois limitée des SRFD, c'est pourquoi on a limité les échanges à ce niveau et préféré une architecture en client/serveur.

Chapitre 8

Modélisation des données et procédures

8.1 Élaboration du Modèle Conceptuel des Données

Sur la première version du MCD (cf. annexe C, page 46), on voit apparaître deux entités essentielles : SESSION et EFFECTIF STAGIAIRES. À ces entités sont rattachées toutes les informations que la maîtrise d'ouvrage souhaitait retrouver dans l'enquête. Cependant, cette modélisation pose plusieurs problèmes :

- beaucoup d'entités ne sont pas assez *souples*. Par exemple, si la liste des origines de financements s'allonge, il faut modifier la base de données, alors que ajouter un champ serait beaucoup plus simple.
- L'historisation, nécessaire pour comparer les données sur plusieurs années, n'est pas prise en compte, il faut donc rajouter la notion d'enquête.

La version numéro quatre (visible en annexe C, page 46) apporte les améliorations nécessaires : entités stockant un libellé et un code pour chacune des catégories nécessaires, et une nouvelle entité ENQUETE permettant de stocker les dates, donc de faire des comparaisons entre plusieurs années. Cependant d'autres problèmes se posent alors :

- concordance avec la base SCHEMAS du référentiel, pour certaines des entités (UAI Genome, MAPAR, Origine financement, ...);
- symbole XT^1 pour une partition (relation exclusive et totale : un et un seul des deux éléments) non pris en charge par le logiciel PowerAMC, qui pose donc certains problèmes pour la génération du MPD, puis des scripts.

On passe donc par différentes versions pour en arriver à la version huit (cf. annexe C, page 46). Les tables du référentiel sont ramenées depuis le modèle de la base SCHEMAS. De plus la relation XT est symbolisée par deux relations différentes, de cardinalité $0, 1 \rightarrow 0, n$. Cette façon de modéliser n'implique pas l'obligation d'un et un seul lien, mais elle permet de générer des scripts de création de base corrects, et l'exclusivité et la totalité seront vérifiées dans le code de l'objet métier.

Plusieurs versions ont ensuite été nécessaires pour peaufiner la modélisation, y ajouter quelques informations (nombre de stagiaires diplômables, nombre d'échecs au diplôme, ...). La version finale est la version numérotée seize (visible en annexe C, page 46). Cette dernière version permet de tout prendre en compte :

- concordance avec le référentiel (utilisation de vues);

¹À cette occasion, je remercie les participants au forum « UML, Merise, Modélisation, Méthodes, ... » sur le site <http://www.developpez.net>

- génération correcte des scripts ;
- historisation et comparaison entre années prise en compte ;
- possibilité d'allonger la liste des financements, rémunérations, ...

On peut en revanche constater la disparition de l'entité GROUPE STAGIAIRES (nom différent selon les versions). Cette entité n'était en effet pas nécessaire, car on en arrivait à une relation de cardinalité $1, 1 \rightarrow 1, 1$ entre celle-ci et l'entité SESSION.

Sur la dernière version du MCD, les entités ramenées du référentiel apparaissent juste sous la forme d'une table avec pour seul champ l'identifiant. Cette représentation permet de voir que l'on se rattache au référentiel, et permet l'exportation des clefs étrangères à la génération du MPD. La décision prise a finalement été de créer des vues sur les tables SCHEMAS pour encore améliorer la concordance. On peut reconnaître sur le schéma les tables issues du référentiel, car elles sont ombrées (sauf CONTENU et VALEUR_SEUIL).

Enfin, les noms dans le modèle ont beaucoup évolué, en partie afin de respecter certaines conventions de nommage. Les vues sur les tables du référentiel ont finalement le même nom que les tables elles-mêmes, pour faciliter la maintenance. Les noms des attributs sont composés d'un préfixe dérivant du nom de l'entité (SESSION donne FSES), du nom du champ suivi d'un suffixe correspondant au type de données :

- CDN pour un identifiant ;
- RFA pour un identifiant utilisateur, en réalité une chaîne de caractères de longueur fixe ;
- LB pour un libellé, chaîne de caractères de longueur variable ;
- DT pour un attribut de type date ou heure ;
- ON pour un booléen ;
- NB pour un nombre, entier ou réel.

8.2 Passage au Modèle Physique, puis génération des scripts

Avec l'utilisation du logiciel PowerAMC, la génération du MPD est automatique à partir du MCD. Il faut cependant préciser un fichier de configuration qui contient les informations suivantes :

- le SGBD utilisé ;
- les conventions de nommage (pour les remontées de clefs, les index, ...);
- quelques options graphiques pour le MPD.

La dernière version du MPD est disponible en annexe D, page 47. Pour la dernière version, les entités référencées ont juste leurs champs identifiants.

La génération des scripts est elle aussi automatique. Elle a posé quelques problèmes, en particulier parce que des objets (entités, relations) n'étaient plus affichés mais pas supprimés. Cela produisait des avertissements lors de la génération du MPD, mais bloquait la création des scripts. Les scripts issus de la dernière version des modèles sont présentés en annexe E, page 48.

De plus, pour l'utilisation des vues sur les tables du référentiel, il ne faut pas générer les liaisons et les tables vers ce référentiel. Ainsi, lors de la génération du script, il faut désélectionner tous les objets liés au référentiel.

Le script pour la création de vues est à part. En effet, sur les modèles, les entités sont présentes de façon à générer les clefs étrangères. Cependant, lors de la génération des scripts, on ne génère pas les tables issues de SCHEMAS. Le script de création des vues est écrit ensuite, et il est joué avant la création de la base (cf. annexe E, page 48), pour que le script de création de la base puisse faire remonter les clefs.

8.3.2 Procédure complète – de la saisie à la consultation

Le graphe des flux 8.2, page 24 reprend en simplifiant la procédure d'importation de données de CIGALE, et y ajoute la procédure globale de la nouvelle enquête FPC.

On peut ainsi dégager plusieurs étapes :

1. saisie des données ;
2. vérification de la saisie ;
3. intégration des données à la base FPC– cette étape prend place avant la vérification de la saisie lors de l'utilisation du module *Web* ;
4. vérification des données ;
5. modification des données (on boucle alors le circuit depuis la saisie, sauf si le temps presse ; si c'est le cas, les SRFD ou la DGER peuvent même modifier eux-mêmes les données) ;
6. consultation des résultats et exploitation.

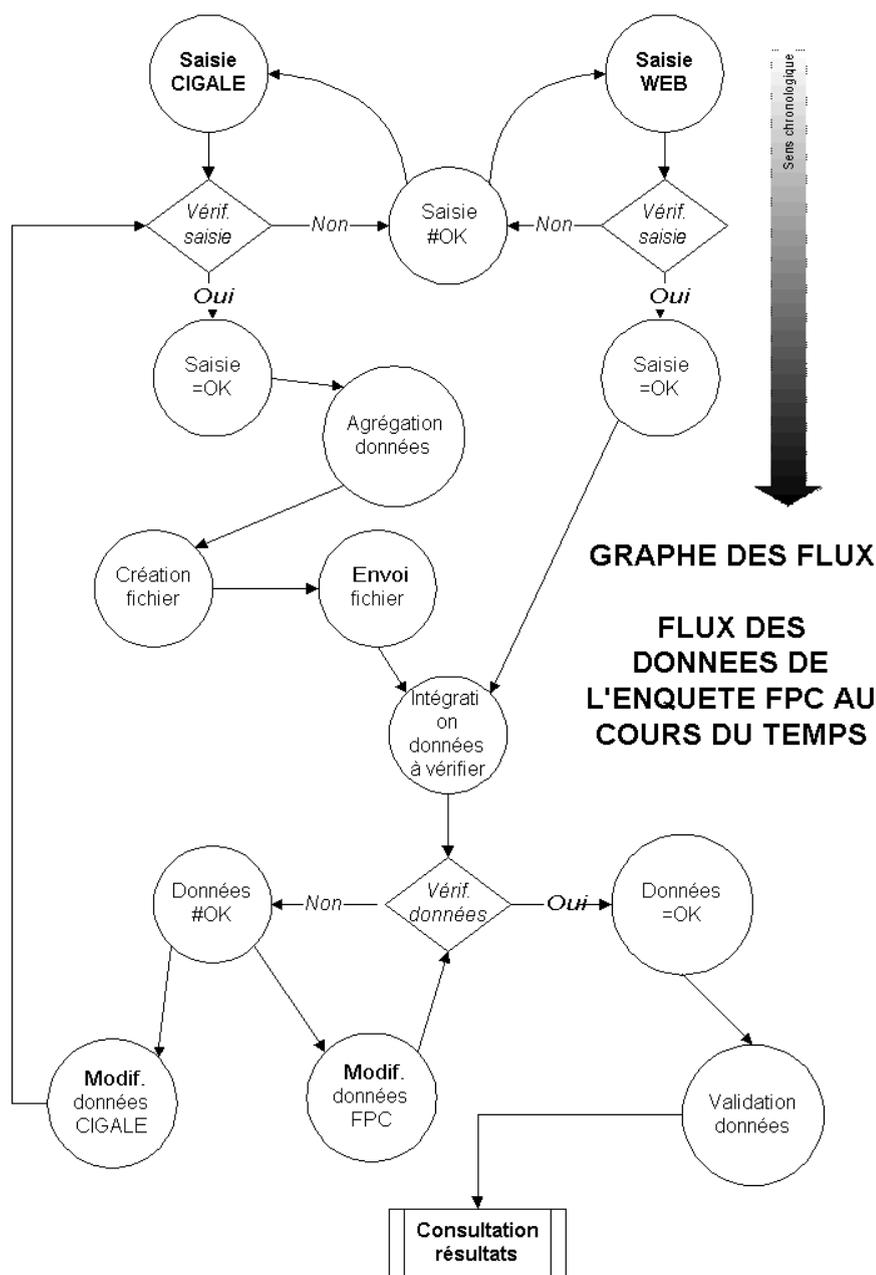


FIG. 8.2 – Graphe des flux – enquête FPC

Chapitre 9

Développement

9.1 Présentation de l'environnement de développement

9.1.1 PowerBuilder

POWERBUILDER est un langage de quatrième génération, orienté objet. Il permet de développer des applications de gestion pour un environnement MS WINDOWS. L'organisation se fait par bibliothèques, qui contiennent différents objets. La liste ci-dessous présente les principaux objets gérés par POWERBUILDER.

