

Annexe 1 : Extrait de cahier d'épandage de l'exploitation « La Hautière »

CAHIER D'EPANDAGE

Année culturale 2006/2007

Renseignements sur la parcelle		Renseignements sur les apports d'azote								Renseignements sur les effluents				
Référence de la parcelle : numéro îlot-numéro de parcelle dans l'îlot surface	Espèce cultivée implantée Rendement prévu Rendement réalisé	Date des divers apports	Libellé, nature du produit épandu (minéral ou organique)	Quantité de produit épandu / ha	Quantité totale épandue	Teneur en azote minérale estimée	Apport total d'azote organique	Apport total d'azote minéral	Dose d'azote	Origine de l'effluent E=exploitation, T=tiers	Délai d'enfouissement	Teneur en azote mesurée	Traitement anti-odeur (oui / non)	
				kg / ha	t	kg / t	kg	kg	kg / ha		immédiat / 12h / 24h	kg / t		
Sur la parcelle 1 de l'îlot 1 (5 ha), culture de betterave avec un objectif de 60 tonnes. Lors du premier épandage, réalisé en septembre, il y a eu apport de 30 tonnes de fumier / ha, la teneur moyenne de ce fumier a été mesurée à 5,5 kg d'azote par tonne. La fertilisation est complétée en avril par un second épandage avec un apport d'ammonitrate 27 dosé à 55 unités par hectare et en juin par un apport de fumier de 20 tonnes par hectare avec une teneur d'azote mesurée de 6,5 kg par tonne.														
1-1 5 ha	Betteraves 60 t 65 t	1	5-sept.	Fumier	30 000 kg / ha	150 t		825		165	E	24	5,5	Non
		2	15-avr.	ammo 27	200 kg / ha	1,02 t	270		275	55				
		3	3-juin	Fumier	20 000 kg / ha	100 t		650		130	T		6,5	Oui
		4												
							sous-total	1475	275					
Sur la parcelle 2 de l'îlot 1 pour une surface 7,5 ha de blé (objectif 80 quintaux), la fertilisation a été réalisée en 3 apports d'ammonitrate 27 : 50 kg / ha en février, 90 kg / ha en mars et à nouveau 50 kg / ha en avril.														
1-2 7,5 ha	Blé 80 q 90 q	1	15-févr.	ammo 27	185 kg / ha	1,38 t	270		373	50				
		2	9-mars	ammo 27	330 kg / ha	2,5 t	270		675	90				
		3	20-avr.	ammo 27	185 kg / ha	1,38 t	270		373	50				
		4												
							sous-total	0	1 421					
L'îlot 2 (une seule parcelle) est un pâturage de 2 ha. Elle a reçu en mars un apport de 40 t de purin par ha, dosé à 0,4 kg d'azote (mesuré), en avril un apport de 13/8/24 pour une dose de 45 kg / ha, suivi de deux autres passages avec de l'ammonitrate 27 à la dose de 45 kg / ha														
2-1 2 ha	Pâturage	1	5-mars	Purin	40 000 kg / ha	80 t		32		16	E		0,4	Oui
		2	7-avr.	13/8/24	350 kg / ha	0,7 t	130		90	45				
		3	2-mai	ammo 27	170 kg / ha	0,33 t	270		90	45				
		4	8-juin	ammo 27	170 kg / ha	0,33 t	270		90	45				
							sous-total	32	270					

