



HAL
open science

Système de gestion de l'information pour les activités expérimentales et agricoles du site de Montoldre

K. Soullami

► **To cite this version:**

K. Soullami. Système de gestion de l'information pour les activités expérimentales et agricoles du site de Montoldre. Sciences de l'environnement. 2012. hal-02598107

HAL Id: hal-02598107

<https://hal.inrae.fr/hal-02598107>

Submitted on 15 May 2020

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Master 1 Informatique Décisionnelle et Logicielle

Rapport de stage

Système de gestion de l'information pour les activités expérimentales et agricoles du site de Montoldre

Kenza SOULLAMI

Responsable pédagogique : *M. Yannick LOISEAU*, Enseignant et Responsable du Master 1 IDL.

Responsable professionnel : *M. Frédéric CHABOT*, Responsable qualité de l'unité de recherche TSCF
et responsable des moyens agricole du site de Montoldre.

Période de stage : Du 03 Avril 2012 au 31 Août 2012
Au sein d'Irstea Site de Montoldre



Domaine des Palaquins
03150 Montoldre

Remerciements

Je tiens tout d'abord à remercier mes professeurs de l'Université Blaise Pascal pour leur soutien durant mon année universitaire et plus particulièrement Monsieur Yannick LOISEAU, responsable du Master 1 IDL, qui m'a suivi tout au long de mon stage.

Je tiens aussi à remercier Monsieur Frédéric CHABOT, responsable des moyens agronomiques du site et responsable qualité de l'unité de recherche TSCF, de m'avoir accueilli dans son service, de m'avoir suivi tout au long de mon stage et de m'avoir fait confiance et considéré comme membre à part entière de son équipe.

Mademoiselle Géraldine ANDRÉ qui m'a beaucoup aidée et m'a soutenue tout au long du stage.

Monsieur Nicolas DE FREITAS pour son aide et ses conseils.

Je remercie aussi Daniel BOFFETY, Eveline BONAMY, Irène MINGOT, Anne-Laure REVERDY, Alice GUEUDET, Denis MICLET Anicet MARIONNEAU, Jean-Christophe ROUX, Marilys PRADEL, Sérigne MBAYE, Jean-Claude MEGNIEN ainsi que tous les stagiaires (Fabrice, Michaël, Arnold, Damien, Julian, Teddy, Morgan) pour leur accueil, leur aide et leur sympathie. Ils ont été de précieux atouts tout au long de ce stage et ce fut un réel plaisir de travailler avec eux.

Enfin, un grand merci à toutes les personnes de l'équipe COPAIN qui ont toujours pris le temps de répondre à mes questions afin de satisfaire ma curiosité et de partager leurs connaissances. Je les remercie également pour tous les conseils qu'ils m'ont prodigués.

Table des figures

Figure 1: Répartition des effectifs d'Irstea	10
Figure 2: Le site de Montoldre	11
Figure 3: Organigramme TSCF.....	12
Figure 4: Implantations géographiques	12
Figure 5: Gestion des calques sous Carto'Ciel	15
Figure 6: Schéma organisationnel	19
Figure 7: Modèle conceptuel simplifié	24
Figure 8: Exemple de classes Python : Equipement	24
Figure 9: Structure du projet	26
Figure 10 : Interface d'authentification	27
Figure 11: Code d'ajout des tables dans l'interface d'administration	27
Figure 12: Administration du site.....	28
Figure 13: Ajout d'un équipement.....	29
Figure 14: Administration de l'interface d'ajout d'une intervention.....	30
Figure 15 : Représentation de plusieurs classes dans un même formulaire.....	31
Figure 16 : Choix du type de l'intervention.....	32
Figure 17: Schéma simplifié du fonctionnement du script	33

Résumé

Depuis 2006, le site de Montoldre a fait le choix de s'approprier de l'outil Carto'Ciel, un logiciel de gestion technico-économique, qui a été paramétré pour permettre de tracer les interventions réalisées sur les parcelles de l'exploitation agricole. Les fonctionnalités de Carto'Ciel ne permettent pas de répondre totalement aux besoins de traçabilité pour l'ensemble des activités, conduites sur le site notamment pour les données expérimentales générées par les chercheurs et les ingénieurs qui utilisent l'exploitation agricole comme plateforme expérimentale dans le cadre de leur développement de recherche.

Aujourd'hui Irstea a besoin d'une autre solution pour subvenir à la totalité de ses besoins. Pour cela il est nécessaire de développer un nouveau système de gestion d'informations du site de Montoldre, aussi bien pour les informations classiques des expérimentations agricoles que géographiques (SIG).

Mon projet consiste tout d'abord à concevoir un modèle de base de données en me basant sur les informations nécessaires extraites de Carto'Ciel, ainsi qu'un cahier des charges réalisé par une stagiaire l'année dernière. Par la suite, je développerai un système pour la gestion des informations concernant les expérimentations agricoles du site Montoldre.

Sommaire

INTRODUCTION	7
I. CONTEXTE ET SUJET DE STAGE	8
1. PRÉSENTATION DE L'ÉTABLISSEMENT D'ACCUEIL	8
1.1 <i>Irstea</i>	8
1.2 <i>Présentation du site de Montoldre [1]</i>	10
2. DESCRIPTION GÉNÉRALE DU SUJET DU STAGE	13
4. II. ANALYSE ET CONCEPTION DU PROJET	14
1. PROBLÉMATIQUE.....	14
2. L'OUTIL CARTO'CIEL.....	14
3. MODÉLISATION ET CONCEPTION DU PROJET	15
4. MISSIONS À ACCOMPLIR	15
5. CHOIX DES OUTILS ET TECHNOLOGIES.....	16
5.1 <i>Objecteering</i>	16
5.2 <i>PostgreSQL</i>	17
5.3 <i>Django, le Framework Web-Python</i>	17
III. TRAVAIL EFFECTUÉ	19
1. ANALYSE DES DONNÉES CARTO'CIEL	20
2. BASE DE DONNÉES	22
2.1 <i>Modélisation</i>	22
2.2 <i>Structure</i>	23
2.3 <i>Implémentation</i>	24
3. CRÉATION DES INTERFACES DE SAISIE.....	25
3.1 <i>Administration du système</i>	27
3.2 <i>Interface de saisie</i>	31
4. SCRIPT D'INTÉGRATION.....	32
CONCLUSION ET PERSPECTIVES	34
WEBOGRAPHIE ET BIBLIOGRAPHIE	35
GLOSSAIRE	36
ANNEXES	38

Introduction

Dans le cadre de ma formation en première année du Master Professionnel Informatique Décisionnelle et Logicielle, j'ai effectué un stage de 5 mois au sein d'Irstea (Institut de Recherche en Sciences et Technologies pour l'Environnement et l'Agriculture) et plus particulièrement sur le site de Montoldre.

Ma mission au cours de ce stage, est de recueillir et d'analyser les différents besoins des utilisateurs, d'établir un cahier des charges rattaché aux données de l'outil Carto'Ciel, de modéliser et de mettre en place une base de données centrale regroupant toutes ces données relatives à l'utilisation des moyens agronomiques du site, puis de développer un système pour la consultation et la restitution des informations nécessaires aux utilisateurs.

Le présent rapport présente ce que j'ai appris et réalisé tout au long de mon stage. Son objectif premier est de proposer une description globale des problématiques posées par le sujet du stage. Il expose aussi la démarche à suivre pour répondre aux différentes problématiques. Il s'organise en trois parties :

- Présentation de l'établissement d'accueil, l'activité et le contexte général du stage.
- Description du cadre et du déroulement de ma mission, le sujet de stage, ses objectifs et les problématiques engendrées.
- Présentation de la mise en place de la base de données centrale ainsi que la réalisation du projet en sa globalité.
- Le rapport se terminera par une conclusion.
- Un glossaire est présent à la fin du rapport. Il définit les différents mots techniques et fonctionnels utilisés.

I. Contexte et sujet de stage

1. Présentation de l'établissement d'accueil

1.1 Irstea

Pour mieux affirmer ses missions, le Cemagref adopte un nouveau nom : Irstea, "Institut national de recherche en sciences et technologies pour l'environnement et l'agriculture." [1]

Créé en 1981, le Cemagref est depuis 1986 un Établissement Public à caractère Scientifique et Technologique (EPST). Il est placé sous la double tutelle des ministères en charge de la recherche et de l'agriculture. Les équipes de chercheurs et d'ingénieurs du Cemagref travaillent sur l'adaptation au changement global, associant les sciences expérimentales, les sciences économiques et sociales et les sciences informatiques (modélisation, intégration de données). Il a pour mission de répondre à des questions concrètes de société dans différents domaines afin de produire des connaissances nouvelles et des innovations techniques utiles aux gestionnaires, aux décideurs et aux entreprises :

- Risques environnementaux crues, inondations, avalanches, feux de forêts, pollution diffuses;
- Technologies propres écotechnologie, éco-évaluation, éco-toxicologie, traitement et valorisation énergétique des déchets organiques;
- Aménagement du territoire;
- Économie et sociologie de l'environnement observatoire de la biodiversité, télédétection.

