

Base de données biogéographiques sur Internet

SYRFID

Presentation Documentation Serveur cartographique Lien

SYRFID v3.0

SYRphidae of France Interactive Data

Données interactives sur les observations des espèces
de DIPTERA SYRPHIDAE et MICRODONTIDAE en France

INP ENSAT

Site réalisé par Mlle Gautier, Mr Gloaguen, Mr Granjon et Mr Thieu dans le cadre du projet de fin d'étude du DESS SIGMA, en collaboration avec Mr Monteil, Mr Sarthou et Mr Souques.

MapServer
PostgreSQL
PostGIS

Référence à utiliser pour citer SYRFID
SARTHOU J.P., MONTEIL C. et DUSSAIX C., 2005
SYRFID v3.0 : Syrphidae of France Interactive Data [On-Line URL : <http://syrfid.ensat.fr/>].
Dernière mise à jour : 20 février 2005
© SYRFID 2005. INP/ENSAT

GAUTIER Bénédicte
GLOAGUEN Jérémy
GRANJON Ludovic
THIEU Vincent



Responsables : Claude MONTEIL et Marc SOUQUES

Fevrier 2005

SOMMAIRE

I. INTRODUCTION	2
1.1 Cadre du projet	2
1.2 Contexte du projet	2
1.3 Objectif du projet	2
1.4 L'application	2
II. PRESENTATION DE L'EXISTANT	3
2.1 L'interface du site Internet : SYRFID	3
2.1.1 La première version	3
2.1.2 La deuxième version	3
2.2 La base de données	4
2.2.1 Le Modèle Conceptuel de Données	4
2.2.2 Contenu de la base de données Excel	5
III. CHOIX DE LA CONFIGURATION	8
3.1 Proposition de configuration	8
3.2 Autre configuration : notre choix	9
IV. LA NOUVELLE VERSION DU SITE INTERNET	10
4.1 Restructurer la base de données	10
4.1.1 Nouveau Modèle Conceptuel de Données	10
4.1.2 Nouvelles tables et vues	11
4.2 Interrogation de la base de données, les interfaces	12
4.2.1 Interface cartographique	12
4.2.2 Autres fonctionnalités	13
4.3 Administration de la base	14
V. BILAN	15

Un manuel d'utilisation et un guide de maintenant sont également rédigés.

I. INTRODUCTION

1.1 Cadre du projet

Dans le cadre de la formation S.I.G.M.A (Science de l'information géoréférencée pour la maîtrise de l'environnement et de l'aménagement du territoire), nous avons réalisé un projet de 5 semaines (février) pour le laboratoire Biodiversité dans les agroécosystèmes de l'Ecole Supérieure Agronomique de Toulouse nous permettant de mettre en pratique et d'approfondir nos connaissances.

1.2 Contexte du projet

Ce laboratoire dispose d'un vaste ensemble d'observations spatiales et temporelles sur une famille d'insectes (les Syrphidés) sous forme de triplets (département, date, observateur) pour chacune de près de 500 espèces (rassemblées dans 81 genres), chaque observation étant rattachée à une source bibliographique. Chaque espèce possède 2 attributs caractérisant son degré de vulnérabilité, en France d'une part, et en Europe d'autre part. En outre, certaines espèces sont associées à une macro-photographie permettant de mieux visualiser à quoi elle ressemble.

Les personnes travaillant sur les Syrphidés, souhaitent pouvoir effectuer diverses opérations de visualisation et de sélection (temporelle, spatiale, biologique) sur ces données, a priori en local, mais si possible également sur Internet. La mise à jour de cette base de données doit aussi être prise en compte, en version locale exclusivement avec une fréquence relativement faible (semestriellement).

Deux versions de cette application existent. La première est opérationnelle tandis que la seconde a fait l'objet d'un prototype apportant tout un ensemble de fonctions nouvelles à travers une page d'interface développée en JavaScript

1.3 Objectif du projet

Il s'agira de développer une nouvelle version permettant :

- D'effectuer une génération dynamique des cartes; ceci supposera de reprendre le MCD et revoir la structure de la base de données
- D'assurer si possible toutes les fonctions de la version 2 avec une ergonomie analogue
- D'ajouter de nouvelles fonctions de sélection cartographiques (sélection de tout un ensemble de département par dessin d'un rectangle à la souris et/ou par clics successifs sur certains départements)
- De permettre une mise à jour sécurisée des données (ce qui est une limite de la version Excel actuelle)

1.4 L'application

- a) La première partie permet à tout public de consulter (lecture seule) de manière distante les données d'une base de données relationnelle via le Web. La programmation en HTML, PHP, JavaScript et l'utilisation dans PHP d'une bibliothèque de génération de graphiques vectoriels pour remplacer ArcView et générer les cartes sont envisagées.
- b) La deuxième partie permet aux seules personnes habilitées d'administrer en local à l'ENSAT de manière sécurisée les données (ajout, modification, suppression, consultation sur critères,...) d'une base de données via une interface à définir (Web ou Access). Une exportation des données de la base locale vers le format Excel est à prendre en compte. Un couplage avec ArcView pourra éventuellement être envisagé pour des traitements spatiaux additionnels.

Avant de commencer l'application, nous avons dû dans un premier temps analyser, étudier le contenu de l'existant au niveau de la base de données et des sites web réalisés ultérieurement puis dans un deuxième temps choisir la configuration qu'il était préférable d'utiliser pour ce projet (Access, MySql ou autres). Suite à cette étude et ce choix réalisés, nous avons pu débiter le projet dont l'objectif est de développer une nouvelle version du site Internet.