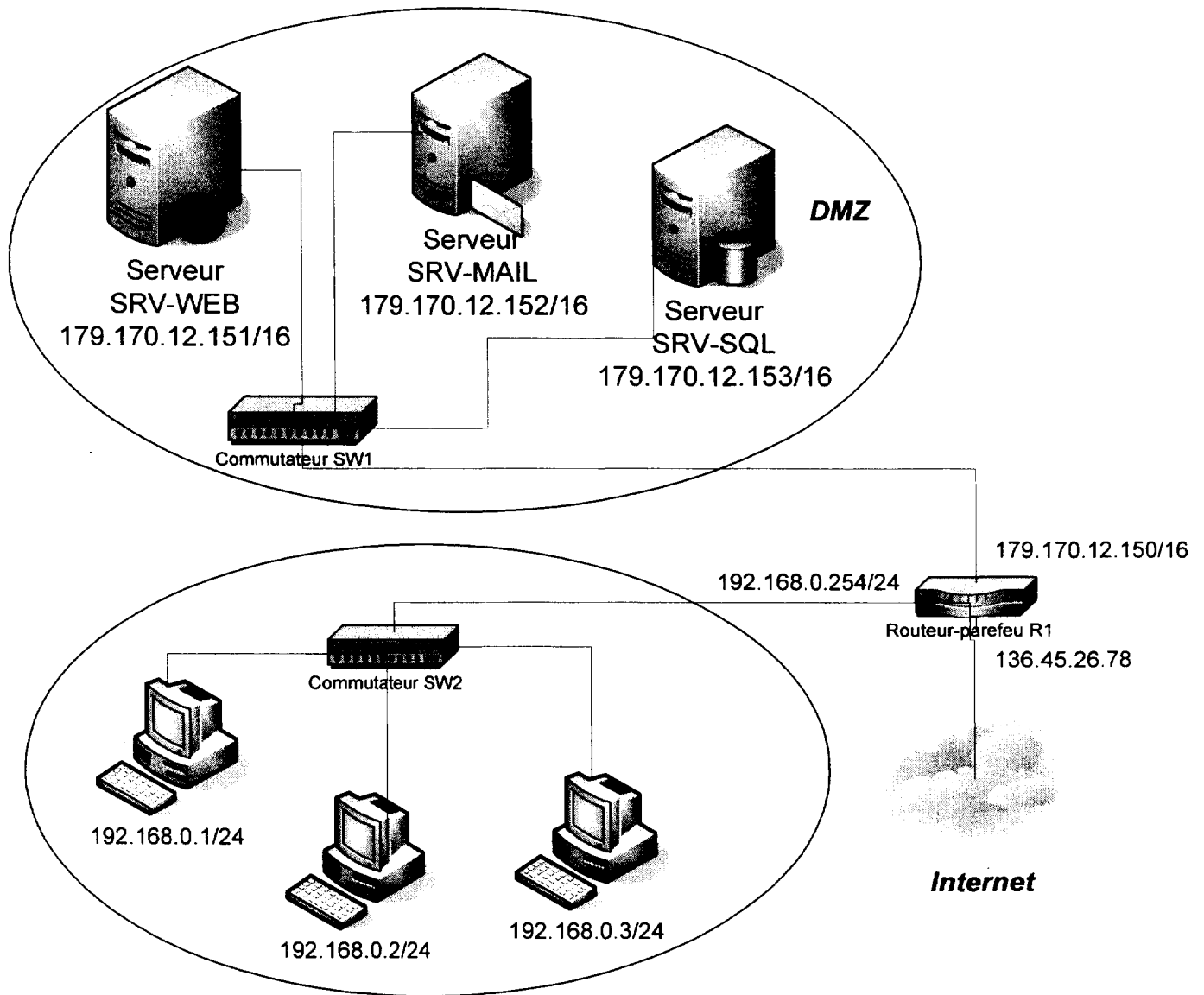
Total des surfaces épandues en ha

organique	minéral
12	33,5

Total des apports d'azote en kg

organique	minéral
1 507	1 966

Annexe 2 : Schéma partiel du site informatique de la Chambre d'Agriculture



Annexe 3 : Extrait du schéma relationnel de la base de données « TraitementPhyto »

Le dossier 3 est indépendant du dossier 1 : l'extrait du schéma relationnel ne représente pas une solution au dossier 1.