Historique [12]

En **1981** : Le Cemagref est fondé à partir d'équipes scientifiques et techniques constituées par l'État dans les années 1960-1970 ;

De **1981** à **1993** : Les employés du Cemagref sont répartis en 43 divisions et une direction des programmes gère la cohésion entre les différentes activités ;

De **1994** à **1997** : Premier plan stratégique qui permet de mettre en place quatre départements possédant chacun un plan quadriennal. Le Cemagref (originellement Centre National du Machinisme Agricole, du Génie Rural des Eaux et des Forêts) devient l'Institut de Recherche pour l'Ingénierie de l'Agriculture et de l'Environnement.

La création du département "Gestion des territoires" témoigne du développement d'une nouvelle thématique ;

En **1998** : Une direction scientifique et une direction du développement et de l'innovation sont mises en place. Cette distinction permet une amélioration en terme de qualification scientifique et de recherche de partenaires ;

De **1999** à **2003** Un nouveau plan stratégique modifie la maille des thèmes de recherches, qui devient celle du pilotage stratégique et du suivi rapproché par les départements ;

De **2004 à 2008** : Un nouveau plan stratégique est mis en place. Ce dernier s'inscrit dans une vision sur dix ans et dans une optique d'insertion européenne. Le nombre de thèmes de recherche est réduit à 27 afin de resserrer les collectifs thématiques par implantations.

Chaque collectif doit fixer ses objectifs à 4 ans. Un des quatre départements s'oriente résolument vers le développement des écotechnologies ;

En **2009** : L'objectif fixé par l'État est clair : le Cemagref doit devenir la référence en sciences pour l'ingénierie de la gestion durable des eaux et des territoires. La nouvelle dénomination de l'établissement, "Sciences, eaux et territoires ", confirme cette orientation. 2009 a été la première année de la mise en œuvre du nouveau plan stratégique "Cemagref 2020". Ce plan à moyen terme oriente les ambitions scientifiques de l'institut sur trois grands axes : l'eau et les territoires, les risques naturels et la qualité environnementale et apporte une cohérence dans tous les thèmes de l'institut.

Domaine d'activité

Établissement public à caractère scientifique et technologique, le Cemagref forme un trait d'union entre la recherche fondamentale, les pouvoirs publics et les utilisateurs (constructeurs de machines agricoles, exploitants, etc.). En effet, le Cemagref développe ses recherches et travaille en collaboration avec d'autres organismes de recherche, université, industriels et collectivités territoriales publiques, ce qui lui permet d'intégrer les réalités du terrain. [1]

Le Cemagref s'appuie aussi sur de nombreux partenariats, qu'ils soient scientifiques, publics ou socio-économiques. Il répond aux besoins exprimés par les ministères de l'agriculture et de l'environnement ainsi qu'à ceux des collectivités locales mais il développe également des projets avec des industriels et des bureaux d'études. Par ailleurs, l'établissement valorise les résultats de sa recherche auprès des entreprises en déposant des brevets (au nombre de 45 à ce jour), en créant des marques (16 sont déposées) et en élaborant des logiciels professionnels (dont 11 sont commercialisés). [1]

Une des missions de l'organisme de recherche est aussi de développer la production et la diffusion de nouvelles connaissances. Ceci passe par des publications scientifiques, au rythme d'environ 400 par an (pour 2009). Enfin, les enseignants-chercheurs du Cemagref dispensent des cours aux étudiants de licence, master et doctorat à hauteur de 10 % du temps annuel des ingénieurs (soit 10 000 heures d'enseignement supérieur par an en 2009). [1]

La répartition des effectifs (1600 personnes) en fonction des différentes activités de l'établissement est représentée ci-dessous :

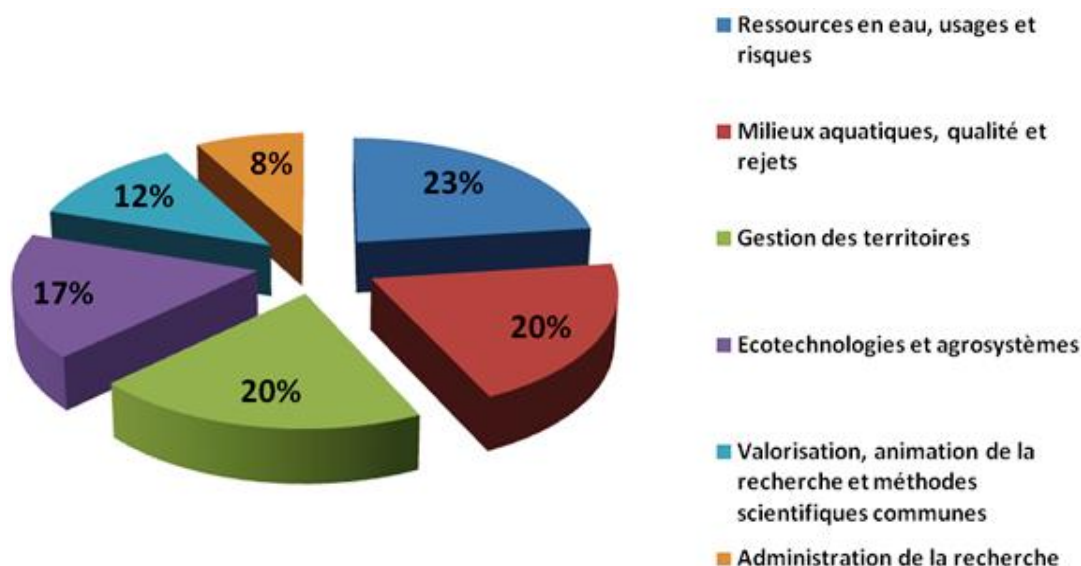


Figure 1: Répartition des effectifs d'Irstea

Le centre de Clermont-Ferrand

Le centre de Clermont-Ferrand bénéficie d'une localisation bi-site: Aubière, sur le campus universitaire des Cézeaux, et Montoldre, où résultats académiques et réalités de terrain peuvent être confrontés. Les recherches conduites concernent les technologies et systèmes d'information pour les agro-systèmes, la modélisation des systèmes complexes et les dynamiques territoriales des espaces ruraux. Ses chercheurs sont sollicités pour leurs capacités d'expertise par la sphère publique. Le centre agit également en interconnexion avec le secteur économique : ses recherches sont vectrices d'innovation. Fort d'une réelle reconnaissance régionale, fruit de son implication, il pratique une recherche pluridisciplinaire en faveur du développement des territoires. [1]

1.2 Présentation du site de Montoldre [1]

Le site de recherche et d'expérimentation de Montoldre, est l'une des deux seules fermes expérimentales d'Irstea, la deuxième ferme se situant à Saint Seurin sur l'Isle en Gironde.

Le site de Montoldre dépend du centre régional de Clermont-Ferrand et porte les activités de l'unité de recherche TSCF (Technologie et Système d'information pour les agro-systèmes Clermont-Ferrand), qui mène des recherches sur les méthodes et les outils pour une ingénierie des systèmes complexes que sont les systèmes agro-environnementaux, elle-même rattachée au département écotechnologie et agro-système. Ce site comprend une exploitation agricole d'une superficie de 140 Ha ce qui permet de disposer du matériel nécessaire aux essais et travaux de recherches (plate-formes expérimentales, engrais minéraux, engrais organiques, exploitation agricole, etc.). Le site accueille 30 employés permanents.

Les recherches portent sur les technologies de la mobilité pour la sécurité et la qualité du travail des machines mais aussi sur les technologies d'épandage de matériaux organiques ou minéraux. Les systèmes écologiques sont également abordés, à travers les technologies

pour la perception et la caractérisation de l'environnement. Enfin, des travaux ont pour objectifs d'adapter les systèmes d'informations aux besoins individuels et collectifs de gestion des entreprises agricoles et des milieux naturels, de garantir la qualité de conception et de réalisation et d'accroître la communication inter-systèmes. Un plateau technique sur les épandages et l'environnement mène les recherches dans ce domaine et facilite l'appui aux industriels en matière d'innovation ainsi que la promotion du concept d'écotechnologie de l'épandage. Ce plateau est doté d'une plate-forme technologique d'expérimentation de pointe unique en Europe. L'unité de recherche TSCF est membre de la fédération de recherche TIMS (Technologies de l'Information, de la Mobilité et de la Sécurité). Cette unité de recherche regroupe trois équipes réparties sur deux sites (Clermont-Ferrand et Montoldre) :

- Systèmes d'information agri-environnementaux communicants (équipe COPAIN).
- Technologies de la mobilité pour la sécurité et la qualité du travail des machines agri-environnementales (équipe TEAM).
- Technologies pour la caractérisation du sol appliquées aux éco et agro-systèmes, Sciences-technologies et procédés d'épandage. (équipe Carac"Terre).