II. PRESENTATION DE L'EXISTANT

2.1 Le site Internet : SYRFID (acronyme de SYRphidae of France Interactive Data)

2.1.1 La première version

Une première version de cette application est opérationnelle: <http://syrfid.ensat.fr>

SYRFID vol.2
Syrphidae of France Interactive Data
volume 2

**Données interactives sur les observations des espèces de
DIPTERA SYRPHIDAE et MICRODONTIDAE en France**

site développé par [Jean-Pierre SARTHOU](#) et [Claude MONTEIL \(INP/ENSAT, Toulouse\)](#)
en collaboration avec [Cyrille DUSSAIX](#)

[Présentation et guide d'utilisation](#)
?

[Bibliographie](#)

[Galerie photographique](#)
NOUVEAU

Accès aux observations :

- [par genre/espèce](#)
- [par département](#)
- [par cartes globales](#)

La base de données est gérée sous forme de classeur Excel, à partir duquel sont générés les fichiers HTML permettant d'assurer une visualisation sous 2 formes (tabulaire et cartographique) et une navigation entre espèces et départements.

La partie cartographique apparaît sous forme d'images JPEG générées via un couplage entre Excel (en langage VBA) et ArcView (en langage Avenue).

Ces cartes sont cliquables grâce à l'utilisation de balises MAP en langage JavaScript.

Les limites de cette version sont de deux ordres :

- 1/ la structure du serveur Internet est totalement statique (toutes les cartes potentielles pré-existent sous forme d'image JPEG)
- 2/ les fonctions demeurent limitées (pas de zoom, pas de sélection spatiale, pas de sélection par genre, ...etc.).

2.1.2 La deuxième version

Une deuxième version a fait l'objet d'un prototype apportant tout un ensemble de fonctions nouvelles à travers une page d'interface développée en JavaScript : [http://syrfid.ensat.fr/ Syrfid.htm](http://syrfid.ensat.fr/Syrfid.htm).

CARTES D'EFFECTIFS

GENRES ESPECES

Période d'observation :

globale (observé au moins 1 fois)
 avant 1970 et réobservé depuis
 avant 1970 seulement, pas depuis
 depuis 1970 seulement, pas avant

CARTES DE REPARTITION

Genre (0/81) Espèce (0/0)

—> choisir un genre <— —> choisir une espèce <—

Critères de choix de l'espèce :

— par nom d'espèce — — par degré de vulnérabilité —

en France en Europe

Espèces observées au moins une fois

Nombre d'espèces

- 0
- 1 - 10
- 11 - 50
- 51 - 100
- 101 - 200
- plus de 200

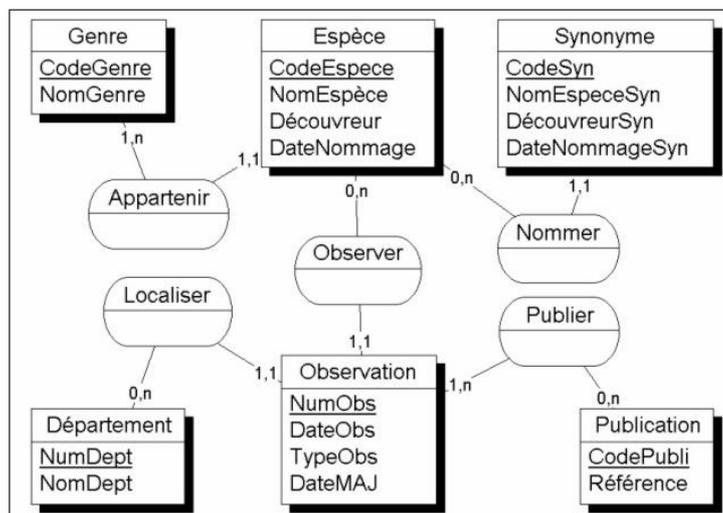
Détail des observations

Zones biogéographiques

Ce prototype est partiellement associé à la version 1 car il agit comme un tableau de bord permettant de renvoyer aux tableaux et cartes de celle-ci. Par contre, les tableaux ou cartes ne permettent pas de revenir sur la page centrale de la version 1 en l'état actuel des choses.

2.2. La base de données

2.2.1 Modèle Conceptuel de Données



2.2.2 Contenu de la base de données Excel

Le contenu de la base tient compte des deux dimensions à appréhender plus particulièrement :

- niveau spatial : étude à l'échelle des départements du territoire français
- niveau temporel : étude sur deux périodes – avant et après 1970.

Le fichier Excel est constitué de **5 feuilles nécessaires** pour notre étude :

➤ Genres

La feuille « Genres » liste les **81 genres** de syrphidés.

NUM	CODE	GENRE	NbEspeces	DepTOTAL	DepAv70	DepAv70Ap	Dep70Ap
1	Anas	Anasimyia	5	19	6	10	3
2	Arct	Arctophila	2	15	4	8	3
3	Bacc	Baccha	1	20	3	17	0
4	Bler	Blera	1	10	3	6	1
5	Bra1	Brachyopa	8	17	6	9	2
6	Bra2	Brachypalpoïdes	1	18	3	14	1
7	Bra3	Brachypalpus	3	14	7	5	2
8	Cali	Caliprobola	1	16	2	12	2
9	Call	Callicera	6	23	13	7	3
10	Ceri	Ceriana	2	21	17	3	1
11	Chal	Chalcosyrphus	5	22	4	13	5

Numéro d'ordre → NUM
 Nom → GENRE
 Nombre de total de département où au moins 1 espèce a été vue → DepTOTAL
 Nombre de département où au moins 1 espèce a été vue avant et après 1970 → DepAv70, DepAv70Ap, Dep70Ap
 4 premiers caractères du nom du genre. En cas d'homonymie, le 4^{ème} caractère est remplacé par un chiffre → CODE
 Nombre d'espèces que comporte le genre → NbEspeces
 Nombre de département où au moins 1 espèce a été vue qu'avant 1970 → DepAv70
 Nombre de département où au moins 1 espèce a été vue qu'après 1970 → Dep70Ap

➤ Espèces

La feuille « Espèces » liste les **489 espèces** de syrphidés.