PRODUITPHYTOSANITAIRE(id, libelle)

Clé primaire : id

EXPLOITATION(id, nomExploitant, melExploitant)

Clé primaire : id

PARCELLE(id, dateSemis, dateRecoltePrevue, surface, idExploitation, idEspeceCultivee)

Clé primaire : id

Clé étrangère : idExploitation en référence à id de EXPLOITATION

Clé étrangère : idEspeceCultivee en référence à id de ESPECECULTIVEE

TRAITEMENT(id, typeTraitement, dosageTraitementSemence, idParcelle, idProduitPhytosanitaire)

Clé primaire : id

Clé étrangère : idParcelle en référence à id de PARCELLE

Clé étrangère : idProduitPhytosanitaire en référence à id de PRODUITPHYTOSANITAIRE

PULVERISATION(id, datePulverisation, dosage, idTraitement)

Clé primaire : id

Clé étrangère : idTraitement en référence à id de TRAITEMENT

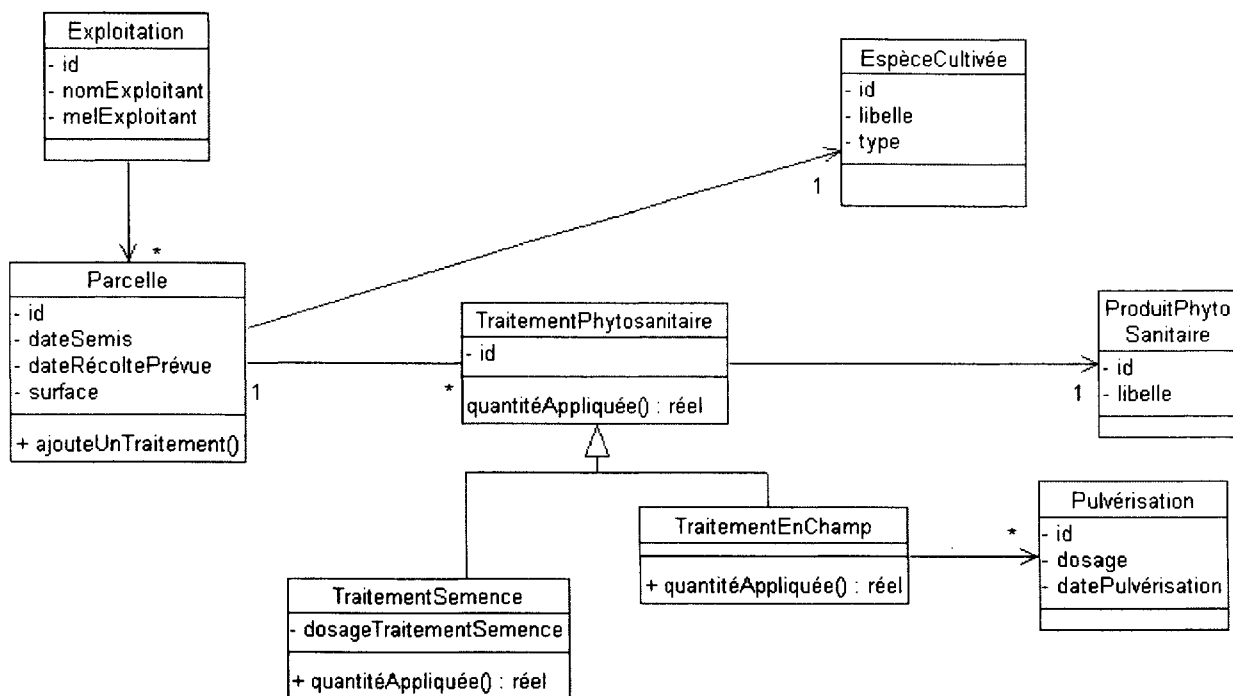
ESPECECULTIVEE(id, libelle, type)

Clé primaire : id

Dans la relation TRAITEMENT :

- Si le traitement est en semence (produit mélangé au semis), le champ *typeTraitement* prend la valeur 's' et le champ *dosageTraitementSemence* prendra une valeur.
- Si le traitement est en champ (plusieurs pulvérisations), le champ *typeTraitement* prend la valeur 'c' et le champ *dosageTraitementSemence* prendra la valeur nulle. Dans ce cas seulement, plusieurs occurrences de la relation PULVERISATION pourront être associées à ce traitement.

Annexe 4 : Diagramme partiel des classes métiers



Remarque : Les constructeurs, accesseurs et paramètres des méthodes ne sont pas présentés sur ce diagramme.