Bibliothèque : une bibliothèque est un ensemble de fichiers sauvegardés sur un serveur ou sur le poste local, il est possible de spécifier un chemin d'accès vers plusieurs bibliothèques (.PBL) ;

Objet : un objet permet de communiquer avec l'utilisateur (fenêtre, menu, ...). on peut en particulier citer la *Datawindow*, qui permet d'exécuter une requête SQL et d'en montrer le résultat à l'utilisateur ;

Évènements : l'évènement est provoqué par l'utilisateur, par une commande PowerScript, ou par le système, on peut citer en exemple les évènements *Clicked()*, *Open()*, *Close()*, ... ;

Scripts : codification des traitements par le langage PowerScript ; en particulier, un script peut contenir les actions à effectuer lorsqu'un évènement se produit ;

Contrôles : dans une fenêtre (objet graphique), il est possible de placer des objets servant à exécuter différentes actions, en particulier un *DataWindow Control*, qui sert à exécuter les instructions contenues dans une *DataWindow* (requêtes SQL) et à en afficher le résultat dans la fenêtre graphique.

9.1.2 ODIN, *framework* du Ministère de l'Agriculture

Avec POWERBUILDER, les services *Examens, Concours et Formation* et la *Protection des Végétaux*¹ ont développé un *framework*, nommé ODIN. Ainsi, cela leur a permis de normer en particulier l'aspect graphique des applications, pour que les utilisateurs ne soient pas « perdus » en passant de l'une à l'autre.

Ainsi, les informations sont présentées sous forme d'arbres (comme l'explorateur de WINDOWS) et chaque niveau peut se développer en ses *files*. Par exemple, pour l'application FPC, le niveau « session » est un fils du niveau « enquête ». En double-cliquant sur un

¹On retrouve la concaténation abrégée des initiales de ces deux services dans le préfixe *epv*, qui est utilisé pour de nombreux ancêtres avec ODIN.

niveau, on accède à sa fenêtre de propriétés ; en revanche, certaines fonctions remontent sur le père. Par exemple, pour créer une nouvelle session, on clique avec le bouton droit sur une enquête, puis sur Nouveau.

L'utilisation du *framework* s'accompagne d'autres normes, en particulier pour les noms d'objets, ce qui facilite la maintenance des applications².

L'interface ODIN nécessite de saisir plusieurs choses pour une application : une transaction (connection à une base de données), des objets (métiers, avec fenêtre de *tree-view*, propriétés, ...), puis des fenêtres possédant des onglets présentant les objets par niveau.

En résumé, ODIN apporte les avantages suivants : interface graphique normée et claire, maintenabilité des applications et adaptabilité des développeurs grâce aux différentes règles de nommage et de codage.

9.1.3 Le partage de code

Le partage de code n'est pas une norme du CERIT. Tout le monde ou presque l'utilise, mais les implémentations sont différentes selon les départements. Au département ECF, le partage de code est découpé en deux niveaux : le partage des données (de la base de référence) et le partage des objets dans l'application.

Le partage de données

Généralement, dans le service, on utilise des vues pour se connecter au référentiel. Ainsi, les données sont toujours à jour, et cela évite de vérifier ou d'assurer la concordance des données. Une vue est en effet dynamique ; elle correspond à un ordre `SELECT` et renvoie toujours les dernières données du référentiel. Cependant, lorsque la structure de la base référencée change (modification des tables, ...), il faut maintenir les vues :

- en cas d'ajout d'un champ, la vue continuera à renvoyer les bonnes valeurs, à l'exception du nouveau champ (il faut cependant faire attention au sens du nouveau champ par rapports aux champs existants) ;
- en cas de suppression ou de modification d'un champ, il faut impérativement rejouer les scripts de création des vues, car les ordres SQL ne pourront pas être correctement exécutés.

Le partage d'objets

Dans PowerBuilder L'application permettant de gérer le référentiel s'appelle GENOME. L'équipe chargée de gérer les mises à jour a développé des bibliothèques pour POWERBUILDER, à intégrer aux applications clientes, pour bénéficier du concept d'objet et réutiliser le code des objets métier qui sont déjà présents dans GENOME (données du référentiel). Ainsi, l'application FPC doit permettre, pour la DGER, d'afficher les sessions pour une enquête par centre de formation. L'arborescence permettant d'afficher les centres (avec région et département) a été récupérée dans les bibliothèques de GENOME. Il n'y a donc pas nécessité de recréer certains objets, déjà utilisés dans d'autres applications.

Dans la base d'administration (ODIN) Il est possible de récupérer des niveaux dans des fenêtres pointant directement sur des objets de l'application GENOME. Ainsi, un objet existant dans GENOME peut directement être partagé par toutes les applications du service. Il est aussi possible de ne récupérer que des parties de fenêtres en héritant l'objet

²Pour plus d'informations, se reporter aux conventions de notations, en annexe F, page 49.

de GENOME puis en le surchargeant, ou en créant un niveau fictif pointant vers un niveau de GENOME. Il est aussi possible de récupérer des éditions, des filtres, ...

9.2 Avantages et inconvénients de ces outils

9.2.1 PowerBuilder

POWERBUILDER est utilisé uniquement pour développer des applications WINDOWS. De ce fait, des avantages et des inconvénients en découlent :

- l'adaptation à un seul système d'exploitation facilite la gestion des objets, qui sont conçus seulement pour ce système ;
- l'environnement de travail est facile à utiliser, en particulier pour hériter, construire des *datawindows*, tout ce qui est standard se fait graphiquement, et on peut souvent travailler en *wysiwyg*³, ou presque ;
- cependant, l'environnement de développement est assez gourmand en ressources, et même une bonne machine connaît parfois quelques lenteurs ;
- enfin, le fait de développer pour un seul système d'exploitation est aussi un inconvénient : les utilisateurs doivent être équipés de WINDOWS (95, 98, NT, 2000 ou XP).

9.2.2 ODIN

Ce *framework* est assez problématique au sein du CERIT. En effet, les personnes ayant développé sous POWERBUILDER natif le trouvent très lourd à manipuler, d'autant plus qu'il interdit certaines opérations très faciles à mettre en place avec POWERBUILDER⁴.

Cependant, même s'il est lourd à manipuler, il apporte de nombreux avantages, en particulier au niveau de l'interface. En effet, celle-ci étant normée sous forme d'arbres, la gestion de ces arbres est quasiment automatique. Il compte beaucoup d'autres avantages, en particulier la mise en place d'un bouton de rapatriement (). Ce bouton est l'équivalent du bouton WINDOWS et permet de ramener une information à partir d'une vue en arbre, et donc en parcourant différents niveaux de détails.

En ce qui concerne l'application FPC, les normes de développement imposées par ODIN ont posé certains problèmes. Pour le module « Saisie et vérifications de sessions », l'adaptation à la présentation en arbres puis en feuilles de propriétés a été simple, d'autant plus qu'elle permettait d'introduire plusieurs niveaux de détails. En effet, les SRFD et la DGER doivent avoir des niveaux supplémentaires pour voir les départements, les établissements, et éventuellement les régions. Le module « Éditions » a été plus problématique, car il faut introduire des niveaux fictifs dans les arbres pour faire des groupements d'éditions ; cependant, les spécifications de ce module ne sont pas encore très détaillées.

9.2.3 Le partage de code

Le partage de code a été, lors du développement de FPC, plus une contrainte qu'un avantage. L'idée de profiter des objets déjà créés pour sa propre application est en effet logique à partir du moment où l'on utilise des technologies objet. Cependant, la plupart des objets utilisés dans FPC n'étaient pas créés dans l'application du référentiel, mais gèrent

³ *What You See Is What You Get* : ce que vous avez à l'écran est ce que vous obtiendrez

⁴ Le chef du département ECF, Pierre LAMOTHE, aime à comparer les différences entre POWERBUILDER et ODIN à celles entre le tennis et le squash. En effet, le terrain est plus « encadré » mais les solutions possibles en sont d'autant plus riches ...

des objets du référentiel. Ainsi, il a fallu utiliser les conventions de normage de GENOME pour créer beaucoup d'objets (rapatriement de valeurs ou listes déroulantes en particulier).