NUM	CODE	GENRE	ESPECE	Dept Tota	Dept Av70	Dept 70Ap	Dept Av70 Ap	REFERENCE	SYNO1	REF1	SYNO2	REF2
1	AnasCont	Anasimyia	contracta	7	4	3	0	Claussen & Torp, 1980				
2	AnasInte	Anasimyia	interpuncta	3	2	1	0	(Harris), 1776				
3	AnasLine	Anasimyia	lineata	7	1	5	1	(Fabricius), 1787				
4	AnasLunu	Anasimyia	lunulata	1	1	0	0	(Meigen), 1822				
5	AnasTran	Anasimyia	transfuga	8	2	5	1	(L.), 1758				
6	ArctBomb	Arctophila	bombiforme	8	3	4	1	(Fallen), 1810				
7	ArctSupe	Arctophila	superbiens	11	4	6	1	(Muller), 1776	A. mussitans	Fabricius, 1776	A. fulva	Harris, 1780
8	BaccElon	Baccha	elongata	20	3	17	0	(Fabricius), 1775	B. obscuripennis	Meigen, 1822	B. karptica	Violovitsh, 1976

Numéro d'ordre → NUM
 Nom du genre → GENRE
 Nombre de département où au moins 1 espèce a été vue : Total, avant, après, avant et après 1970 → Dept Tota, Dept Av70, Dept 70Ap, Dept Av70 Ap
 Nom du 1^{er}, 2^{ème} synonyme → SYNO1, SYNO2
 Code du genre (4 car.) + 4 premiers caractères du nom de l'espèce. En cas d'homonymie, le 8^{ème} caractère est remplacé par un chiffre → CODE
 Nom → ESPECE
 Auteur ayant nommé l'espèce, année de publication → REFERENCE
 Référence du 1^{er}, 2^{ème} synonyme → REF1, REF2

Espèces indicatrices de forêts d'importance internationale pour la conservation de la nature à l'échelle européenne (d'après Good & Speight, 1996)

Nom du 3^{ème}, 4^{ème} synonyme

Indicateur d'impression de l'espèce (si 1) dans le fichier pour l'UEF

SYNO3	REF3	SYNO4	REF4	Fich UEF	Statut Europ	statut France	Import. internat	Photo
				1		S		0
				1	S	D		0
				1		S		0
				1	D	M		0
				1		S		0
				1	S	S		0
				1				0
B. tuvinica	Violovitsh, 1976			1				0

Référence du 3^{ème}, 4^{ème} synonyme

Statut de l'espèce au niveau français, européen :
 S : à surveiller, probablement en diminution
 D : en nette diminution
 M : menacée

Disponibilité d'une photo numérique de l'espèce

➤ **Départements**

La feuille « Départements » liste les **489 espèces** de syrphidés.

Nombre de genre vus dans le département : Total, avant, après, avant et après 1970

Code

Nom

Nombre d'espèces vues dans le département : Total, avant, après, avant et après 1970

CODE	NOM	EspTOTAL	EspAv70	Esp70Ap	EspAv70Ap	GenTOTAL	GenAv70	Gen70Ap	GenAv70Ap
01	Ain	85	0	85	0	38	0	38	0
02	Aisne	7	7	0	0	6	6	0	0
03	Allier	74	30	38	6	38	13	17	8
04	Alpes-de-Haute-Provenc	36	15	18	3	10	3	2	5
05	Hautes-Alpes	58	20	30	8	20	9	5	6
06	Alpes-Maritimes	49	6	43	0	21	1	17	3
07	Ardèche	4	2	1	1	4	2	1	1
08	Ardennes	18	18	0	0	14	14	0	0
09	Ariège	107	4	100	3	42	1	38	3
10	Aube	2	2	0	0	2	2	0	0
11	Aude	76	11	64	1	37	7	25	5

➤ **Observations**

La feuille « Observations » recense les **4593 observations** syrphidés, référencées ou non (données supplémentaires)

En ce qui concerne les observations, quatre classes nous intéressent pour la représentation : avant 1970, après 1970, avant et après 1970, pas d'observation.

Concernant l'information « année d'observation », celle-ci peut être sous la forme d'une année exclusive, d'une période de temps.

Diagramme illustrant la structure d'un tableau de données :

- Numéro d'ordre
- Département dans lequel l'espèce a été observée
- Année(s) de l'observation
- Date mise à jour

NUM	ESPECE	DEPT	TYPE	ANNEE	REFERENCE	DATE_MAJ
1	AnasCont	02	1	1920	Speight, 1994	30/07/1999
2	AnasCont	18	2	1992	Speight, 1996	30/07/1999
3	AnasCont	21	1	1925	Speight, 1994; Speight, 1996	30/07/1999
4	AnasCont	31	2	1994	Sarthou & Speight, 1997	30/07/1999
5	AnasCont	35	2	1991	Speight, 1994	30/07/1999
6	AnasCont	78	1	1926	Speight, 1994	30/07/1999
7	AnasCont	80	1	1910, 1927	Speight, 1994	30/07/1999
8	AnasInte	37	2	1981	Speight, 1994; Speight, 1996	30/07/1999
9	AnasInte	55	1	1943, 1945	Speight, 1994	30/07/1999
10	AnasInte	80	1	1949, 1950	Speight, 1994	30/07/1999

Code de l'espèce sur 8 lettres

1 si observée avant 1970
2 si observée en 1970 ou après

Référence bibliographique : auteurs et année

➤ Refs

La feuille « Refs » liste les **93 références bibliographiques** citées dans les observations.