L'unité de recherche TSCF dirigée par Monsieur Emmanuel HUGO, regroupe 51 personnes (personnels permanents) dont 29 chercheurs ou ingénieurs, 16 assistants ingénieurs ou techniciens, 6 administratifs, ainsi que 7 doctorants, 2 post-doctorants et 15 stagiaires par an en moyenne.



Figure 2: Le site de Montoldre

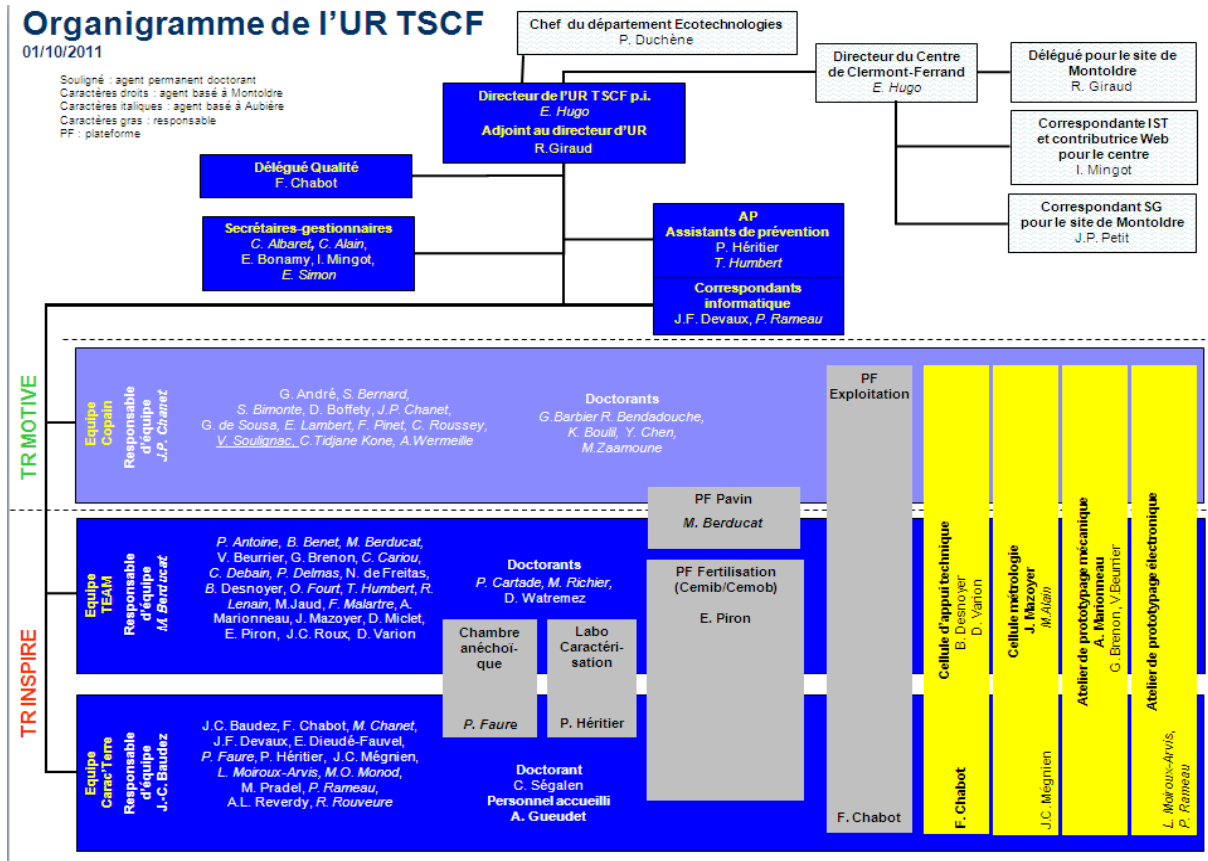


Figure 3: Organigramme TSCF

Implantations géographiques d'Irstea



Figure 4: Implantations géographiques

Les infrastructures d'Irstea sont réparties sur neuf centres (Figure 4).

Chaque centre possède ses propres spécialités en fonction de son implantation sur le territoire. Ainsi, les différents thèmes concernant chaque centre sont: [11]

- Aix-en-Provence: Risques liés à l'eau et vulnérabilité des écosystèmes.
- Antony: Eau, déchets, froid, agroéquipements.
- Bordeaux: Eau, biodiversité, gestion des territoires.
- Clermont-Ferrand: Agrosystèmes et dynamiques territoriales des espaces ruraux.
- Grenoble: Gestion des territoires de montagne et risques.
- Lyon: Hydrologie, qualité des eaux, biologie des écosystèmes.
- Montpellier: Agrosystème, biodiversité, aménagement du territoire.
- Nogent sur Vernisson: Biodiversité et gestion forestière.
- Rennes: Froid, traitement des déchets.

2. Description générale du sujet du stage

Le projet de développement d'un système de gestion de l'information pour les activités expérimentales passe préalablement par le choix de la technologie appropriée. Les étapes à suivre pour cette réalisation consistent à implanter une base de données centralisée qui regroupe toutes les données concernant les activités agricoles et expérimentales du site de Montoldre. La centralisation des données permettra aux utilisateurs d'accéder à l'information, puisque la problématique générale est de compléter les fonctionnalités de l'outil Carto'Ciel qui est utilisé depuis 2006 pour tracer toutes les opérations effectuées sur l'exploitation, d'intégrer des données d'agriculture de précision comme les cartographies de rendements ou les préconisations de modulation d'engrais. Ces données ainsi enregistrées ne peuvent pas être mises à disposition puisque CartoCiel est installé uniquement sur le poste du responsable des moyens agronomiques. Il est maintenant souhaité d'étendre le partage des données issues des activités conduites sur l'exploitation agricole du site de Montoldre pour que celles-ci soient accessibles à un plus grand nombre d'agents de l'unité de recherche TSCF.

C'est dans ce but que ce projet sera réalisé, avec l'intention de concevoir un nouvel outil permettant d'interroger la base de données centrale, dont la capacité de stockage est plus importante que celle de la base de données Carto'Ciel. Pour cela il sera nécessaire de récupérer les informations de la base Carto'Ciel générées sous format de fichiers .dbf qui concernent les interventions réalisées pour toutes les activités agricoles effectuées sur l'exploitation. Pour les données expérimentales, ce sera aux propriétaires des données (chercheurs, ingénieurs et techniciens) de renseigner la base à partir des expérimentations qu'ils conduiront sur les parcelles de l'exploitation.

4. II. Analyse et conception du projet

1. Problématique

Les questions qui se posent sont sur quelles bases se focaliser pour mettre en œuvre une base de données dont l'accès concerne un certain nombre d'acteurs et pas d'autres, comment regrouper ces données et les coordonner avec celles de l'outil Carto'Ciel (pas toutes les données stockées dans la base de données Carto'Ciel, mais seulement celles pouvant être utiles aux chercheurs et ingénieurs de l'unité de recherche).

Pour ce faire, j'ai pensé à créer un script qui permettra de récupérer les données saisies dans Carto'Ciel pour les intégrer dans la base de données centrale.

Ce travail demande un double effort puisqu'il faut d'abord analyser les données de Carto'Ciel et faire la correspondance entre ces données et celles de la BDD centrale dans le but d'éviter la double saisie quand l'utilisateur aura à renseigner la base de données.

2. L'outil Carto'Ciel

Avant d'entamer la conception de la solution citée plus haut, il est important d'introduire l'outil actuellement utilisé sur le site de Montoldre : Carto'Ciel Entreprise.

Carto'Ciel est un logiciel destiné aux techniciens et partenaires des agriculteurs qui réalisent des prestations de gestion et de cartographie des exploitations. Au niveau de l'Irstea, il ne sera utilisé que pour l'exploitation du site mais ce sont ses nombreuses fonctionnalités et les possibilités de paramétrage qu'il offre qui ont orienté vers ce choix. C'est un outil évolutif qui se divise en différents niveaux. Chaque niveau concerne un nombre de thèmes de travail : gestion du parcellaire, agriculture de précision, itinéraires techniques, etc.

Concernant le stockage sur le disque dur, tous les fichiers relatifs au fonctionnement de l'application sont localisés par défaut dans le répertoire C:\Plainets. Cependant, toutes les données sont stockées dans un dossier portant le nom de l'exploitation agricole. Les bases de données, elles, sont stockées au format Database File (*.dbf). Les calques qui enregistrent l'information géographique, ont le format Table Mapinfo (*.tab).

Carto'Ciel est donc un ensemble de bases de données permettant de stocker l'information, non seulement de l'information sémantique, mais aussi géographique. Mais puisque les deux types d'information sont indépendants, la suppression d'un élément géographique n'entraînera pas celle de l'élément sémantique lui correspondant.