Ceci n'est tout de même pas une perte de temps, car d'autres futures applications utiliseront peut-être les objets créés pour FPC, et gagneront ainsi beaucoup plus de temps que je n'en ai perdu.

Enfin, un dernier petit problème avec la récupération d'objets de GENOME est le manque de documentation, qui induit une perte de temps. En effet, lorsque l'on cherche par exemple un arbre de rapatriement qui est censé être géré par GENOME, on ne sait pas

1. s'il existe ;
2. si oui, où le chercher (bien que l'on puisse limiter intuitivement le nombre de bibliothèques pouvant contenir l'objet cherché).

Ainsi, disposer d'un document (ou mieux, d'une base de données du type base de données bibliothécaire) serait un atout non négligeable pour trouver rapidement les objets qu'on cherche.

9.3 Les principales difficultés rencontrées

Les extraits de code dont il est fait mention ici se trouvent dans l'annexe G, page 50.

9.3.1 Réalisation d'un assistant

La réalisation d'un assistant pour la saisie de sessions dans l'application a été l'une des tâches les plus hardues. Ceci pour plusieurs raisons :

- la documentation d'ODIN n'est pas bien conçue ;
- le paramétrage fait intervenir beaucoup d'objets et d'évènements.

De fait, ces deux raisons se rejoignent, parce que c'est aussi en raison du grand nombre d'objets, d'évènements et de scripts intervenants que la documentation est mal structurée. En effet, un assistant est basé sur une fenêtre de propriétés comportant des onglets, et la documentation fait souvent référence à cet objet. Ainsi, pour tenter de comprendre au préalable le fonctionnement de cet objet, il faut constamment faire des allers-retours entre plusieurs pages de la documentation. De plus, cette documentation comporte beaucoup de références aux scripts exemples sur les objets ancêtres, mais ceux-ci ne sont pas très explicites ; on ne sait parfois pas à quel objet correspond une méthode.

Enfin, le nombre d'évènements à coder est particulièrement élevé (en comparaison avec un objet « classique ») ; de plus, le code se partage entre l'objet propriété relatif à l'assistant, et tous les objets métiers gérant chacun des onglets.

Initialisation de l'affichage

Ce script permet d'initialiser l'affichage de la fenêtre de propriétés. Chaque ligne définit un onglet, avec le code de l'onglet, son titre affiché, la *datawindow* associée et le mode de saisie des données (une seule instance dans la table en monoligne – cf. G.1, page 50). Il fait appel à la fonction `uf_initdefDwVisuelle`, qui est attachée au même objet `fpc_n_cst_propriete_session`.

Pour les onglets qui contiennent un tableau de saisie (on peut alors y saisir plusieurs lignes), il faut préciser le mode `MULTILIGNE` dans l'évènement `uf_initdef1DwVisuelle`.

Les problèmes se posent à partir de cet évènement. En effet, si tous les onglets ont le même *resultset* (sont issus de la même requête), l'initialisation se fait correctement.

Cependant, les *datawindows* définies ici ne sont pas basées sur les mêmes tables, il faut donc redéfinir l'initialisation de l'affichage, grâce à l'évènement `ue_corrigeDefdwvisuelle`.

L'évènement de correction d'affichage

Pour chaque onglet (excepté les deux premiers), on spécifie les caractéristiques du père (la session en l'occurrence). Les deux premiers onglets sont particuliers parce qu'ils sont déjà basés sur la *datawindow* déclarée dans ODIN. En revanche, lorsqu'un onglet n'est plus basé sur cet objet (`fpc_d_prop_session`), il faut corriger son affichage comme montré dans l'annexe G.2, page 50 pour l'onglet *AUTFORM*.

L'enchaînement des onglets

Cet enchaînement ne doit pas être linéaire. En effet, lorsque l'on est sur le premier onglet, on doit saisir le type de formation relatif à la session. Selon cette donnée, l'onglet suivant sera *MAAPAR* (en cas de formation référencée) ou *AUTFORM* (autre type de formation).

Le même système de codage permet, dans l'évènement `ue_preprecedant`, de « sauter » un onglet lorsque l'utilisateur clique sur le bouton Précédent.

9.3.2 Rapatriement

ODIN a fait en sorte d'implémenter une fonctionnalité très utile : le rapatriement. Ceci permet, à partir d'une fenêtre (en particulier de propriétés), d'ouvrir une fenêtre *tree-view list-view* pour choisir un élément parmi une liste. Cette fonction permet d'éviter les menus déroulants trop longs (ou incompréhensibles). Ainsi, on peut choisir parmi une liste de libellés, et c'est l'identifiant de la table qui est pris en compte par l'application. De plus, cette fonctionnalité permet d'utiliser la présentation en arbres avec différents niveaux de détail.

Pour rapatrier des éléments, il faut coder deux évènements (l'exemple de l'annexe G.4, page 51 est celui de la lettre de spécialité – nomenclature interministérielle – lors de la saisie d'une session dont la formation n'est pas référencée) : `ue_prerapatriegetinfo` dans l'objet `fpc_n_cst_propriete_session` et `ue_retourmodal` dans l'objet métier gérant le niveau à rapatrier.

Le premier, `ue_prerapatriegetinfo`, sert à appeler la fenêtre de *tree-view list-view* dans laquelle on veut aller chercher des informations. Le second évènement (`ue_retourmodal`), dans l'objet métier, permet de récupérer le libellé et le code de la lettre de spécialité, d'afficher le libellé dans l'onglet, et de conserver le code dans la *datastore*, pour ensuite mettre à jour les tables.

Chapitre 10

La gestion de projet

10.1 Découpage du projet en tâches

L'objectif du projet est la réalisation d'un prototype de l'application. Pour parvenir à cet objectif, le projet a été divisé en tâches, réparties sur toute la durée du stage. Au premier niveau, on retrouve des macro-tâches, qui permettent de créer ensuite un découpage plus précis. Ces regroupements sont les suivants :

Immersion dans le projet Cette tâche regroupe toute la prise en main du projet : prendre connaissance des différents acteurs, maîtriser le vocabulaire, ...

Analyse L'analyse comprend toutes les tâches permettant de mener à la rédaction des spécifications techniques détaillées, ainsi que la présentation de ces spécifications au client.

Formation Cette phase regroupe les formations nécessaires au développement : POWER-BUILDER, ODIN, ainsi que d'autres logiciels utilisés pour la modélisation.

Modélisation Utilisation de la méthode MERISE pour modéliser les données, et schématisation de la procédure de travail à partir des connaissances acquises pendant les phases d'*Immersion dans le projet* et d'*Analyse*.

Réalisation et développement Réalisation du prototype à proprement parler selon les spécifications, en utilisant les données de la base modélisée.

Corrections finales Le stage se finissant après la présentation du prototype à la maîtrise d'ouvrage (la DGER), cette phase permettra d'apporter des corrections au prototype, selon les remarques des utilisateurs.

10.2 GANTT prévisionnels et mises à jour

Les mises à jour des diagrammes ont parfois fait apparaître de nouvelles tâches. Cependant, les macros-tâches listées ci-dessus n'ont pas été modifiées, seulement leur découpage en tâche plus fines.

10.2.1 Prévisions de la première semaine

On peut voir sur le tableau 10.1, page 31 les premières prévisions effectuées pour le projet FPC.

Les diagrammes de GANTT sont disponibles en annexe H, page 53.

N°	Nom de la tâche	Durée	Début	Fin
1	Immersion dans le projet	5 jours	10/02/03	17/02/03
2	Prise en main du projet	3 jours	10/02/03	13/02/03
3	Catalogue éditions existant	1 jour	13/02/03	14/02/03
4	Comparatif données	1 jour	14/02/03	17/02/03
5	Analyse	27 jours	17/02/03	25/03/03
6	Rédaction spécifications	15 jours	17/02/03	10/03/03
7	Identification architecture	8 jours	14/03/03	25/03/03
8	Formation	9 jours	03/03/03	13/03/03
9	Sybase	4 jours	03/03/03	06/03/03
10	PowerBuilder	4 jours	10/03/03	13/03/03
11	Modélisation	15 jours	26/03/03	17/04/03
12	Modélisation données	5 jours	26/03/03	01/04/03
13	Modélisation procédures	10 jours	02/04/03	17/04/03
14	Réalisation & développement	63 jours	18/04/03	25/07/03
15	Création base de données	2 jours	18/04/03	21/04/03
16	Création interface de saisie	15 jours	22/04/03	05/05/03
17	Création d'éditions	5 jours	06/05/03	12/05/03
18	Procédure d'export - Cigale	5 jours	13/05/03	19/05/03
19	Interface accès et modification	13 jours	20/05/03	02/06/03
20	Assemblage composants	10 jours	03/06/03	01/07/03
21	Exportation format CSV	4 jours	02/07/03	07/07/03
22	Tests - corrections	9 jours	15/07/03	25/07/03

TAB. 10.1 – Tableau des tâches initial (au 17 février 2003)

10.2.2 Diagramme remis à jour en milieu de stage

Un peu avant le milieu du stage (début avril, après la présentation des spécifications à la DGER), le planning a été remis à jour. On peut voir sur le diagramme de GANTT H.2, page 54 les nouvelles prévisions. On peut constater quelques changements sur le diagramme (enchaînement des tâches) mais aussi sur les tâches en elles-mêmes.