Annexe 5 : Description textuelle des classes métiers

Classe Exploitation

Attributs privés :

id : Chaîne
 nomExploitant : Chaîne
 melExploitant : Chaîne
 lesParcelles : Collection de Parcelle

Méthodes publiques :

Fonction getLesParcelles() : Collection de Parcelle
 // accesseur sur l'attribut privé lesParcelles

FinClasse

Remarque : La classe technique Collection est décrite en annexe 6.

Classe EspèceCultivée

Attributs privés :

id : Chaîne
 libellé : Chaîne
 type : Chaîne

FinClasse

Classe ProduitPhytosanitaire

Attributs privés :

id : Chaîne
libellé : Chaîne

FinClasse

Classe Parcelle

Attributs privés :

id : Chaîne
dateSemis : Date
dateRécoltePrévue : Date
surface : Réel
lEspèceCultivée : Espèce
lesTraitementsPhytosanitaires : Collection de TraitementPhytosanitaire

Méthodes publiques :

Fonction getSurface() : Réel
// accesseur sur l'attribut surface
Fonction getLesTraitementsPhytosanitaires() : Collection de TraitementPhytosanitaire
// accesseur sur l'attribut lesTraitementsPhytosanitaires
Procédure ajouteUnTraitement(unTraitementPhytosan : TraitementPhytosanitaire)
// permet d'ajouter un traitement à la parcelle courante

FinClasse

Classe TraitementPhytosanitaire

Attributs privés :

id : Chaîne
laParcelleCultivée : Parcelle
leProduitPhytosanitaire : ProduitPhytosanitaire

Méthodes publiques :

TraitementPhytosanitaire(unIdTraitementPhytosan : Chaîne, uneParcelleCultivée : Parcelle,
unProduitPhytosanitaire : ProduitPhytosanitaire)
// constructeur de la classe
Fonction abstraite quantitéAppliquée() : Réel
Fonction getLaParcelleCultivée() : Parcelle
// accesseur sur l'attribut laParcelleCultivée

FinClasse

Classe **TraitementSemence** hérite de **TraitementPhytosanitaire**

Attributs privés :

dosageTraitementSemence : Réel // indique la quantité par unité de surface, par ex. 200 (kg/ha)

Méthodes publiques :

TraitementSemence(unIdTraitementPhytosan : Chaîne, uneParcelleCultivée : Parcelle, unProduitPhytosanitaire : ProduitPhytosanitaire, unDosage : Réel)

// constructeur à écrire

Fonction quantitéAppliquée() : Réel

// la quantité est calculée en effectuant le produit de la surface de la parcelle par le dosage

// du traitement concerné

FinClasse

Remarque : Le constructeur s'utilise de la manière suivante :

Tr : TraitementSemence

Tr ← new TraitementSemence(< arguments du constructeur >)

Classe **TraitementEnChamp** hérite de **TraitementPhytosanitaire**

Attributs privés :

lesPulvérisations : Collection de Pulvérisation

Méthodes publiques :

TraitementEnChamp(unIdTraitementPhytosan : Chaîne, uneParcelleCultivée : Parcelle, unProduitPhytosanitaire : ProduitPhytosanitaire, desPulvérisations : Collection de Pulvérisation)

// constructeur à écrire

Fonction quantitéAppliquée() : Réel

// la quantité est calculée en effectuant le cumul des quantités pulvérisées

Fonction getLesPulvérisations() : Collection de Pulvérisation

// accesseur sur l'attribut lesPulvérisations

FinClasse

Classe **Pulvérisation**

Attributs privés :

id : Chaîne

dosage : Réel

// indique la quantité par unité de surface, par ex. 200 (kg/ha)

datePulvérisation : Date

Méthodes publiques :

Pulvérisation(unId : Chaîne, uneDate : Date, unDosage : Réel)

// constructeur de la classe Pulvérisation

Fonction getDosage() : Réel

// accesseur sur l'attribut dosage

FinClasse

Remarque : Le constructeur s'utilise de la manière suivante :

pv : Pulvérisation

pv ← new Pulvérisation(<arguments du constructeur>)