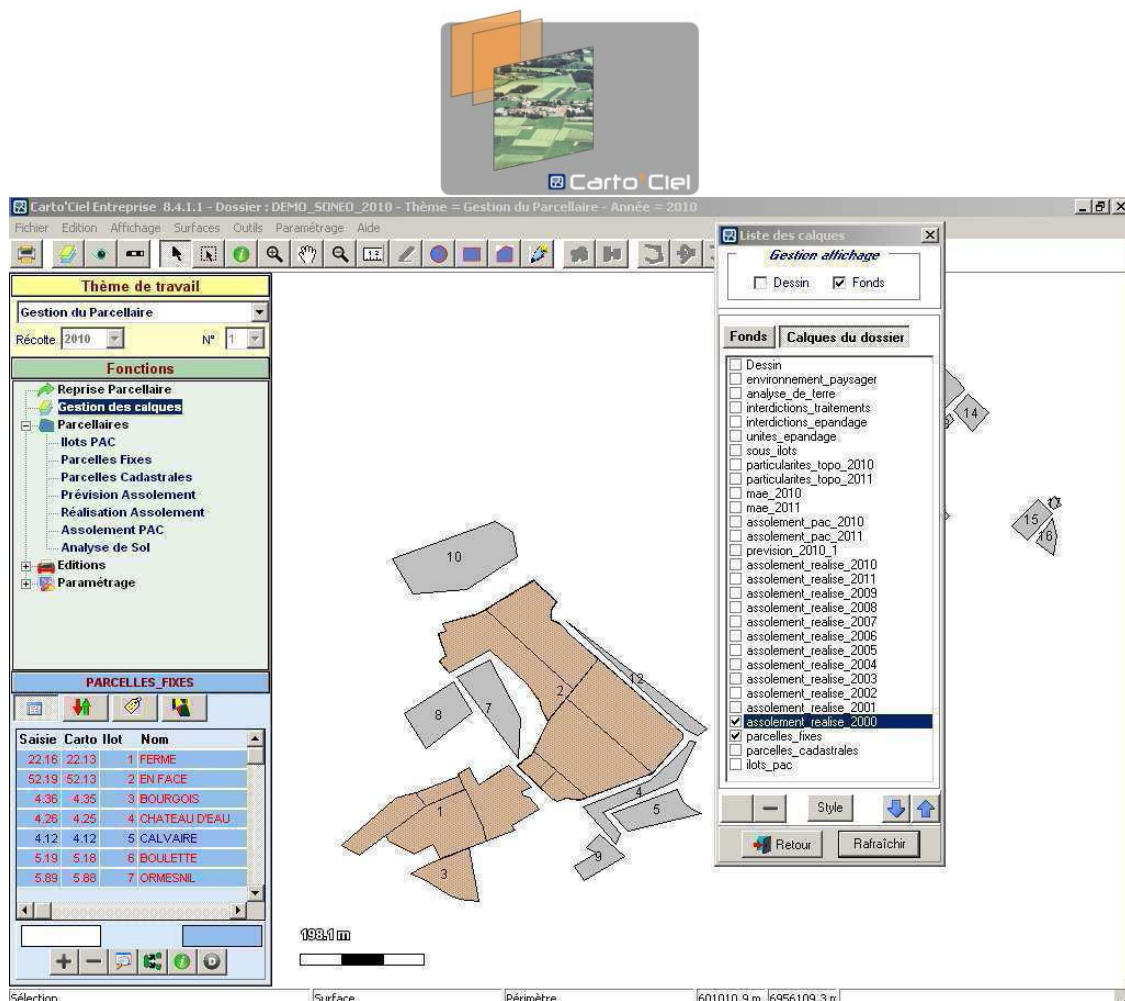


Figure 5: Gestion des calques sous Carto'Ciel

3. Modélisation et conception du projet

Dans un premier temps, j'ai été amenée à faire le recueil du besoin tout en me basant sur l'analyse de l'existant et en vérifiant les données traitées par Carto'ciel ainsi que l'analyse faite par une ancienne stagiaire. Par la suite j'ai entamé la conception de la base de données en UML, sous le logiciel Objectteering Modeler. J'ai généré par la suite un dictionnaire de données décrivant les données dont on aura besoin pour la base de données centrale. (Voir l'annexe)

4. Missions à accomplir

La base de données centrale doit permettre de tracer l'ensemble des opérations réalisées sur l'exploitation agricole comme support d'expérimentations, une meilleure traçabilité des interventions réalisées sur la totalité des parcelles, aussi bien le travail agricole que les expérimentations dans le cadre des travaux de recherche.

Pour répondre à la problématique, les missions que j'ai eu à accomplir consistent à :

- Recueillir les besoins utilisateurs.
- Analyser les besoins recueillis.
- Analyser le travail de Sarah BONIS.

- Rédiger les comptes-rendus des réunions réalisées durant la réalisation de ce projet. (Voir Annexe).
- Produire un modèle conceptuel pour la BD.
- Générer un cahier des charges et le finaliser.
- Produire un modèle physique de la BD.
- Mettre en place la BD sous PostgreSQL.
- Développer le système de gestion des données.
- Préparer une documentation pour les utilisateurs.

5. Choix des outils et technologies

Afin de mener à bien à la réalisation du projet, il a fallu choisir les outils les plus adaptés pour créer une interface qui interagit avec la base de données centrale tout en optimisant le temps de création et en permettant d'assurer la maintenance du système pour une éventuelle évolution aussi bien pour l'interface que pour le modèle de données.

Après une étude des différentes technologies j'ai choisi d'utiliser le framework Django basé sur le langage Python, pour sa génération rapide d'une interface web, Cela m'a permis notamment d'acquérir des connaissances en Django ainsi que dans la programmation Python.

Cependant j'ai choisi le langage Python pour le développement parce qu'il est relativement multiplateforme et qu'il possède une syntaxe légère et compréhensible grâce à l'utilisation des indentations.

5.1 Objecteering

Objecteering 6 est un outil de modélisation UML 2.0. Objecteering 6 permet l'analyse et la conception jusqu'à la génération de code, tests et déploiement d'application de notre projet. Cet outil offre aussi de nombreux modules à déployer selon les besoins du projet. La modélisation est facilitée, les documents et le code restent conformes aux modèles produits et tout est automatisé ce qui permet un gain de temps évident. [3]

Objecteering fournit une solution de modélisation UML complète. Il permet :

- La création et l'édition de modèles dans l'explorateur et les diagrammes.
- La génération de documentation.
- La génération de tout le code (Java, C#, C++).
- La vérification de la qualité et de la précision de la modélisation grâce aux contrôles de cohérence et aux métriques d'Objecteering.

Objecteering fournit de puissants mécanismes de cohérence :

- Assistance à la modélisation : aide pendant la modélisation en fournissant la liste des éléments de modèle qui peuvent être liés à un élément édité.
- Gestion de cohérence : la garantie qu'un modèle reste cohérent en fonction des changements apportés. Les modifications apportées à une vue du modèle (diagramme, fenêtre auxiliaire ou explorateur) sont automatiquement reportées aux autres vues. Par exemple, si le nom d'une classe est modifié dans l'explorateur, il est également

modifié dans la fenêtre auxiliaire, dans tous les diagrammes qui contiennent la classe, et dans tous les éléments qui s’y réfèrent (paramètres d’opérations, instances, ...).

Dans certaines situations de modélisation, il peut être préférable d’avoir une certaine flexibilité dans la gestion des contrôles de cohérence. Pour cette raison, certains contrôles de cohérence Objecteering peuvent être désactivés.

5.2 PostgreSQL

Mon choix de Système de Gestion de Base de Données (SGBD) était dirigé vers PostgreSQL qui est un SGBD OpenSource, et parce qu’il est capable de gérer les données géographiques pour l’évolution à venir de la Base de Données (BDD).

Pendant des années, le marché considérait MySQL comme plus rapide et plus facile à utiliser que PostgreSQL. PostgreSQL avait la réputation d’être plus puissant, focalisé sur l’intégrité des données, et plus respectueux des normes SQL, mais également plus lent et plus compliqué à utiliser. [4]

Ces perceptions appartiennent au passé. Aujourd’hui les deux systèmes ont évolué avec des versions notables qui rendent leur comparaison beaucoup plus compliquée. PostgreSQL stabilise et distribue une nouvelle version majeure de sa base de données chaque année. Le cycle de développement actuel est composé de deux grandes étapes, six mois de développement suivis de six mois de relectures, tests, correctifs, polissages et améliorations de la documentation.

Il existe ensuite de nombreux points techniques donnant un avantage très net à PostgreSQL, soit qu’il s’agisse d’innovations technologiques issues de la recherche, ou bien qu’il s’agisse de supports avancés aux développeurs, telles les recherches des plus proches voisins via un parcours d’index, les requêtes avancées en écriture, ou bien le support avancé des types de données souvent utilisés. Ces capacités avancées, cette souplesse d’utilisation et de paramétrage dynamique, associés à des caractéristiques de performance époustouflantes, la qualité de sa documentation, tout cela donne un avantage crucial à PostgreSQL. [5]

La raison pour laquelle les qualités techniques de PostgreSQL sont vantées s’adressent à la simplicité d’utilisation. Il s’agit d’un outil robuste permettant de produire des applications meilleures car plus simple à développer, à faire évoluer, à maintenir, déployer et administrer. Bien évidemment, tout cela sans s’acquitter de coûts de licence appliqués par serveur ou qui dépendent de la capacité et du nombre d’installations dont on a besoin pour déployer une architecture. Mettre en place une solution tolérante aux pannes, de la répartition de charge, des instances de tests ou d’exports de données deviennent une décision plus facile à prendre car elle ne dépend pas des coûts de licence.