Évolutions des tâches Premièrement, dans la phase d'analyse, un jalon a été rajouté : la présentation des spécifications à la DGER, à Paris. Cette réunion a permis de voir que les besoins du client n'avaient pas radicalement changé, et donc de vérifier l'adéquation de notre démarche avec leurs besoins. Au niveau de la formation, la formation prévue (SGBD SYBASE) a été remplacée par la formation au *framework* ODIN.

Pour la modélisation : réévaluation des temps pour la modélisation des données et des procédures (inversion du rapport), due en particulier au fait que les MCT prévus par la méthode MERISE ne sont pas utilisés au CERIT.

Au niveau du développement, les tâches ont été plus détaillées et réagencées. En effet, certains détails n'avaient pas été pris en compte pour les premières prévisions, comme par exemple la demande d'espace réservé sur le serveur de développement, la présentation du prototype à la DGER, . . . De plus, certaines tâches pouvaient être menées en parallèle, la planification a donc été revue en ce sens (plus de linéarité dans l'enchaînement des tâches).

L'enchaînement En effet, même si j'étais quasiment la seule ressource affectée au projet, je pouvais mener plusieurs tâches en parallèle. Ainsi, cela permet de gagner du temps s'il faut, par exemple, sur une tâche, attendre la réponse d'un tiers non disponible pour avancer. Ainsi, l'enchaînement quasiment linéaire du premier diagramme est bouleversé pour prendre en compte ces changements.

10.2.3 Prévisionnel pour la fin du stage

Le tableau 10.2, page 33 montre les tâches effectuées et restant à faire début juin. Certaines étaient déjà prévues au début du stage, d'autres se sont greffées lors de l'affinage des prévisions début avril, d'autres encore ont été ajoutées début juin. On peut voir sur le diagramme de GANTT H.3, page 54 Les modifications apportées sur le planning.

On peut constater un allongement démesuré des tâches autour de la fin juin, ceci parce que la période de remise des diplômes est prise en compte dans le calendrier.

N°	Nom de la tâche	Durée	Début	% achevé ¹	Durée au 17.02
1	Immersion dans le projet	5 jours	Lun 10/02/03	100 %	5 jours
2	Prise en main du projet	3 jours	Lun 10/02/03		
3	Catalogue éditions existant	1 jour	Jeu 13/02/03		
4	Comparatif données	1 jour	Ven 14/02/03		
5	Analyse	30 jours	Lun 17/02/03	100 %	27 jours
6	Rédaction spécifications	30 jours	Lun 17/02/03		15 jours
7	Identification architecture	8 jours	Ven 14/03/03		8 jours
8	Formation	6 jours	Lun 03/03/03	100 %	9 jours
9	POWERBUILDER	3 jours	Mar 11/03/03		4 jours
10	ODIN	3 jours	Mar 25/03/03		4 jours
11	Modélisation	31 jours	Mer 17/02/03	100 %	15 jours
12	Modélisation données	26 jours	Lun 24/02/03		5 jours
13	Modélisation procédures	10 jours	Lun 17/02/03		10 jours
14	Réalisation & développement	76 jours	Mar 01/04/03	45 %	63 jours
15	Création base de données	4 jours	Mar 01/04/03	100 %	2 jours
16	Création interface d'accès (saisie et modification)	50 jours	Lun 07/04/03	80 %	28 jours
17	Passage mode <i>Web</i>	13 jours	Lun 30/06/03	0 %	
18	Procédure d'import - Cigale	45 jours	Lun 21/04/03	40 %	5 jours
19	Création d'éditions	20 jours	Mar 10/06/03	0 %	5 jours
21	Tests - corrections	9 jours	Jeu 17/07/03	5 %	9 jours
22	Corrections	5 jours	Lun 04/08/03	0 %	

TAB. 10.2 – Tableau des tâches final (au 02 juin 2003)

10.3 Analyse des écarts

Il est normal d'avoir des écarts entre les différents plannings prévisionnels pour le stage. En effet, le premier jet ne peut pas être le bon, puisqu'on ne connaît alors pas très bien le sujet. Ainsi, les prévisions s'affinent de jour en jour. Il faut alors savoir poser ces prévisions pour bénéficier de l'expérience acquise, non seulement en les faisant, mais en analysant les écarts.

¹Estimations plus ou moins grossières.

Phase d'analyse

À quelques jours près (5 jours d'écart entre les prévisions au début du stage et les heures effectivement réalisées), cette tâche avait été bien estimée. Cependant, les quelques jours d'écart entre prévisions et réalité pourraient peut-être s'expliquer par le fait que l'on m'a demandé de refaire certaines parties des spécifications selon les normes en vigueur dans le département ; normes dont je ne connaissais pas l'existence en commençant.

Modélisation

La modélisation a été beaucoup plus longue que prévu, en particulier parce qu'elle s'est étalée dans le temps. En effet, les 35 jours prévus n'ont pas été employés à temps plein pour cela. On peut donc estimer le temps réel passé entre quinze et vingt jours, ce qui correspond aux prévisions en début de stage.

En fait, les tâches d'analyse et de modélisation ont été menées en parallèle, et ne correspondent pas exactement aux dates montrées sur les diagrammes et les tableaux. Cependant, ceux-ci sont une estimation du temps pris par ces tâches au total.

Développement

On peut voir sur l'évolution des temps pour le développement qu'effectuer ce genre de prévisions sans connaître l'environnement de développement est un échec. En effet, les prévisions au début du stage n'ont rien à voir avec celles de début juin (au niveau de l'enchaînement, des durées et même des tâches).

En effet, on peut trouver heureux que les phases avant le développement aient pris moins de temps car menées en parallèle, car le développement en a demandé beaucoup plus. Ceci pour plusieurs raisons :

- trois jours de formation pour POWERBUILDER, puis trois jours pour ODIN, c'est relativement peu pour arriver à maîtriser les outils ;
- plusieurs développements ne sont pas appris pendant la formation : création d'un assistant, passage en mode *Web*, ... J'ai donc dû prendre beaucoup de temps pour m'auto-former à ces techniques ;
- la tâche « assemblage des composants » a disparu car tous les objets viennent s'ajouter les uns aux autres, il n'est donc pas nécessaire de les assembler ensuite ;
- pour causes de restrictions budgétaires au MAAPAR, nous n'avons pas pu rencontrer le CNERTA et avons donc travaillé à distance, ce qui demande plus de temps, en particulier pour la rédaction et la lecture de documents.

On peut aussi attribuer cet échec en partie au fait que les retours sur l'analyse pendant le développement n'avaient pas du tout été prévus.

Chapitre 11

Conclusion

11.1 Avancement du projet

L'avancement du projet suit à peu près le cours qui avait été prévu au début du stage, c'est-à-dire que le prototype devrait être prêt pour une démonstration à la DGER fin juillet. Cependant, quelques déceptions sont au rendez-vous : en particulier, j'espérais pouvoir finir le développement des éditions avant cette présentation, et ce ne sera pas possible. Seulement quelques éditions seront disponibles pour l'exemple.

En revanche, mon tuteur et mon chef de projet ont décidé de ne pas abandonner le travail qui est en cours sur l'enquête FPC. Ainsi, un stagiaire issu d'IUP MIAGE a été recruté pour finir le prototype avant la fin de l'année et ainsi pouvoir définir une région test avant une utilisation nationale pour l'enquête sur l'année 2004.

Ainsi, je suis plutôt satisfait que mon travail n'ait pas servi à rien et soit poussé plus avant de façon à être utilisé.

11.2 Mes objectifs pour le stage

Mon objectif principal pour le stage était de participer à un projet ressemblant à celui que nous avons mené au sein de l'option GSI au mois de janvier : rédaction de spécifications, modélisation, puis réalisation d'une application de gestion. Cet objectif a été entièrement rempli, car c'est exactement la tâche qui m'a été confiée pour ce TFE.

Cependant, un petit regret vient obscurcir le tableau, celui de ne pas avoir été intégré à une équipe. Je n'ai donc pas pu observer tous les avantages et les inconvénients liés au partage de code et au travail au sein d'une équipe de développement travaillant sur le même logiciel.

Enfin, l'objectif principal a pour moi été rempli, et même si je pensais pouvoir plus avancer le développement du prototype, ce stage reste une bonne expérience sur tous les points de vue : analyse, modélisation, développement et planification. En effet, il m'a permis de découvrir de nouveaux outils (POWERBUILDER, ODIN et normes du CERIT) et d'acquérir une expérience sur d'autres (modélisation des données en MERISE, planification, ...).