Diagramme illustrant la structure d'un tableau de données :

- Nom
- Année
- Auteur et année
- Référence complète

CodeREF	AUTEUR	ANNEE	Ordre	CITATION	REF COMPLETE
Mou1980	Mouillé	1980		Mouillé, 1980	MOUILLE P., 1980 - Liste des captures de Diptères en Normandie. Muséum d'Histoire Naturelle. Elbeuf. 1973-1980 : 37-44.
Pec1988	Peck	1988		Peck, 1988	PECK L.V., 1988 - Syrphidae. In: Soos, A. & Papp, L. (eds.) Catalogue of Palaearctic Diptera, 8 : 11-230. Akad.Kiado, Budapest.
Ree2000	Reemer	2000		Reemer, 2000	REEMER M., 2000 - A new species of Parhelophilus Girschner, 1897 (Diptera, Syrphidae) from south-western Europe. Dipteron, 3 : 1-6.

3 premières lettres du 1er nom + année + éventuellement a ou b

Cette première étape : consultation et étude de l'existant, nécessaire pour s'appropriier le sujet, a permis de choisir la configuration adaptée pour répondre aux objectifs du cahier des charges.

III. CHOIX DE LA CONFIGURATION

3.1 Proposition de configuration

Lors de la proposition de projet, les responsables nous ont proposé 3 configurations de base de données possibles:

Solution tout Access

Définition :

La partie Web accède aux Tables d'une **base de données Access via PHP** et le protocole de communication ODBC (pour accéder aux tables Access).

La partie locale ENSAT est gérée par cette même **application Access** (Tables, Requêtes, Formulaires, Etats, Macros).

Notre point de vue :

- Les avantages
 - Cette solution permet de mettre en œuvre le développement d'une application unique.
 - Il n'y aura pas de problème de mise à jour puisqu'il s'agira d'une seule application.
- Les inconvénients
 - Access dispose d'une connexion multi-utilisateurs restreinte, ainsi dans le cas de site Web, si la consultation est importante, le serveur risque de ne pas répondre correctement.
 - Il paraît difficile de lier une image de la France avec Access sans passer par un SIG et il semble donc difficile de pouvoir générer des analyses thématique à la demande.

Solution tout MySql

Définition :

La partie Web accède aux Tables d'une **base de données MySql via PHP** sans passerelle ODBC.

La partie locale ENSAT est gérée par une **application Web sécurisée** (login-mot de passe) accédant à la même base MySql que la partie publique.

Point de vue :

- Les avantages
 - Cette solution permet de mettre en œuvre le développement d'une application unique.
 - Il n'y aura pas de problème de mise à jour puisqu'il s'agira d'une seule application.
 - Il est possible de coupler MySQL à MapServer en utilisant une carte de la France sous forme de fichier shape et de réaliser des analyses thématiques à la demande par ce lien.
- Les inconvénients
 - Il n'est pas possible de gérer l'intégrité des clés étrangères.

Solution Mixte MySQL – Access

Définition :

La partie Web accède aux Tables d'une **base de données MySql via PHP** sans passerelle ODBC.

La partie locale ENSAT est gérée par une **application Access** (Tables, Requêtes, Formulaires, Etats, Macros). Il faut prévoir le transfert périodique (semestriel) des données de la base locale Access vers la base web MySql.

Point de vue :

- Les avantages
 - Il est possible de coupler MySQL à MapServer en utilisant une carte de la France sous forme de fichier shape et de réaliser des analyses thématiques à la demande par ce lien.
 - Il n'y aura pas de problème de mise à jour car il est possible de faire communiquer Access à MySQL par lien ODBC.
- Les inconvénients
 - Il n'est pas possible de gérer l'intégrité des clés étrangères.
 - Cela nécessite le développement de deux applications distinctes.

3.2 Autre configuration : notre choix

Compte tenu des spécifications du projet et de nos connaissances, il nous est apparu intéressant d'utiliser une quatrième solution :

Utiliser comme base de données : **PostgreSQL**, qui nous permettait de stocker la géométrie de la carte de France directement dans une base de données en utilisant la cartouche spatiale **PostGIS**. Ainsi, nous évitons de passer par un fichier shape comme spécifié dans la solution MySQL.

Coupler cette base de données à **MapServer** pour nous permettre de générer des cartes à la demande en interrogeant la base de données. En théorie, il est également possible d'utiliser MapServer en mode Query afin de pouvoir interroger la base de données à partir d'une sélection graphique (cette dernière fonctionnalité n'a pu être mise en œuvre du fait d'un manque de temps pour la compréhension de ce système).

Gérer la partie administrative directement à partir du site Web, grâce à un accès sécurisé. Compte tenu du délai dont nous disposions et de la découverte de la possibilité de faire communiquer **Access** et PostgreSQL par **lien ODBC** et cela même en connexion distante, nous avons choisi de développer une application administrative sous Access.