5.3 Django, le Framework Web-Python

Histoire de Django

Django est un framework développé en Python, initialement pour un journal local dans le Kansas : World Online. [7]

En 2003, deux développeurs (Adrian Holovaty et Simon Willison) ont décidé d'abandonner le langage PHP pour se mettre au Python afin de développer leur site dédié aux faits actuels. Pour améliorer le temps de développement, ils ont décidé de mettre en place un framework, une structure simple permettant de réduire considérablement le temps de développement d'un site. En deux ans, ce moteur a beaucoup changé, et change encore aujourd'hui, avec des ajouts et corrections. [7]

C'est en 2005 que World Online décide d'ouvrir les portes de son framework : Django. Depuis ce framework est utilisé par de nombreux développeurs ayant besoin de développer des sites de grande qualité et très rapidement.

Pourquoi utiliser Django ?

L'avantage d'utiliser un framework existant et surtout Open Source tel que Django, est que les bugs sont corrigés rapidement et les améliorations sont exécutées par plusieurs personnes.

Il existe de nombreux framework web, dans différents langages de programmation. Pourquoi utiliser Django et pas un autre ?

J'ai choisi Django pour plusieurs raisons :

- La simplicité d'apprentissage.
- La qualité des applications réalisées.
- La rapidité de développement.
- La sécurité du site Internet final.
- La facilité de maintenance des applications sur la durée.

En plus, on bénéficie de la clarté de Python, qui permet à plusieurs développeurs de travailler sur le même projet. Le style est imposé, donc tout le monde suit les mêmes règles, ce qui facilite les travaux en équipe et la clarté du code. En comparaison avec le PHP, on peut placer des morceaux de codes PHP au milieu de pages HTML (Hypertext Markup Language), ou encore utiliser un moteur de Template pour séparer le code du HTML.

En Python/Django, tout est mis en place, et ce dès le début de l'apprentissage.

Le code Python est repartitionné selon son objet, et donc on aura différents fichiers à traiter :

- Le fichier `models.py` : décrit la table pour stocker les données sous la forme d'une classe Python. Cette classe est appelée modèle. C'est dans ce fichier que j'ai spécifié les classes de la BDD.

- Le fichier `views.py` contient la logique de la page, sous la forme d'une fonction Python. Cette fonction est appelée vue. Ce fichier m'a servi pour écrire les fonctions que j'ai utilisées durant le développement.
- Le fichier `urls.py` définit quelle vue sera appelée pour un modèle d'URL (Uniform Resource Locator) donné.
- Le fichier `nom_fonction.html` est un gabarit HTML définissant l'aspect de la page. On l'appellera un template.

Malgré ses points positifs, Django possède quelques inconvénients. En effet, Django facilite la réalisation d'applications web avec une base de données grâce à son ORM (Mapping Object Relationnel) qui permet l'accès à la base de données via une classe Python. [9]. Par contre, la limite de l'ORM vient lorsqu'il est nécessaire de faire évoluer le modèle de données (exemple : ajouter un champ dans le modèle correspond à ajouter une colonne dans une table), il faut faire évoluer la base pour ajouter une nouvelle colonne dans une table. Malheureusement la commande « syncdb », pour la mise à jour de la base de données ne gère pas les modifications apportées à une table.

Pour résoudre ce problème, j'ai utilisé un utilitaire nommé « South » qui est une application permettant de simplifier la gestion de la base de données durant le processus de développement d'une application Django, en offrant un système de gestion de versions pour les modèles de données.

III. Travail effectué

Pour introduire cette partie j'ai pensé à schématiser la totalité du travail que je dois réaliser, pour avoir une idée global du projet :

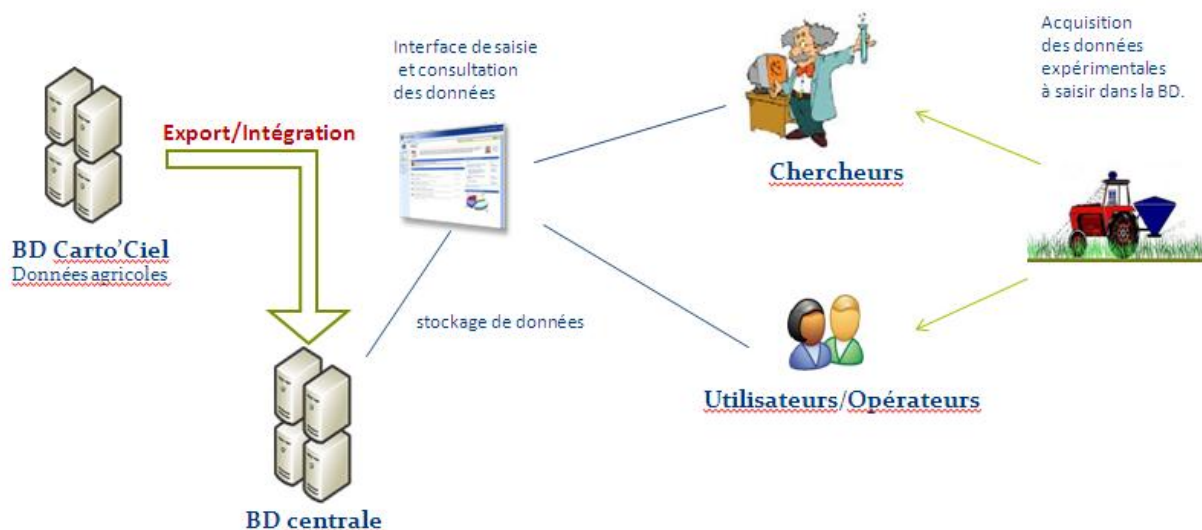
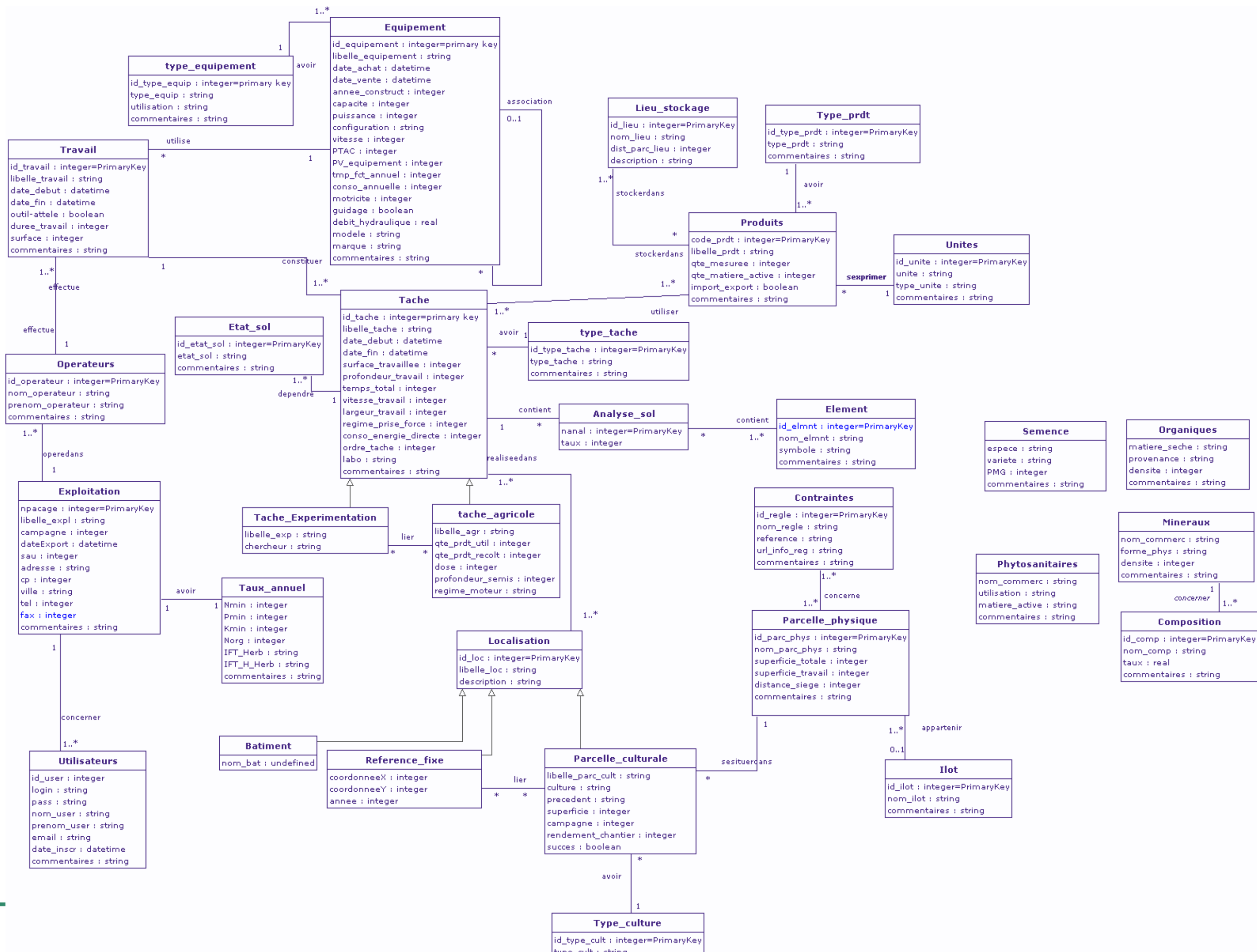


Figure 6: Schéma organisationnel

1. Analyse des données Carto'Ciel

Parmi les objectifs de ce projet, la réalisation d'un outil de conversion qui permet l'intégration des données exportées depuis Carto'Ciel dans la base de données centralisée. C'est pour cela que j'ai réalisé une analyse des données Carto'Ciel et des types de ces données pour faire leur corrélation avec les types des données qui seront stockées dans la base.