Glossaire

- CERIT** : Centre d'Études et de Réalisations Informatiques de Toulouse i, 2, 3, 5, 6, 9, 14, 16, 17, 20, 23, 27, 28, 33, 36, 38, 40, 41
- CFPPA** : Centre de Formation Professionnelle et de Promotion Agricole, par extension, tous les centres dispensant une formation rentrant dans le cadre de l'enquête 11, 13–15, 17, 20, 23, 43
- CIGALE** : Application de gestion locale des centres de formation professionnelle 5, 9, 11, 13–18, 23, 24, 38, 43
- CNERTA** : Centre National d'Études et de Ressources en Technologies Avancées, situé à Dijon 9, 14, 16, 17, 23, 35, 38
- DGA** : Direction Générale de l'Administration 9
- DGER** : Direction Générale de l'Enseignement et de la Recherche ii, 2–5, 7, 9–11, 13–15, 20, 23, 24, 27, 28, 31, 33, 36, 38, 43
- EN** : Éducation Nationale — formation ou autre issue d'un référentiel du Ministère de l'Éducation Nationale 44
- FPC** : Formation Professionnelle Continue — par extension, nom du projet consistant à réaliser l'application de gestion des enquêtes ii, 4, 9, 11, 16, 17, 24–29, 31, 36, 49
- GENOME** : Application de gestion des nomenclatures dans le Ministère ii, 9, 11, 27–29
- LIBELLULE** : Application de gestion locale des centres de formation initiale (lycées agricoles, ...) 9
- MAAPAR** : Ministère de l'Agriculture, de l'Alimentation, de la Pêche et des Affaires Rurales ii, 9, 35, 38, 40, 41, 44
- MCD** : Modèle Conceptuel des Données, réalisé d'après la méthode MERISE 21, 22, 47–49
- MCT** : Modèle Conceptuel des Traitements, d'après la méthode MERISE 5, 33
- MPD** : Modèle Physique des Données, réalisé d'après la méthode MERISE 21, 22, 47–49
- ODIN** : Framework utilisé pour le développement d'applications Windows ; outil de développement normalisé d'interface i, 5, 13, 26–31, 33–36, 38
- PowerBuilder** : Langage de quatrième génération, comportant des objets de haut niveau, utilisé pour développer des applications Windows i, 5, 26–28, 31, 34–36
- SRFD** : Service Régional de la Formation et du Développement 2, 5, 7, 9–11, 13–15, 20, 23, 24, 28, 43

Bibliographie

Références bibliographiques

- [1] CNERTA. Cahier des charges de l'application CIGALE. Document interne au MAAPAR. Pour l'interaction avec CIGALE.
- [2] MAAPAR DGER. La formation professionnelle continue – résultats statistiques. Pour la compréhension du besoin, 2001.
- [3] Cellule ODIN. Documentation utilisateur ODIN. Document interne au CERIT. Pour l'aide au développement.
- [4] SYBASE, Inc. *PowerBuilder — Getting started*, 1999. Formation de base à PowerBuilder.
- [5] Thomas VAN OUDENHOVE. Catalogue des éditions du logiciel existant (fpc). Document interne au CERIT. Pour la compréhension du besoin.
- [6] Thomas VAN OUDENHOVE. Rapport de spécifications détaillées de l'application FPC. Document interne au CERIT.

Références Internet

- [7] Comment ça marche ? <http://www.commentcamarche.net/merise/concintro.php3> — Informations sur MERISE.
- [8] Comment ça marche ? <http://www.commentcamarche.net/xml/xmlintro.php3> — Informations sur XML.
- [9] Site du world wide web consortium. <http://www.w3.org/> — Informations sur XML.
- [10] Marc-Aurèle DARCHE. Développement d'architectures à base d'objets distribués. <http://cynode.projet-enoch.net/xml/memoire/generated-html/memoire-index.html> — Informations sur XML et le partage de données.

Troisième partie

Annexes



Annexe A

Organigrammes

Dans les pages suivantes, vous trouverez les organigrammes suivants :

- MAAPAR : schéma A.1, page 40 ;
- CERIT : schéma A.2, page 40.

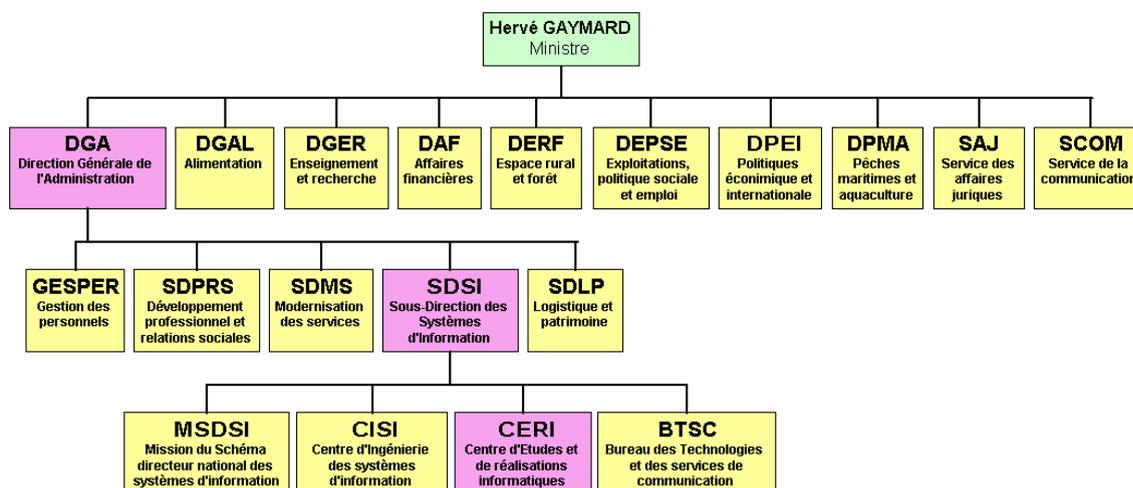


FIG. A.1 – Organigramme du MAAPAR

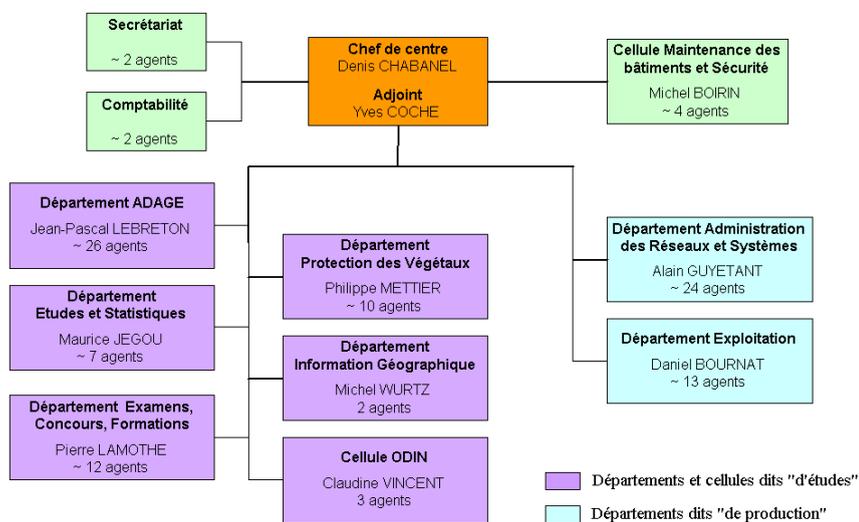


FIG. A.2 – Organigramme du CERIT

Annexe B

Fiche objet de l'objet session

B.1 Description générale

L'objet *session* est rattaché à la table FPC_SESSION. Les objets non-visuels définissant son comportement sont : *fpc_n_cst_metier_session* et *fpc_n_cst_propriete_session*. Une session peut se définir comme suit : « l'unité d'organisation des formations, une session correspond à une certaine formation dispensée à un groupe de stagiaires ».

B.2 Description détaillée

B.2.1 Arbre et liste

On peut voir les informations relatives au *tree-view* et au *list-view* sur le tableau B.1, page 41.

Label	Code de la session pour le centre
Critère de tri	Ordre du numéro de session du centre
Colonnes du <i>list-view</i> (LV)	Informations essentielles relatives à la session (non-définies dans le cadre du prototype)
Picture	3 pictures différents : <ul style="list-style-type: none">– un livre ouvert lorsque tous les champs ne sont pas renseignés,– un document (type icône <code>txt</code> ou <code>rtf</code>) lorsque la session est allégée,– un livre fermé lorsque tous les champs nécessaires sont renseignés.
Overlay picture	Un cadenas lorsque la session est verrouillée (après vérification)

TAB. B.1 – Tableau des informations « Arbre et liste »

B.2.2 Habilitations

Le tableau B.2, page 42 résume les utilisations de l'objet *session*, et les utilisateurs habilités à utiliser ces fonctionnalités.