Par conséquent la configuration utilisée pour répondre aux objectifs du projet est la suivante :

- Le système d'exploitation : Windows
- Le serveur Web : Apache
- Le langage de script pour le Web dynamique : PHP
- Le serveur cartographique : Mapserver
- La base de données : PostgreSQL
- La base de données spatiales : Postgis
- La gestion administrative de la base de données : Access
- Des fichiers issus d'un SIG contenant les objets géographiques au format .shp (ArcView)

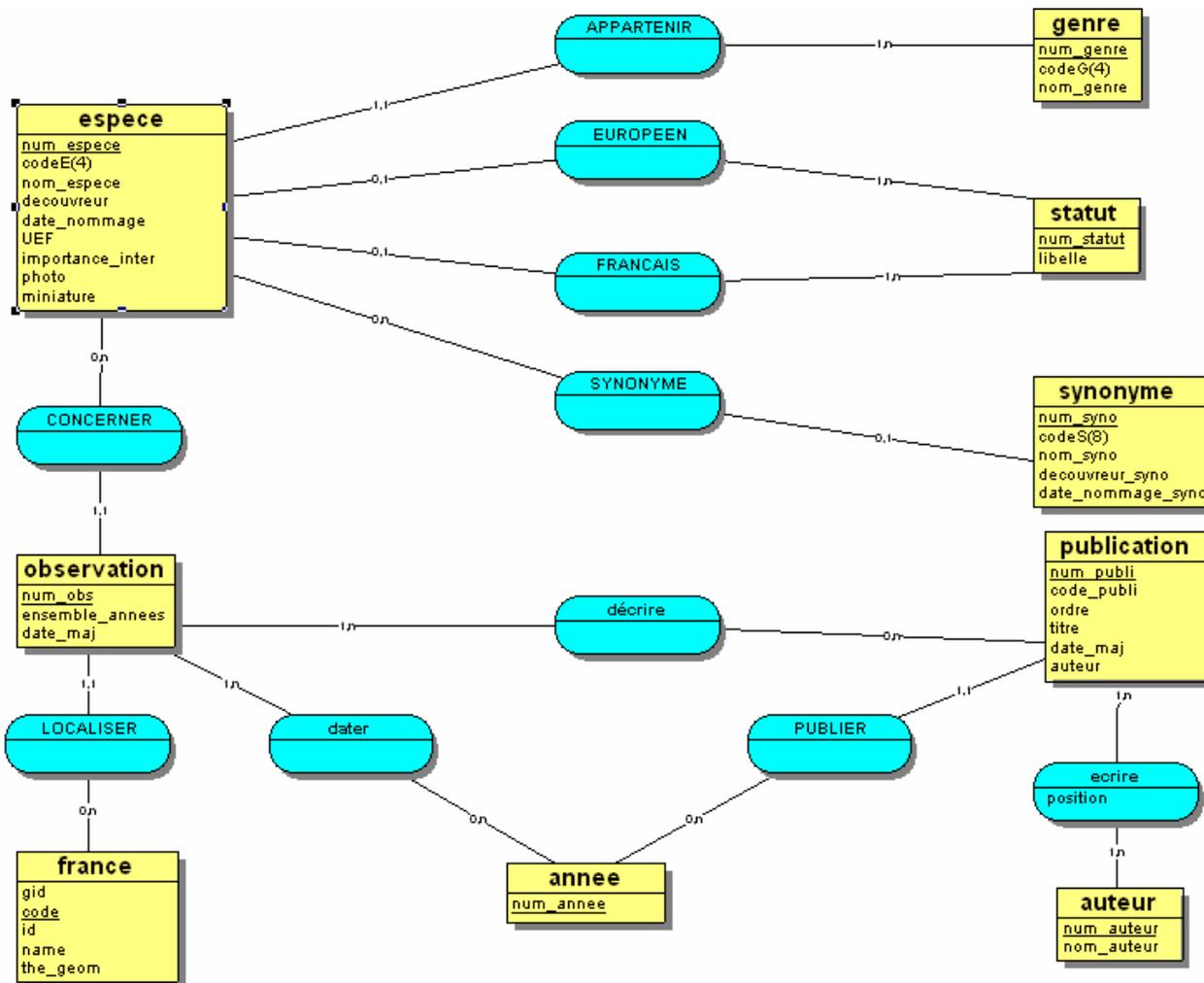
IV. LA NOUVELLE VERSION DU SITE INTERNET

Avant toute création d'interface cartographique, nous avons dû restructurer la base de données. Nous vous présentons, dans cette partie, les points essentiels du projet : la base de données, les deux interfaces cartographiques et la gestion administrative. L'ensemble du site est consultable dans le guide de l'utilisateur.

4.1 Restructurer la base de données

4.1.1 Nouveau Modèle Conceptuel de Données

Tenant compte du premier MCD et du contenu de la base (Excel), nous avons conçu le MCD suivant :



Voici le LMD :

espece	=	{ num espece ; codeE(4) ; nom_espece ; decouvreur ; date_nommage ; UEF ; importance_inter ; photo ; ref_statut_fr ; ref_statut_eu ; ref_genre }
genre	=	{ num genre ; codeG(4) ; nom_genre }
statut	=	{ num statut ; libelle }
synonyme	=	{ num syno ; codeS(8) ; nom_syno ; decouvreur_syno ; date_nommage_syno ; ref_espece }
observation	=	{ num obs ; ensemble_annees ; date_maj ; ref_dept ; ref_espece }
france	=	{ gid ; code ; id ; name ; the_geom }
publication	=	{ num publi ; code_publi ; ordre ; titre ; date_maj ; ref_annee_publi }
auteur	=	{ num auteur ; nom_auteur }
annee	=	{ num annee }
dater	=	{ ref_observ ; ref_annee_obs }
ecrire	=	{ ref_publi ; ref_auteur , position }

décrire	=	{ ref_obs ; ref_publi }
---------	---	-------------------------

Le contenu de ces tables et vues sont disponibles dans les annexes de ce rapport.

4.1.2 Nouvelles tables et vues

Table	Propriétaire	Tablespace	Nombre d'enregistrements estimés
annee	syrfid		2011
auteur	syrfid		76
dater	syrfid		6293
décrire	syrfid		4742
ecrire	syrfid		144
espece	syrfid		511
france	syrfid		95
genre	syrfid		81
geometry_columns	syrfid		16
observation	syrfid		4593
publication	syrfid		93
spatial_ref_sys	syrfid		1785
statut	syrfid		3
synonyme	syrfid		163

Voici les 14 tables nécessaires et le nombre d'enregistrements contenus dans chaque table.