Cette analyse se résume à la lecture des fichiers .dbf (Rappel : les exports Carto'Ciel sont des fichiers sous format dbf), lister les champs dont les significations correspondent à celles des champs de la base de données centralisée et l'intégration de cette correspondance de données dans le dictionnaire de données dans le but de les consulter lors de l'écriture du script d'intégration.



2. Base de données

2.1 Modélisation

Dans le but de mener à bien la mise en place de la base de données, j'ai commencé par l'étape de conception, qui est une étape cruciale du fait qu'elle permet de définir la structure de la base de données, avec les tables et les champs qui contiendront les informations saisies.

Au début, j'ai été amenée à faire le recueil du besoin tout en me basant sur l'analyse de l'existant et en vérifiant les données traitées par Carto'Ciel, ainsi que l'analyse faite par Sarah, l'ancienne stagiaire.

Le résultat se résume par la réalisation d'un modèle conceptuel représenté par un diagramme de classes qui contient toutes les tables de la base de données. Ainsi que la réalisation d'autres diagrammes UML, un diagramme de cas d'utilisation pour tracer le fonctionnement global du système, et un diagramme de séquence pour l'exemple d'ajout d'une intervention type expérimentale. (Voir l'annexe)

Des traitements de données ont été faits dans le cadre de ce projet, afin de regrouper le plus de données possible concernant les interventions agricoles et expérimentales.

Le modèle a évolué tout au long de l'avancement du projet au court de réunions de concertations jusqu'à obtenir un diagramme de classes finalisé. Il constitue la structure de la base de données où apparaissent plus en détail, les entités de la base de données et qui concernent les travaux divisés en plusieurs tâches, les équipements utilisés et les opérateurs concernés. Le diagramme met en évidence les produits qu'ils soient des apports ou des imports. Durant l'analyse j'ai pu distinguer deux types de tâches essentielles, la tâche agricole et la tâche expérimentale, que j'ai déclarée comme étant des classes filles de l'entité « Tâche ».

Par exemple : l'épandage est un travail constitué de plusieurs tâches, notamment épandre le produit, transporter les produits à épandre, les épandre dans le sol... ceci requiert du matériel (un tracteur et un épandeur) et bien sûr des opérateurs. Dans ce cas les produits utilisés sont des « entrants ».

Cette Base de données sera accessible par des utilisateurs ayant un compte pour s'authentifier via une interface de saisie et/ou de consultation. Cela permettra une sécurisation et un accès limité aux données.

La conception de la base de données représente une structuration des informations qui seront saisies sur l'Interface Homme-Machine (IHM) qui sera mise en place et qui constitue le système intermédiaire entre l'utilisateur et la base de données. Ainsi chaque entité de la base avec ses propriétés constitue une interface parmi d'autres. Cette correspondance permet de stocker l'ensemble des informations à saisir ou celles que je vais intégrer à partir des exports Carto'Ciel.

Après avoir modélisé la base de données, j'ai utilisé Objectteering Modeler, le logiciel de modélisation orienté objet, pour représenter le modèle conceptuel de données en langage de modélisation UML. Cet outil m'a permis de faire évoluer le modèle tout au long de la période de stage en y apportant plusieurs modifications en s'appuyant sur les remarques et les discussions que j'ai eu avec les membres du projet.

À partir du modèle conceptuel de données, Objecteering permet de générer un modèle physique de données, qui permet de représenter la structure physique et concrète de la base de données.

2.2 Structure

Le modèle conceptuel de données représente une image de la base de données et il contient toutes les entités ainsi que leurs propriétés nécessaires pour le développement du système. Il regroupe six entités principales : Travail, Tache, Tache_experimentale, Tache_agricole, Equipement et Produit. L'entité Travail est reliée à Tache, chose qui explique qu'un travail est constitué d'un ensemble de tâches ou d'interventions. Cependant, une tâche est soit une expérimentation soit une tâche agricole (cette relation est représentée par un lien d'héritage entre l'entité Tâche et les entités Tache_experimentalion et Tache_agricole qui sont également reliées entre elles). Ensuite, on trouve aussi l'entité Equipement, pour les matériels utilisés durant les travaux d'Irstea Montoldre. Equipement est reliée à elle-même par une association, et cela s'explique du fait qu'il peut y'avoir une combinaison de matériels. Par exemple, un tracteur X peut être utilisé avec une charrue pour un travail donné et ce même tracteur X peut être utilisé pour un autre travail et associé à un autre équipement, épandeur par exemple. Cela permet d'associer un travail à un ensemble d'équipements. Ainsi que la partie qui traite les produits utilisés ou obtenus durant une intervention. Ces produits sont caractérisés avec un type, un lieu de stockage, une unité etc.

Ces détails cités concernent les entités principales seulement, alors qu'en réalité ces entités sont reliées à d'autres pour constituer la totalité du modèle.

L'entité Travail est reliée aussi à une entité Operateurs représentant les personnes qui vont effectuer des interventions. Chaque tâche selon son type est affectée à une tâche agricole ou à une expérimentation et toutes les deux constituent un travail.

Exemple : le travail « épandage d'engrais » est constitué de plusieurs interventions :

- Attelage de l'épandeur.
- Chargement de l'épandeur.
- Aller au champ.
- L'épandage.
- Retour du champ.
- Le dételage.

Le travail est effectué à l'aide de deux équipements, le tracteur qui est motorisé et l'épandeur qui ne l'est pas. L'opération utilise de l'engrais qui constitue un produit apporté.

La deuxième phase concerne l'implémentation des interfaces avec le framework Django. Ce framework facilite la création des interfaces homme/machine et ceci, indépendamment du SGBD.

3. Création des interfaces de saisie

Cette étape nécessite d'abord la création d'un projet dans le répertoire de Django, et des applications dont chacune est chargée de traiter une partie du projet.

Ainsi la structure du projet sera comme suit :

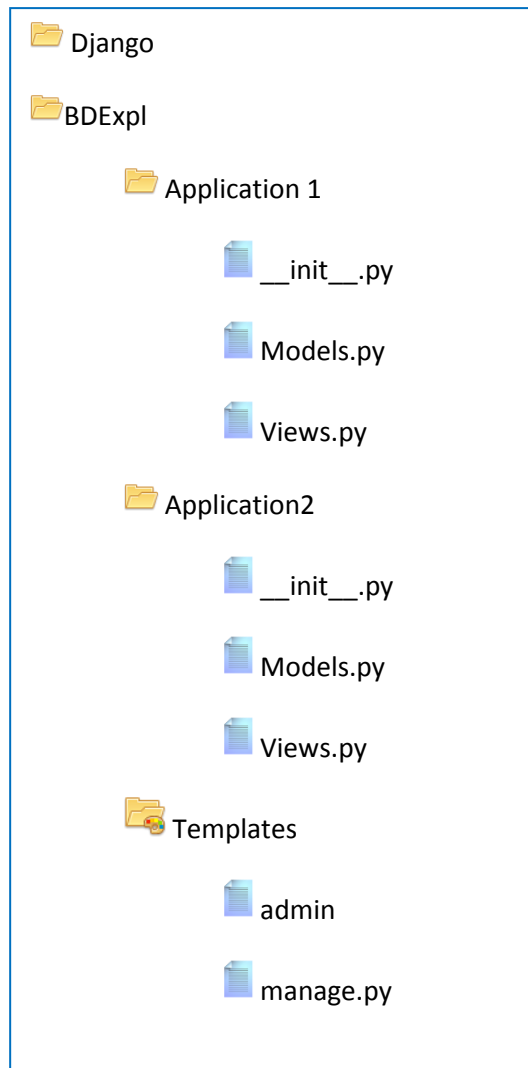


Figure 9: Structure du projet

Le répertoire ‘Templates’ contient tous les fichiers html responsables de l'apparence de l'interface. Pour accéder à l'interface d'accueil du système, les utilisateurs peuvent se connecter grâce à un compte utilisateur qui leur a été attribué par l'administrateur et qui est composé d'un nom d'utilisateur et d'un mot de passe.