Code	Habilitation	Description
HAB1	Consultation	CFPPA – SRFD (dont dépend le CFPPA en question – DGER
HAB2	Création	CFPPA
HAB3	Modification	CFPPA qui a créé la session
HAB4	Verrouillage / Déverrouillage	SRFD et DGER

TAB. B.2 – Tableau des habilitations par utilisateur

B.2.3 Fenêtre de propriétés

Celle-ci sera accessible seulement sous la forme d'un assistant. La figure B.1, page 45 montre l'enchaînement des écrans de l'assistant pour la saisie de données relatives à une session.

Une fois la description graphique effectuée, il faut décrire plus précisément toutes les colonnes utilisées. Pour ceci, on procède par onglet, dans un tableau (cf. B.3, page 44).

Code	Colonne	Description
<i>ONGLET : Informations Générales</i>		
FSES_NOM_LB	Nom de la session	Texte libre attribué par le centre dispensant la formation
FSES_CODE_SESSION_RFA	Code de la session	Code attribué par le centre ; dans CIGALE, il est auto-incrémenté par année
FSES_DATEDEBUT_DT	Date de début	Date de début de session
FSES_DATFIN_DT	Date de fin	Date de fin de session ; ces dates sont les dates réelles, même si la session se déroule sur plusieurs années
FSES_DUREECENTRE_NB	Durée en centre	Durée en heures de la formation dans le centre
FSES_DUREEENTREPRISE_NB	Durée en entreprise	Durée en heures de la formation en entreprise
FSES_NOM_FORMATION_LB	Nom de la formation	Texte libre, permet de donner des renseignements sur les formations locales
FSES_TYPE_FORMATION_ON	Type de la formation	Entier permettant de savoir si la formation est référencée, puis d'aiguiller sur le bon onglet <i>Formation</i>
FSES_SAISIE_ALLEGEE_ON	Saisie allégée ?	Entier permettant de spécifier si l'on peut ne pas renseigner tous les champs
SZOR_ZONREC_RFA	Zone de recrutement	Informations utiles pour les statistiques
SIND_INDL_RFA	Individualisation	
SRYF_RYTFOR_RFA	Rythme de la formation	
FSES_STAGH_NB	Effectif Hommes	Nombre de stagiaires hommes et femmes
FSES_STAGF_NB	Effectif Femmes	
FSES_HS_H_NB	Heures-stagiaire Hommes	Nombre d'heures réellement effectuées par les hommes durant la formation (tenant compte des absences)
FSES_HS_F_NB	Heures-stagiaire Femmes	<i>idem</i> pour les femmes
<i>À suivre ...</i>		

Code	Colonne	Description
FSES_DIPLOMES_NB	Effectif diplômé	Nombre de diplômés si la session est certifiante et se finit dans l'année
FSES_ECHECS_NB	Effectif échecs	Nombre d'échecs au diplôme
FSES_ABANDONS_NB	Effectif abandons	Nombre d'élèves abandonnant la formation en cours
FSES_DIPLOMABLES_NB	Effectif diplômables	Nombre de stagiaires comptant passer le diplôme
FSES_SECTION_CDN	Section	Pour les formations sur plusieurs années, permet de savoir où on en est
<i>ONGLET : Formation – MAAPAR</i>		
SREF_REFFORNAT_LIBCOURT_LB	Référentiel	À rapatrier depuis la liste des référentiels avec libellés et codes
SARC_REFARCHI_LIBCOURT_LB	Architecture	À rapatrier depuis la liste des architectures (trois clefs)
FSES_EVALUC_ON	Évaluation par UC ?	Entier permettant de savoir si l'évaluation de la formation se fait par unités de capitalisation
<i>ONGLET : Formation – Autre</i>		
SCH_DIPEN_RFA	Code Diplôme EN	Code du diplôme de l'Éducation Nationale, si la formation appartient à ce référentiel
FSES_NOM_FORMATION_LB	Nom de la formation	Texte libre
SFAM_FAMMET_LIBCOURT_LB	Famille de métier	Affichage du libellé et rapatriement du code
SNIV_NIVINT_LIBCOURT_LB	Niveau interministériel	
SFIL_FIN_LIBCOURT_LB	Finalité de la formation	
STC_CONT_LIBCOURT_LB	Type de certification	
FAF_NUM_CERTIF_RFA	Numéro de certification	Code intégré à la base pour répondre à une éventuelle future nomenclature des certifications
SSPE_LETSPE_LIBLONG_LB	Lettre de spécialité	Ces deux libellés sont aussi rapatriés avec leurs codes
SSPE_GRP_SPE_LIBLONG_LB	Groupe de spécialité	
<i>ONGLET : Financement</i>		
FSF_MONTANT_CONVENTION_NB	Montant de la convention	Saisies en tableau, ces informations sont groupées par type de convention
FSF_STAG_CONVENTION_NB	Nombre de stagiaires	
FSF_HS_CONVENTION_NB	Heures-stagiaires	
SOFL_ORIFINA_LIBCOURT_LB	Origine du financement	Rapatriée avec les clefs correspondantes
<i>ONGLET : Âge des stagiaires</i>		
FAS_STAGH_AGE_SEXE_NB	Hommes par âge	Effectif par tranche d'âge
FAS_STAGF_AGE_SEXE_NB	Femmes par âge	
SVS_VALSEUIL_LIBCOURT_LB	Tranche d'âge	Rapatriée avec les codes correspondants
<i>ONGLET : CSP et activité</i>		
	Hommes par CSP	Effectif par CSP
	Femmes par CSP	
	CSP	Rapatriée avec les codes correspondants
	Hommes par activité	Effectif par activité
	Femmes par activité	
	activité	Rapatriée avec les codes correspondants
<i>À suivre ...</i>		

Code	Colonne	Description
<i>ONGLET : Rémunération</i>		
	Effectif	Effectif par type de rémunération
	Rémunération	Rapatriée avec les codes correspondants
<i>ONGLET : Origine par pays</i>		
	Effectif UE	nombre de stagiaires provenant de l'UNION EUROPÉENNE
	Effectif UE hors France	nombre de stagiaires provenant de l'UE, hors France
	Effectif hors UE	nombre de stagiaires provenant de l'extérieur de l'UE
<i>ONGLET : Origine par région</i>		
	Effectif	Effectif par région d'origine
	Région	Rapatriée avec les codes correspondants

TAB. B.3: Description des colonnes pour l'assistant de création de sessions

B.2.4 Assistant

Les onglets s'enchaînent dans l'ordre :

1. Informations générales ;
2. Formation ;
3. Financement ;
4. Âge des stagiaires ;
5. CSP et activité des stagiaires ;
6. Rémunérations ;
7. Origine par pays ;
8. Origine par région.

Cependant, l'onglet « Formation » se décline en deux possibilités. Ainsi, selon le champ `FSES_TYPE_FORMATION_ON` (type de formation), on passera à l'onglet correspondant aux formations référencées ou aux autres formations. Il faudra donc dans tous les cas « sauter » un onglet en cliquant sur (respectivement).

B.2.5 Règles de gestion

1. Le volume d'heures-stagiaire doit toujours être inférieur ou égal à la durée en centre multipliée par l'effectif total.
2. Cohérence des dates : « date de fin » après « date de début ».
3. Cohérence des effectifs pour tous les onglets (sauf l'onglet « Financement »).
4. Empêcher la saisie légère pour certains types de formation.