Vue	Propriétaire
effectifs_espece	syrfid
effectifs_espece_ap70	syrfid
effectifs_espece_av70	syrfid
effectifs_espece_avap70	syrfid
effectifs_genre	syrfid
effectifs_genre_ap70	syrfid
effectifs_genre_av70	syrfid
effectifs_genre_avap70	syrfid
espece_dept_annee	syrfid
france_annee_geom	syrfid
france_esp_genre	syrfid
genre_dept_annee	syrfid
liste_multiple	syrfid
observation_annee_minmax	syrfid

Les 14 vues suivantes ont été créées pour faciliter l'accès des données contenus dans des tables différentes.

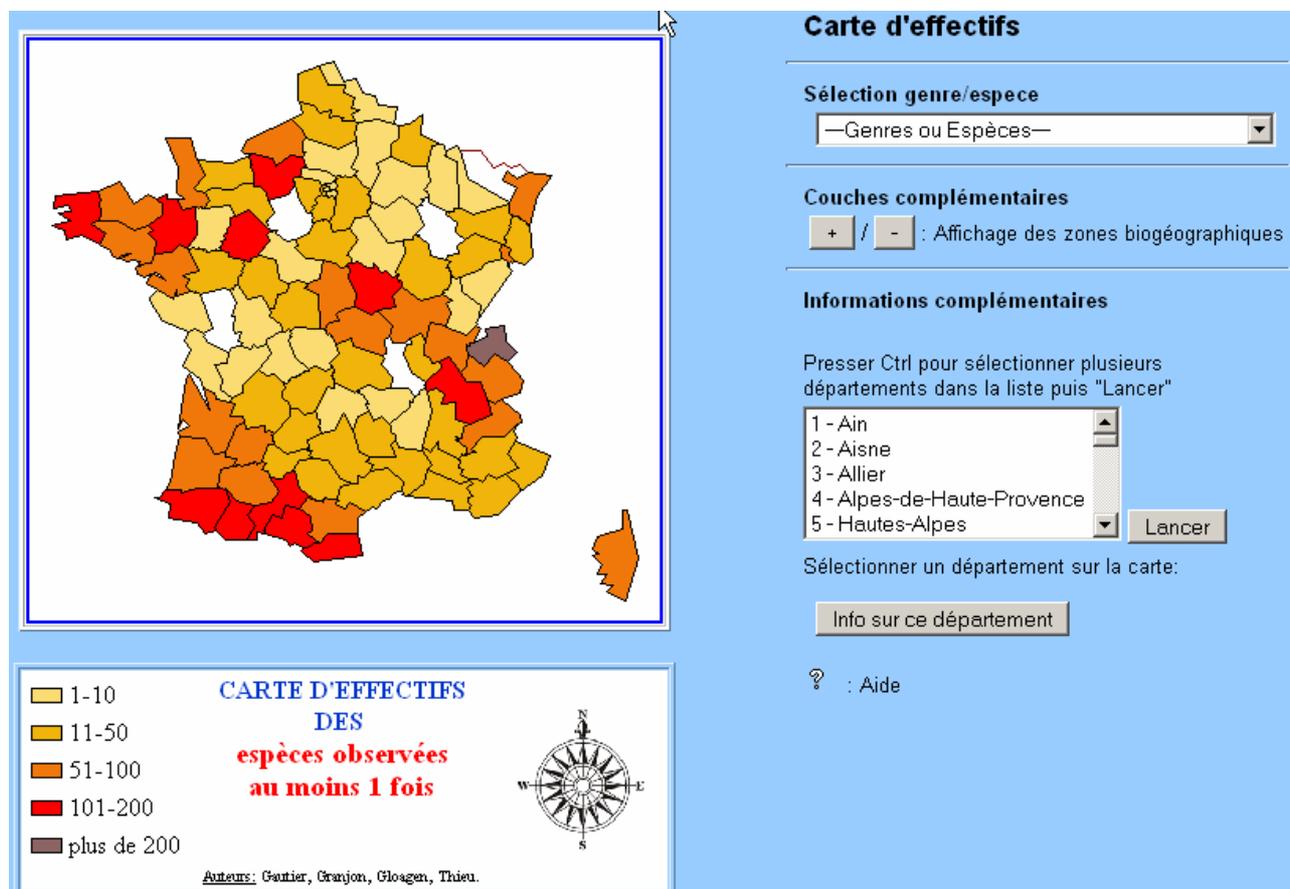
Ces requêtes sont interrogeables en table depuis php.

4.2 Interrogation de la base de données, les interfaces

4.2.1 Interface cartographique

Interface : Carte des effectifs

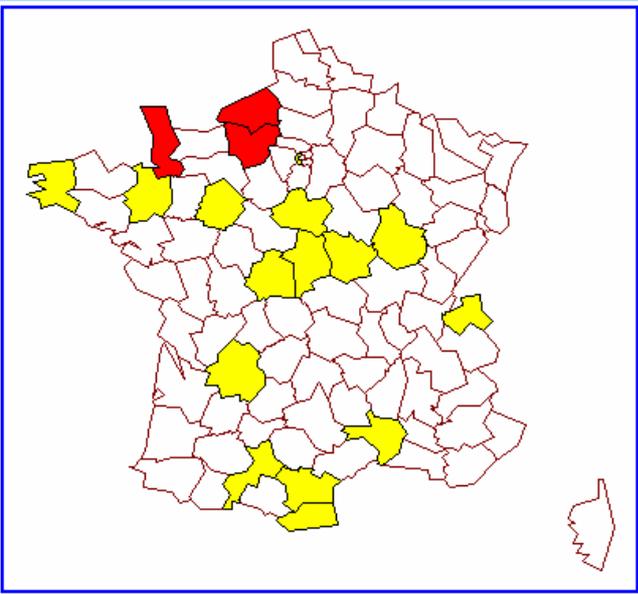
L'objectif est de représenter des cartes d'effectifs basées sur la fréquence d'observation de l'ensemble des genres ou de l'ensemble des espèces de syrphes actuellement recensées.