Annexe 6 : Description textuelle des classes techniques

Classe Passerelle

Méthodes publiques à portée classe :

Fonction `getProduit(unIdProduit : Chaîne) : ProduitPhytosanitaire`

// retourne l'objet de la classe `ProduitPhytosanitaire` dont l'identifiant est passé en paramètre

Fonction `getParcelle(unIdParcelle : Chaîne) : Parcelle`

// retourne l'objet de la classe `Parcelle` dont l'identifiant est passé en paramètre

Fonction `chargerLesPulvérisations(unIdTraitement : Chaîne) : Collection de Pulvérisation`

// Instancie et retourne une collection d'objets de la classe `Pulvérisation` correspondant aux pulvérisations

// concernant le traitement d'identifiant `unIdTraitement`, à partir des données lues dans la base de données

Fonction `chargerLeTraitement(unIdTraitement : Chaîne) : TraitementPhytosanitaire`

// Instancie et retourne un objet de la classe `TraitementPhytosanitaire` correspondant au traitement

// d'identifiant `unIdTraitement`, à partir des informations lues dans la base de données

FinClasse

Classe Collection de <nom de la classe>

Méthodes publiques

Fonction `cardinal() : Entier`

// Renvoie le nombre d'objets de la collection

Fonction `obtenirObjet(unIndex : Entier) : Objet de la classe`

// Retourne l'objet d'index `unIndex`, le premier objet de la collection a pour index 1

Procédure `ajouter(unObjet : Objet de la classe)`

// Ajoute un objet à la collection

FinClasse

Pour instancier une collection :

`uneCollection : Collection de <classe>`

`uneCollection ← new Collection() de <classe>`

Pour parcourir par itération les éléments d'un objet `Collection`

Pour chaque <objet> dans <collection> faire

// instructions avec <objet>

`FinPour`

Classe JeuEnregistrements

Méthodes Publiques

`JeuEnregistrements(chaineSQL : Chaîne) // Constructeur, positionne le curseur sur le premier
// enregistrement`

Procédure `suivant()`

// avance le curseur sur l'enregistrement suivant

Fonction `fin() : Booléen`

// indique si la marque de fin est atteinte

Fonction `getValeur(nomChamp : Chaîne) : Variant`

// renvoie la valeur du champ `nomChamp` de l'enregistrement courant. Variant est un type

//générique pouvant contenir tout type de valeur. On peut utiliser l'opérateur « + » pour concaténer

// des valeurs ou variables de type chaîne ou variant.

Procédure `fermer()`

// ferme le curseur et libère les ressources

FinClasse

Remarque : Le constructeur s'utilise de la manière suivante :

`jeu : JeuEnregistrements`

`jeu ← new JeuEnregistrements(<chaîne SQL >)`

Annexe 7 : Algorithme de la méthode chargerLeTraitement() de la classe Passerelle

```
Publique Statique Fonction Passerelle
:: chargerLeTraitement(unIdTraitement : Chaîne) : TraitementPhytosanitaire

Var
  texteReq : Chaîne
  jeuTraitements : JeuEnregistrements
  leProduit : ProduitPhytosanitaire
  laParcelle : Parcelle
  traitementAReturner : TraitementPhytosanitaire // objet à retourner

Début

  traitementAReturner ← null

  texteReq ← " Select * From Traitement tp Where tp.id = '"
  texteReq ← texteReq + unIdTraitement + "'"

  jeuTraitements ← new JeuEnregistrements(texteReq)
    // Exécution de la requête SQL

  si NON jeuTraitements.fin() alors
    // Il y a un traitement associé

    leProduit ← getProduit(jeuTraitements.getValeur("idProduitPhytosanitaire"))
    laParcelle ← getParcelle(jeuTraitements.getValeur("idParcelle"))

    -- Partie à compléter sur la copie (cf. question 3.9) --

    // instructions créant un objet de la classe TraitementSemence
    // ou de la classe TraitementEnChamp, retourné par la fonction

  fsi

  jeuTraitements.fermer()

  retourner traitementAReturner // objet retourné

Fin
```