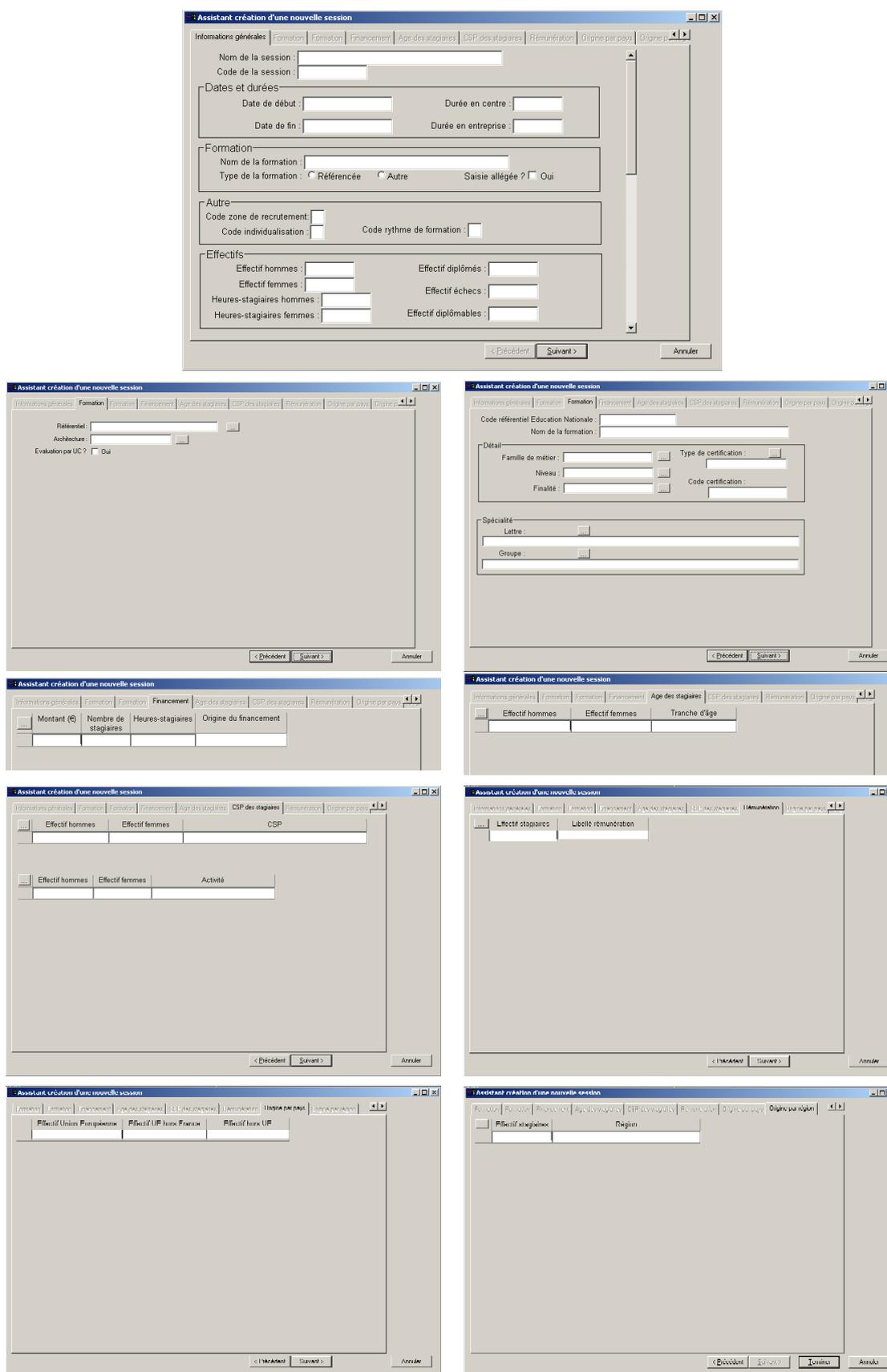


FIG. B.1 – Enchaînement des écrans de l'assistant « propriétés d'une session »

Annexe C

Modèles Conceptuels de Données

Les pages suivantes montrent certains des MCD réalisés.

1. La toute première version ;
2. La version n° 4, intéressante car elle apporte la possibilité de comparer les données sur plusieurs années, et la facilité d'ajout de certaines données (origines de financements, de rémunérations, ...);
3. La version n° 8, qui permet l'interaction avec la base SCHEMAS (référentiel) – c'est aussi la première version à partir de laquelle les MPD ont pu être correctement générés (plus de symbole *XT*) ;
4. La dernière version, n° 16.

Sur ces modèles, les entités extraites de SCHEMAS sont schématisées comme de vraies entités ; en réalité, il s'agit de vues.

Annexe D

Modèle Physique des données

En page suivante, on peut voir le MPD dans sa dernière version n° 13. Les MPD découlant des MCD (ils sont même générés automatiquement par le logiciel PowerAMC), seule la dernière version des MPD est présentée ici (cf. page suivante).

En effet, selon les règles de la modélisation MERISE, les entités deviennent des tables, les associations deviennent des tables et les relations deviennent des liens, avec remontée de clefs (pour simplifier à l'extrême).

De la même façon, les tables issues de SCHEMAS sont en réalité des vues, de façon à conserver la cohérence voulue.

Annexe E

Scripts de création de la base de données

De la même façon que le MPD est automatiquement généré depuis un MCD par PowerAMC, le même logiciel permet de générer des scripts de création de base de données, pour beaucoup de SGBD existants. Il suffit de préciser un fichier de configuration au logiciel pour indiquer sous quel système on travaille, et la génération du script est tout à fait correcte.

Cependant, pour générer les scripts de création des tables de FPC seulement, il faut, lors de la génération, demander au système de ne pas générer les tables qui deviendront des vues et les liaisons vers ces tables. Les pages suivantes montrent le script de création de la base FPC (issu de la dernière version du MCD), qui a servi au développement du prototype, et le script de création des vues vers la base de référence.

Annexe F

Conventions de notations en développement

Type d'objet	Sous-type	Exemple
Fenêtre	<ul style="list-style-type: none">– <i>tree-view list-view</i>– fenêtre de propriétés– assistant– <i>frame</i> (démarrage)	<ul style="list-style-type: none">– fpc_w_tvslvs_sessions– fpc_w_sheet_prop_enquete– fpc_w_assistant_session– fpc_w_frame
<i>Datawindow</i>	<ul style="list-style-type: none">– pour la copie– propriétés– en tableau– <i>tree-view list-view</i>	<ul style="list-style-type: none">– fpc_d_copy_enquete– fpc_d_prop_enquete– fpc_d_tbl_session_oiripays– fpc_d_tv_region
Objet non visuel	<ul style="list-style-type: none">– métier– propriété– arborescence	<ul style="list-style-type: none">– fpc_n_cst_metier_session– fpc_n_cst_propriete_enquete– fpc_n_cst_arborescence_enquete
Objet visuel	<ul style="list-style-type: none">– tableau d'onglets– page (onglet)– <i>control datawindow</i>– tableau d'assistant	<ul style="list-style-type: none">– fpc_u_tab_session– fpc_u_tabpg_session_gen– epv_u_dw– fpc_u_tab_assist_session
Menu	pour chaque fenêtre	fpc_m_light

Annexe G

Extraits de code

G.1 Initialisation de l'affichage

```
uf_initdefDwVisuelle("", "GENERAL","Informations
    générales", "fpc_d_prop_session_gen", "MONOLIGNE")
uf_initdefDwVisuelle("", "MAAPAR","Formation","fpc_d_prop_session_form", "
    MONOLIGNE")
uf_initdefDwVisuelle("", "AUTFORM","Formation","fpc_d_prop_session_autform
    ", "MONOLIGNE")
uf_initdefDwVisuelle("", "FINA","Financement","fpc_d_prop_session_fina", "
    MULTILIGNE")
5 uf_initdefDwVisuelle("", "AGE","Age des stagiaires","fpc_d_prop_session_age
    ", "MULTILIGNE")
uf_initdefDwVisuelle("", "CSP","CSP des stagiaires","fpc_d_prop_session_csp
    ", "MULTILIGNE")
uf_initdefDwVisuelle("", "REMU","Rémunération","fpc_d_prop_session_stag_remu
    ", "MULTILIGNE")
uf_initdefDwVisuelle("", "ORIPAYS","Origine par pays","
    fpc_d_prop_session_oripays", "MULTILIGNE")
uf_initdefDwVisuelle("", "ORIREG","Origine par région","
10    fpc_d_prop_session_orireg", "MULTILIGNE")

return 1
```

G.2 Correction de l'affichage

```
//-----
// ONGLET AUTFORM
//-----
if lower(astr_defdwvisuelle.s_dataobjectvisuel) = "
    fpc_d_prop_session_autform" then
5     // Définir l'objet metier fils XXX
    astr_defdwvisuelle.s_NomObjMetier           = "
        fpc_n_cst_metier_autform"
    astr_defdwvisuelle.s_DataObjectMetier      = "
        fpc_d_prop_session_autform"

    astr_defdwvisuelle.i_ordremaj = 30
10    // Définir l'objet metier pere de AUTFORM
    astr_defdwvisuelle.s_nompereobjmetier      = "fpc_n_cst_metier_session
        "
    astr_defdwvisuelle.s_dwmetierpere         = "
        fpc_d_copy_session"
```

```

15      astr_defdwvisuelle.a_argpere           = astr_defdwvisuelle.a_arg
      astr_defdwvisuelle.tr_datapere = astr_defdwvisuelle.tr_data

      // Faire le retrieve pour une ouverture en mode nouveau ou
      // modification, mais pas en suppression
      // En mode nouveau, une ligne est automatiquement ajoutée à la
      // saisie en tableau
      astr_defdwvisuelle.b_initds = true
      astr_defdwvisuelle.b_majAuto = true
20      astr_defdwvisuelle.b_validAuto = true
      astr_defdwvisuelle.b_initdspere = true
end if

```

G.3 Enchaînement des onglets

```

      CHOOSE CASE as_codeoldonglet
      CASE "GENERAL"
          // Recuperer une info saisie dans un onglet precedent
          if inv_grappemetier.uf_GetObjetMetier( "
              fpc_n_cst_metier_session", lnv_metier_session) < 0
              THEN GOTO Sortie
          if lnv_metier_session.uf_GetDsMetier ("fpc_d_copy_session
              ", lds_metier_session) < 0 then goto Sortie
          // La fonction ci-dessus fait un uf_getDsMetier() pour
          // obtenir
          // une reference a une datastore en cours de saisie puis
          // renvoie
          // une info prelevee dans cette datastore
          ll_typefor = lds_metier_session.GetItemNumber(1, "
              fses_type_formation_on")
10      CHOOSE CASE ll_typefor
          CASE 0
              as_CodeNewOnglet = "AUTFORM"
          CASE 1
              as_CodeNewOnglet = "MAAPAR"
15      END CHOOSE
      CASE "MAAPAR"
          as_CodeNewOnglet = "FINA"
      END CHOOSE [...]