En fonction des identifiants saisis, l'utilisateur est dirigé vers la première page pour saisir ou consulter les données de la base.



IRSTEA : BD Exploitation

Mais qui êtes-vous ??!!

Nom d'utilisateur

Mot de passe



Figure 10 : Interface d'authentification

Lorsque l'utilisateur s'authentifie, une session basée sur le protocole HTTP est créée, pour que la session utilisateur n'expire pas lors d'une saisie de données qui se fait en plusieurs étapes. Ceci permettra d'éviter la double-saisie dans la base de données dans le cas où l'utilisateur ressaisira à nouveau ce qu'il avait déjà renseigné.

Après l'authentification, et selon les droits attribués, les utilisateurs sont dirigés soit vers l'interface d'administration soit vers celle de saisie des données d'exploitation.

3.1 Administration du système

L'interface d'administration est dédiée à l'administrateur seulement, il a la possibilité de créer et supprimer des utilisateurs ainsi que gérer leurs droits d'accès. Ceci est la fonction qui différencie un administrateur d'un utilisateur normal. Cependant, pour la gestion des données de la base d'exploitation, j'ai créé une première interface qui contient la totalité des modèles de la base de données (les tables) que j'ai écrites en Python. Cette interface constitue l'intermédiaire entre l'utilisateur et les interfaces de saisie des données dans chacune des tables de la base.

Je note ci-dessous une partie du code permettant d'afficher les modèles Python dans l'interface de saisie.

```
from applications.models import Travail
from django.contrib import admin
admin.site.register (Travail)
```

Figure 11: Code d'ajout des tables dans l'interface d'administration

L'image suivante représente la première interface qui contient les modèles de la base de données et qui permet l'accès aux interfaces de gestion, avec des raccourcis d'ajout ou de modification de chaque modèle. Il y'a aussi l'identifiant de l'utilisateur qui est connecté actuellement avec la possibilité de déconnexion et de modification de son mot de passe. J'ai

aussi listé à droite des modèles, les actions récentes qui ont été réalisées, dans le cas d'un oubli par exemple, l'utilisateur peut revenir rapidement sur la dernière saisie et la modifier.

Administration du site

Applications	
Equipements	 Ajouter  Modifier
Exploitations	 Ajouter  Modifier
Operateurss	 Ajouter  Modifier
Travaills	 Ajouter  Modifier
Type_equipements	 Ajouter  Modifier
Type_taches	 Ajouter  Modifier
Unites	 Ajouter  Modifier
Auth	
Groupes	 Ajouter  Modifier
Utilisateurs	 Ajouter  Modifier
Sites	
Sites	 Ajouter  Modifier

Actions récentes
Mes actions

Type_tache
 Type_tache object
Type_tache
 Type_tache object
Type_tache
 Travail object
Travail
 Equipement object
Equipement
 Type_equipement object
Type_equipement
 Operateurs object
Operateurs
 Exploitation object
Exploitation

Figure 12: Administration du site

L'administration du site permet généralement de gérer, par l'intermédiaire des interfaces, les entrées de la base de données.

Les interfaces d'administration sont gérées à partir du code Python ; dans le fichier `models.py`, qui contient une description de toutes les tables de la base, et le fichier `admin.py` qu'il fallait créer pour gérer l'affichage des données sur les interfaces (spécifier les tables dont l'affichage est nécessaire et personnaliser les formulaires d'administration).

Dans les formulaires d'ajout, je liste les propriétés de la table concernée, suivant le modèle physique de la base de données, et suivant la personnalisation d'affichage que j'ai configuré dans le fichier `admin.py`.

Les libellés des propriétés affichés sur les formulaires sont configurés dans les classes « Meta » avec les modèles dans `models.py`.

Concernant les liaisons entre les différents formulaires, (les clés étrangères dans la base de données), elles sont représentées par des listes déroulantes contenant l'objet créé de la table liée ou si l'objet n'existe pas, il y'a un bouton d'ajout pour une possibilité de créer un nouvel objet.

Les champs de type « date », sont représentés par un calendrier où on peut sélectionner la date à saisir dans ce champ.




Id équipement:	<input type="text" value="9"/>
Libellé:	<input type="text" value="Charrue"/>
Modèle:	<input type="text" value="mod2010"/>
Marque:	<input type="text" value="renault"/>
Configuration:	<input type="text" value="config1"/>
Année de construction:	<input type="text" value="2009"/>
Capacité:	<input type="text" value="21"/>
Puissance:	<input type="text" value="123"/>
Vitesse:	<input type="text" value="120"/>
PTAC:	<input type="text" value="112"/>
PV équipement:	<input type="text" value="123"/>
Temps fonctionnel annuel:	<input type="text" value="120"/>
Consommation annuelle:	<input type="text" value="123"/>
Type de motricité:	<input type="text" value="2RM"/> 
<input checked="" type="checkbox"/> Guidage	
Débit hydraulique:	<input type="text" value="123.0"/>
Id type equip:	<input type="text" value="Type_equipement object"/>  
Commentaires:	<input type="text"/>

Figure 13: Ajout d'un équipement

Afin de personnaliser l'affichage de certains formulaires, j'ai dû ordonner quelques champs et les mettre, par exemple, sur la même ligne, ou encore exclure définitivement quelques-uns du formulaire. Pour cela, j'ai réalisé des modifications supplémentaires sur le fichier admin.py. Je donne l'exemple de la table « Tache » et « Type_tache » :

```

from applications.models import Tache_agricole
from django.contrib import admin

class Tache_agriInline(admin.StackedInline):
    #classe pour appliquer des modifications
    #d'affichage sur la classe tache_agr
    fieldsets = [
        (None, {'fields': ['id_tache_agr', 'qte_prdt_util', 'qte_prdt_recolt',
                          'profondeur_semis', 'regime_moteur']}), # paramettrer les labels des champs
    ]
    model = Tache_agricole # sur le model "tache_agricole"
    extra = 0

from applications.models import Tache_experimentation # importer "tache_exp" de models.py
from django.contrib import admin #importer admin du package contrib

class Tache_expInline(admin.StackedInline):
    fieldsets = [
        (None, {'fields': ['id_tache_exp', 'chercheur']}),
    ]
    model = Tache_experimentation
    extra = 0

```

Figure 14: Administration de l'interface d'ajout d'une intervention

Les dernières lignes de ce code Python permettent de regrouper deux tables liées dans la même interface de saisie, afin de faciliter le fait de renseigner à la fois la tâche ou l'intervention et son type (agricole ou expérimentale) avec des cases à cocher pour choisir le type de l'intervention. Ceci sera réalisé aussi pour d'autres formulaires cités plus loin. On obtient alors le résultat suivant :

Id type:

Tâche agricole Tâche expérimentale

Taches

Tache: Tache object

Libellé:

Surface travaillée:

Profondeur de travail:

Temps total:

Vitesse de travail:

Largeur de travail:

Régime de prise en force:

Consommation d'énergie directe:

Ordre de tâche:

Laboratoire:

Id travail:

Commentaires:

[+ Ajouter un objet Tache supplémentaire](#)

Figure 15 : Représentation de plusieurs classes dans un même formulaire

3.2 Interface de saisie

L'opération de saisie se fait après identification des différents utilisateurs, et vu l'importance de la quantité des données à saisir, il y a plusieurs interfaces qui seront utilisées par les utilisateurs du système. Je citerai les plus importantes par rapport à la gestion des activités expérimentales.

Cette gestion correspond au fait de saisir une intervention par type (agricole ou expérimentale ou les deux en même temps), il suffit de cocher le type d'intervention et de saisir les informations, d'abord, concernant la tâche générale, puis les informations du type de

tâche. Ainsi, lors de la consultation, l'utilisateur peut vérifier directement l'intervention type à laquelle il veut accéder.



Figure 16 : Choix du type de l'intervention

Concrètement, pour créer des interfaces de ce type, il faut créer des formulaires et des vues et faire la liaison entre ces deux concepts pour obtenir un affichage correct en termes de visualisation et de gestion de contenu.

Les vues correspondent à un type de page web dans l'application Django qui sert à une fonction précise et possède un Template spécifique. Elles sont représentées par de simples fonctions Python. La conception des vues commence par la création d'une structure d'URL située dans `urls.py` pour chaque application, et puis saisir la fonction de la vue dans le fichier `views.py`.

4. Script d'intégration

En suivant les phases de mon projet, et suite à la partie d'analyse des données Carto'Ciel, je serai amenée à intégrer ces données dans la BD Exploitation.

Pour cela, j'ai commencé à traiter la situation. Vu que l'objectif consiste à intégrer des données exportées d'une source différente de la destination (des fichiers de données .dbf vers une base de données PostgreSQL), j'ai commencé d'abord par faire la correspondance entre les champs des fichiers .dbf et ceux des tables de la BD Exploitation, parce que le besoin d'export concerne certaines données et pas d'autres. Cependant, il a fallu écrire des requêtes permettant de filtrer les champs des exports, chose que j'ai réalisée : j'ai regroupé les requêtes dans un fichier en attendant de programmer l'intégration des résultats des requêtes.