Pour plus d'informations, consulter le guide utilisateur.

Interface : Carte par répartition

L'objectif est de représenter la répartition sur l'ensemble du territoire français d'une espèce de syrphe selon qu'elle est été observée uniquement avant 1970, avant et après 1970 ou uniquement après. D'autre part cette interface permet d'obtenir pour l'espèce sélectionnée, des informations synthétiques, devant permettre de faciliter l'identification pour l'utilisateur.



Carte de répartition

Sélection genre/espece
 Volucella [v] zonaria [v]

Sélection statut/espece
 en France [Statut [v]]
 en Europe [Espèce [v]]

Couches complémentaires
 + / - : Affichage des zones biogéographiques

Informations complémentaires
 AA : Info Espece ? : Aide
 Sélectionner un département sur la carte:
 Info sur ce département [b]

/ département
■ avant 1970 seulement
■ avant et apres 1970
■ apres 1970

REPRESENTATION DE LA REPARTITION DE : *Volucella zonaria*

Auteurs: Gautier, Granjon, Gloaguen, Thieu.





Synonyme de l'espece:

Pour plus d'informations, consulter le guide utilisateur.

4.2.2 Autres fonctionnalités

Bibliothèque dynamique

La bibliographie est totalement dynamique. En effet, les publications référencées sont directement liées à la base de données.

Par conséquent, lors de la mise à jour des données et donc peut-être de l'ajout d'une publication, celle-ci viendra directement s'afficher dans cette page.

Références bibliographiques

A - B - C - D - E - F - G - H - I - J - K - L - M - N - O - P - Q - R - S - T - U - V - W - X - Y - Z

A ↑

Aguilar & Coutin, 1988
 AGUILAR J., COUTIN R., 1988 - Cheilosia caerulescens Meigen, ravageur des Joubarbes (Dipt., Syrph.). Bull. Soc. ent. Fr. 92 (9-10) : 307-309.

Andréi-Ruiz, 1998
 ANDREI-RUIZ M. C., 1998 - Stratification verticale et diversité des peuplements d'insectes des végétations sclérophylles méditerranéennes. Réserve de la biosphère du Fangou. Thèse de Doctorat, Université de Corte. 158 pp.

Aubert et al., 1976
 AUBERT J., AUBERT J.-J., GOELDLIN DE TIEFENAU P., 1976 - Douze ans de captures systématiques de Syrphides (Dipt.) au col de Bretolet (Alpes valaisiennes). Mitt. schweiz. ent. Ges. 49 : 115-142.

B ↑

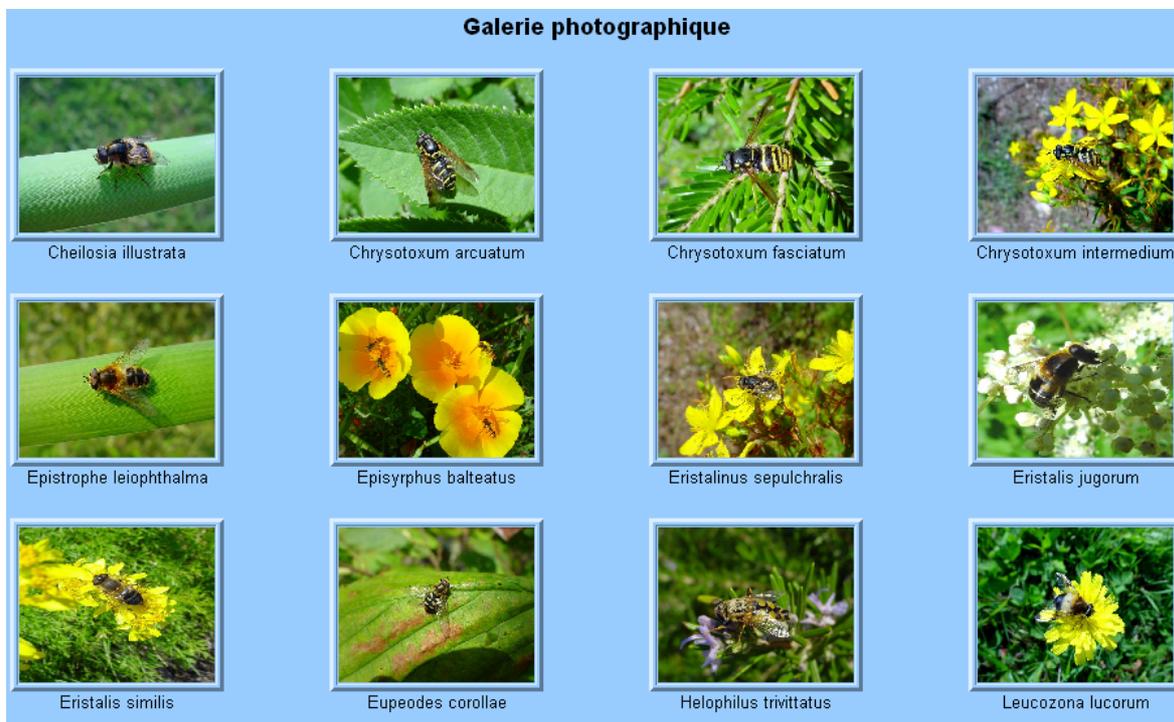
Bausenwein, 1993
 BAUSENWEIN D., 1993 - Schwebfliegen-Nachweise aus Südfrankreich (Dipt.: Syrph.). Entomologische Zeitschrift, 103 (5) : 80-87.