```

G.4 Rapatriement

Appel de la fenêtre de rapatriement

```

      CASE "RAPATRIE_LETSPE"
      // Alimenter la structure de rapatriement
      // Code de l'appli à laquelle appartient la fenêtre à
      // ouvrir (par défaut idem fenetre appelante)
      IF astr_defrapatrie.s_appli = "EPVADMIN" then
          astr_defrapatrie.s_appli = "FPC"
5
      // Nom de la fenêtre à ouvrir (nom PowerBuilder)
      astr_defrapatrie.s_NomFenetreOuvrir = "
          gen_w_tvslvs_rapatric_2"
      // Code de l'onglet actif à l'ouverture

```

```
10      astr_defrapatrie.s_CodeOnglet = "gen_ong_letspe"

      // Les autres onglets sont aussi visibles
      astr_defrapatrie.b_AfficheUnOnglet = true

15      // On cache l'arbre (1 seul niveau => seul, le LV suffit)
      astr_defrapatrie.b_masquearbre = true

      // Nom du niveau à rapatrier
      astr_defrapatrie.s_nomniveau[1] = "LetSpe"
```

Retour des informations

```
IF Not IsValid(astr_defrapatrie.nv_corbeillerapatrie.inv_listecopier) THEN
    return 1

ls_NomRapatriement= Upper(astr_defrapatrie.s_NomRapatriement)
ll_pos = Pos(ls_NomRapatriement,"NOWEBINFO")
5  ls_NomRapatriement=Left(ls_NomRapatriement,ll_pos - 1)
  IF ls_NomRapatriement <> "RAPATRIE_LETSPE" THEN return 1

  //éVrifier que l'on a bien rapatrié qu'un seul enregistrement
  //*****
10 //
  ll_NbRow=astr_defrapatrie.nv_corbeillerapatrie.inv_listecopier.ids_list.
    RowCount()
  IF ll_NbRow<>1 THEN
    ls_ParamMsg[1]="Vous devez rapatrier une seule lettre de spécialité"
    Goto Erreur_nombre
15 END IF

  // Recuperer les infos du rapatriement
  // *****
20 //
  //Recuperer l'identifiant en base de l'enregistrement recupere
  la_arg[1]=astr_defrapatrie.nv_corbeillerapatrie.inv_listecopier.ids_list.
    object.letspe_rfa[ll_NbRow]

  //alimenter la datastore avec l'identifiant et le libelle souhaites
25 astr_defrapatrie.nv_dsmetier.object.letspe_rfa[astr_defrapatrie.l_row]=
    la_arg[1]

  astr_defrapatrie.nv_dsmetier.SetItem(astr_defrapatrie.l_row,"
    letspe_liblong_lb",ls_LibCourt)

return 1
```

Annexe H

Diagrammes de GANTT

Sur les trois pages suivantes se trouvent :

1. le diagramme de GANTT initial, réalisé au cours de la première semaine de stage ;
2. un diagramme mis à jour en milieu de stage (début avril) ;
3. le diagramme remis à jour avant la soutenance, début juin.

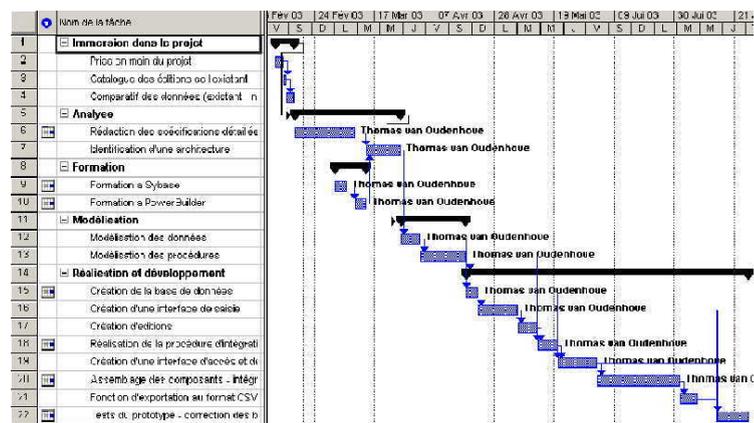


FIG. H.1 – Diagramme de GANTT initial (au 17 février 2003)

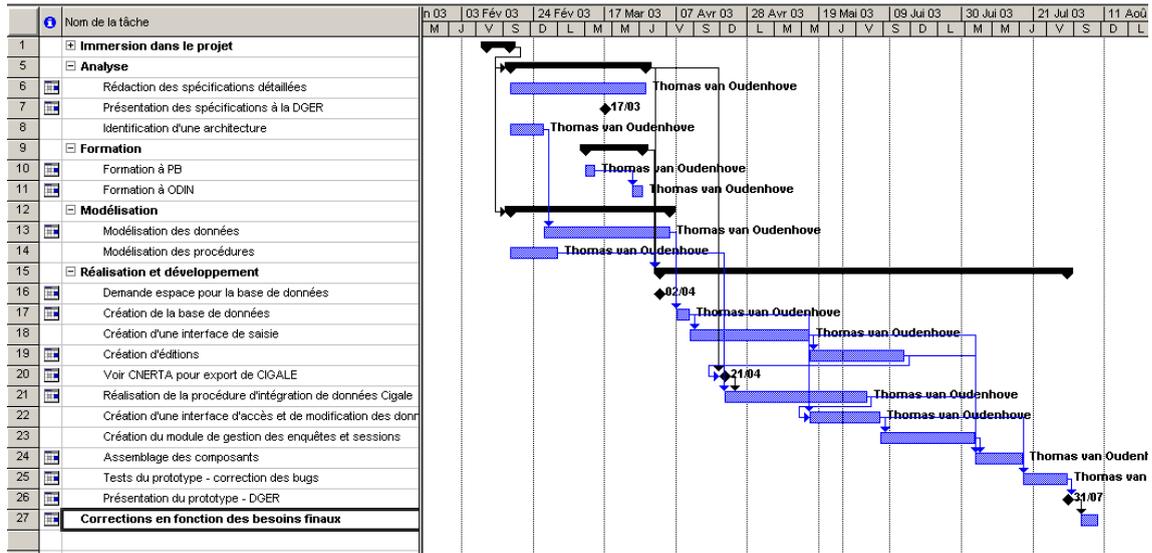


FIG. H.2 – Diagramme de GANTT intermédiaire (au 03 avril 2003)

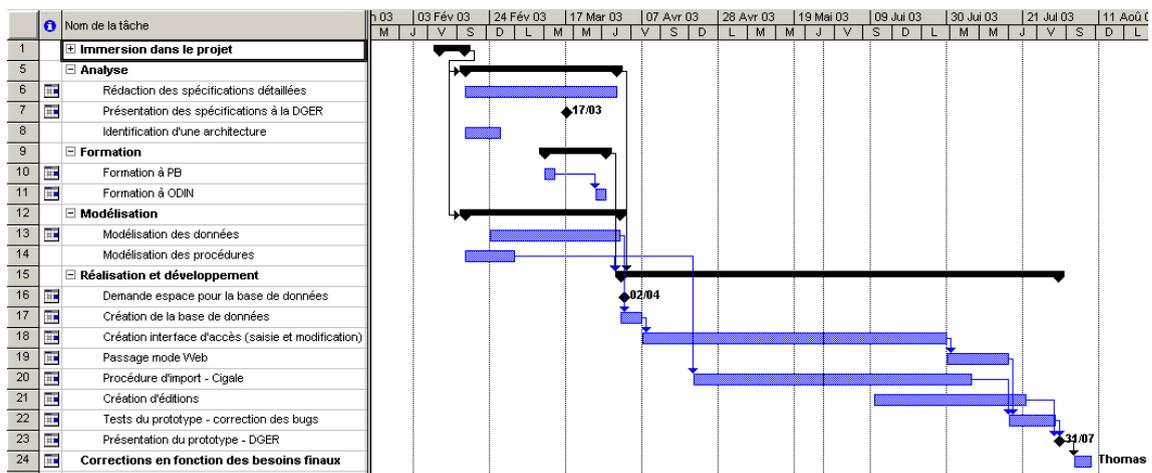


FIG. H.3 – Dernier diagramme de GANTT (au 02 juin 2003)

Annexe I

À propos de ce document

Ce rapport a été rédigé entre les mois d'avril et de juin 2003, en utilisant les outils suivants :

- GNU/(X)emacs pour la saisie du texte ;
- pdfL^AT_EX pour la mise en page ;
- MS WINDOWS et SUN/Solaris comme systèmes d'exploitation ;
- MS VISIO pour les schémas ;
- GNU/Gimp (Gnu Image Manipulation Program) pour les captures d'écrans.

Ce document est la propriété intellectuelle de Thomas VAN OUDENHOVE¹, excepté les images en tête de parties², et est protégé par les lois du copyright.

Copyright ©2003 – Thomas VAN OUDENHOVE DE SAINT GÉRY

¹Pour plus d'informations, cf. <http://www.enstimac.fr/~vanouden/>

²Ces images sont issues du site <http://www.agriculture.gouv.fr>