Par la suite, j'ai pu créer un petit système qui lit un fichier avec l'extension dbf, et qui affiche le résultat de la requête. En effet la fonction principale de ce programme reçoit le nombre de colonnes en paramètres d'entrée, puis les intitulés de ces dernières et elle retourne le résultat en affichant toutes les données des colonnes spécifiées.

Jusqu'à présent j'ai réalisé ce programme en attendant les exports qui seront fournis par Sonéo, la société qui a développé Carto'Ciel. Dès que ses données me seront fournies je créerai l'outil d'intégration qui permettra d'importer ces exports dans la base de données centrale.

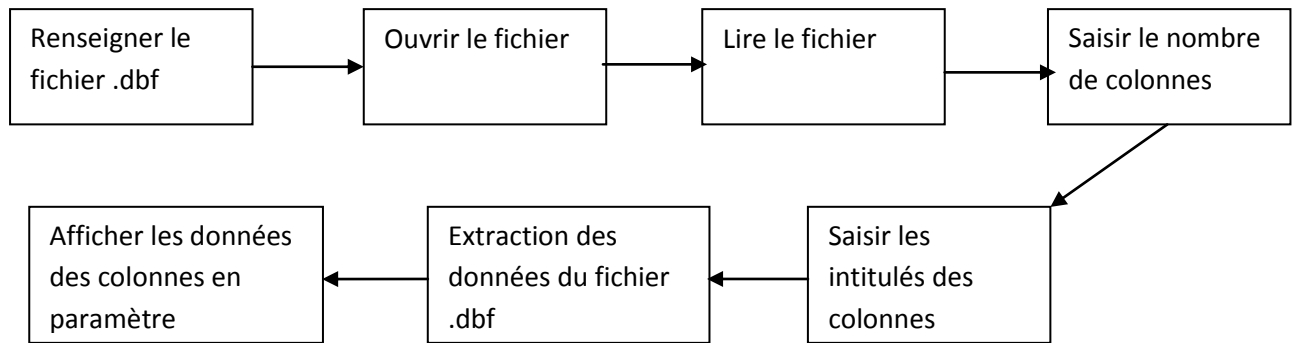


Figure 17: Schéma simplifié du fonctionnement du script

N.B : La continuité de cette partie fait partie des perspectives à réaliser par la suite.

Conclusion et perspectives

Le système de gestion de l'information pour les activités expérimentales du site de Montoldre constitue un complément de poids important pour le système existant qui stocke les interventions agricoles. Ce système a la possibilité d'accueillir autant plus de métadonnées intégrant les expérimentations et leurs résultats. La mise en place de cet outil reste une tâche assez complexe et exigeante.

Durant le stage, j'ai rencontré des difficultés, notamment la compréhension de termes agricoles, l'utilisation de Carto'Ciel ainsi que le développement avec le langage Python que je n'avais jamais étudié. Grâce à l'accompagnement des spécialistes d'Irstea, j'ai pu surmonter et trouver des solutions aux difficultés au fur et à mesure de l'avancement de mon stage.

Ce stage m'a permis de réaliser un projet dans son ensemble avec une méthodologie de travail assez exigeante. Il m'a permis aussi de découvrir les différentes étapes de vie d'un projet et l'intérêt de chacune de ces étapes. J'ai été confrontée particulièrement à l'analyse des données et la conception de la base de données, une activité qui demande beaucoup de rigueur, d'organisation et de compréhension de la demande afin de bien la transcrire dans le modèle.

C'est une expérience autant bien sur le plan professionnel que personnel. J'ai pu développer mon côté relationnel et, aiguïser mon esprit critique mais aussi exprimer mon goût pour le contact, la prise d'initiative et surtout le travail en équipe.

Au cours de ces trois mois, j'ai pu apprécier l'organisation, les enjeux et les moyens liés à un système d'information de grande taille. Ce stage m'a permis de découvrir et de comprendre les métiers rencontrés au sein d'Irstea et les relations entre les interventions des différents agents (chercheurs, ingénieurs, opérateurs,...). J'ai de plus eu la chance de découvrir de nouveaux outils comme Django et Objectteering qui sont d'une puissance remarquable et d'organiser mon travail selon une démarche type projet, cela m'a également appris à gérer un planning et des priorités.

Il me reste deux mois pour finaliser le projet. Durant cette période, je souhaite créer d'autres vues et formulaires Django, développer le script d'intégration des données depuis Carto'Ciel vers la BD d'exploitation et réaliser des tests fonctionnels pour m'assurer du bon fonctionnement du système, et qu'il corresponde parfaitement aux besoins des utilisateurs. Je souhaite qu'à mon départ le système soit utilisé par l'entreprise et que malgré la complexité du projet l'outil soit suffisamment simple afin que les opérateurs se l'approprient. Il restera ensuite une partie que je ne pourrais pas aborder dans ce stage : l'intégration des données géographiques ainsi que des méta-données (références vers les fichiers bruts) dans la base.

Webographie et Bibliographie

- [1]: <http://www.irstea.fr/>
- [2]: <http://blog.postgresql.fr/index.php?post/drupal/216>
- [3]: <http://support.objecteering.com/objecteering6.1>
- [4]: <http://postgresql.developpez.com/articles/pgdebutants/#L2.1.3>
- [5] : <http://docs.postgresql.fr/8.0/ddl-constraints.html>
- [6] : <http://www.developpez.net/forums/d49867/java/general-java/jdbc/jdbc-lecture-fichier-dbf-via-jdbc/>
- [7]: <http://docs.django-fr.org/intro/tutorial01.html#le-serveur-de-developpement>
- [8]: <http://tutos-django.com/2011/10/24/bienvenue/>
- [9] : <http://blog.pilotsystems.net/2011/juin/utilisation-south-application-django-migrations-versions-base-donnees-schema>
- [10] : Gérard SWINNEN : Apprendre à programmer avec Python 3(Édition EYROLLES).
- [11] : Le rapport de stage d’Amine TAIB.
- [12]: Le rapport de Sarah BONIS (2011).
- [13]: le rapport de MFE de Thomas PACAUD.

Glossaire

Irstea : Institut de Recherche en Sciences et Technologies pour l'Environnement et l'Agriculture.

EPST : un Établissement Public à caractère Scientifique et Technologique

TSCF : une unité de recherche de Technologie et Système d'information pour les agro-systèmes Clermont-Ferrand.

SIG : Système d'Information Géographique.

TEAM : Équipe des Technologies de la mobilité pour la sécurité et la qualité du travail des machines agri-environnementales.

Carac'Terre : Équipe des Technologies pour la caractérisation du sol appliquées aux éco et agro-systèmes, Sciences-technologies et procédés d'épandage.

COPAIN : Équipe Systèmes d'information communicants et agri-environnementaux

BDD : Base de Données.

UML : Unified Modeling Language, est un langage de modélisation des Base de données.

Framework : Un framework est un ensemble d'outils et de composants logiciels. Il est conçu en vue d'aider les programmeurs dans leur travail.

C#, C++, Java : sont des langages de programmation informatique. (C sharp, C plus plus, Java).

PHP : un langage de programmation Web.

SQL : Structured Query Language, en français langage de requête structurée est un langage informatique normalisé servant à effectuer des opérations sur des bases de données

Open Source : La désignation open source s'applique aux logiciels dont la licence respecte des critères précisément établis par l'Open Source Initiative, c'est-à-dire la possibilité de libre redistribution, d'accès au code source et aux travaux dérivés.

HTML : L'Hypertext Markup Language, est le format de données conçu pour représenter les pages web.

URL : Une URL (Uniform Resource Locator) est un format de nommage universel pour désigner une ressource sur Internet.

ORM : Mapping Object Relationnel, est une technique de programmation informatique qui crée l'illusion d'une base de données orientée objet à partir d'une base de données relationnelle en définissant des correspondances entre cette base de données et les objets du langage utilisé. On pourrait le désigner par «correspondance entre monde objet et monde relationnel».

DBF : sont des fichiers d'une base de données DBase.

SGBD : un système de gestion de base de données est un logiciel système destiné à stocker et à partager des informations dans une base de données, en garantissant la qualité, la pérennité et la confidentialité des informations, tout en cachant la complexité des opérations.

IntegerField : le type entier des champs d'une base de données.

CharField : le type chaîne de caractères des champs d'une base de données.

HTTP : HyperText Transfer Protocol, est un protocole de communication client-serveur développé pour le World Wide Web. Il peut fonctionner sur n'importe quelle connexion fiable.

Template : est un moyen de séparer le fond (le contenu informationnel) de la forme (la manière dont il est présenté). Très utilisé dans la conception de sites webs, un template agit comme un modèle (une structure) dans lequel seuls certains éléments sont modifiables (le contenu et le style).