Beaucournu & Matile, 1963
 BEAUCOURNU J.C., MATILE L., 1963 - Contribution à l'inventaire faunistique des cavités souterraines de l'ouest de la France. Annales de spéléologie XVIII (4) : 519-531.

Galerie photo

La galerie photo est également dynamique. Les photos ne se trouvent pas dans la base de données mais sont dans un dossier « Photo » à la racine du site.

Par conséquent, lors de l'ajout d'une photo dans ce dossier, celle-ci viendra directement s'afficher dans cette page.

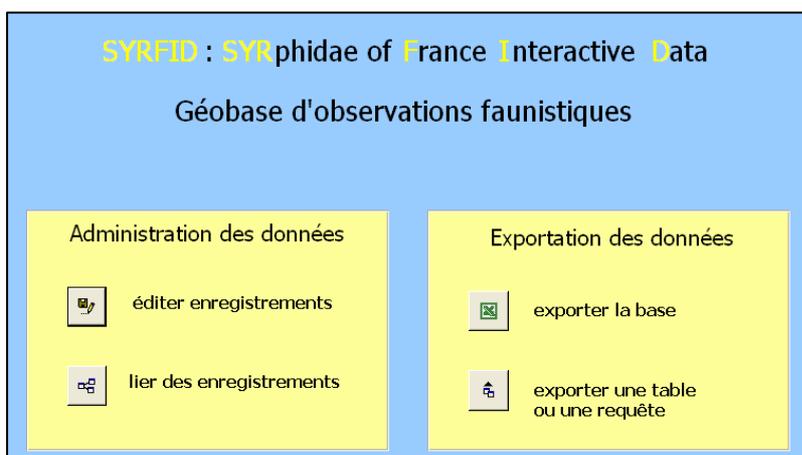


En cliquant sur une photo, celle-ci apparaît dans une nouvelle fenêtre à sa taille originale.

4.3 Administration de la base

Compte tenu du délai dont nous disposons et de la découverte de la possibilité de faire communiquer **Access** et PostgreSQL par **lien ODBC** et cela même en connexion distante, nous avons choisi de développer une application administrative sous Access.

En voici l'interface :



Pour plus d'informations, consulter le guide d'utilisation

V. BILAN ET PERSPECTIVE

Rappelons l'objectif de ce projet :

Développer une nouvelle version Internet de SYRPHID permettant

- D'effectuer une génération dynamique des cartes; ceci supposera de reprendre le MCD et revoir la structure de la base de données
- D'assurer si possible toutes les fonctions de la version 2 avec une ergonomie analogue
- D'ajouter de nouvelles fonctions de sélection cartographiques (sélection de tout un ensemble de département par dessin d'un rectangle à la souris et/ou par clics successifs sur certains départements)
- De permettre une mise à jour sécurisée des données (ce qui est une limite de la version Excel actuelle)

Au final, nous pouvons dire que la nouvelle version du site SYRFID répond en grande partie à la demande émanant du laboratoire Biodiversité dans les agroécosystèmes.

Sa mise en ligne, donc son ouverture au grand public, profite à l'ensemble des professionnels et autres utilisateurs s'intéressant de loin ou de près aux Syrphidés. Cette diffusion sur Internet donne encore plus de valeur à ce travail.

Le cahier des charges a précisé l'importance de pouvoir mettre à jour des données. Cette gestion administrative est possible dès aujourd'hui dans Access.

Pour permettre la mise à jour des données mais également l'utilisation du site, les rédactions du manuel d'utilisation et du guide de maintenance ont été nécessaires. L'enjeu est de garantir la possibilité de modification future de l'interface (ajout de données, de fonctions...).

L'outil de cartographie dynamique SYRFID répond aux volontés des instigateurs du projet définies dans le cahier des charges. La mission est remplie dans la mesure où le produit fini est opérationnel. Cet outil permet une visualisation optimale des données des espèces des Syrphidés et permet également le traitement des données grâce à l'export dans Excel.

Pour conclure, la création d'un site dynamique, cartographique liée à une base de données riches en informations, nous a permis d'enrichir et de consolider nos compétences informatiques.

En effet, ce projet de cinq semaines a été propice à l'apprentissage et à l'expérimentation de nouvelles applications informatiques. Il nous a permis également de mettre en œuvre des langages complexes acquis lors des enseignements du DESS SIGMA.

Manuel de l'utilisateur

SYRFID

Presentation Documentation Serveur cartographique Lien

SYRFID v3.0

SYRphidae of France Interactive Data

Données interactives sur les observations des espèces
de DIPTERA SYRPHIDAE et MICRODONTIDAE en France

INP ENSAT

Site réalisé par Mlle Gautier, Mr Gloaguen, Mr Granjon et Mr Thieu dans le cadre du projet de fin d'étude du DESS SIGMA, en collaboration avec Mr Monteil, Mr Sarthou et Mr Souques.

Map Server
PostgreSQL

Référence à utiliser pour citer SYRFID
SARTHOU J.P., MONTEIL C. et DUSSAIX C., 2005
SYRFID v3.0 : Syrphidae of France Interactive Data [On-Line URL : <http://syrfid.ensat.fr/>].
Dernière mise à jour : 20 Février 2005
© SYRFID 2005. INP/ENSAT

PostGIS

GAUTIER Bénédicte
GLOAGUEN Jérémy
GRANJON Ludovic
THIEU Vincent



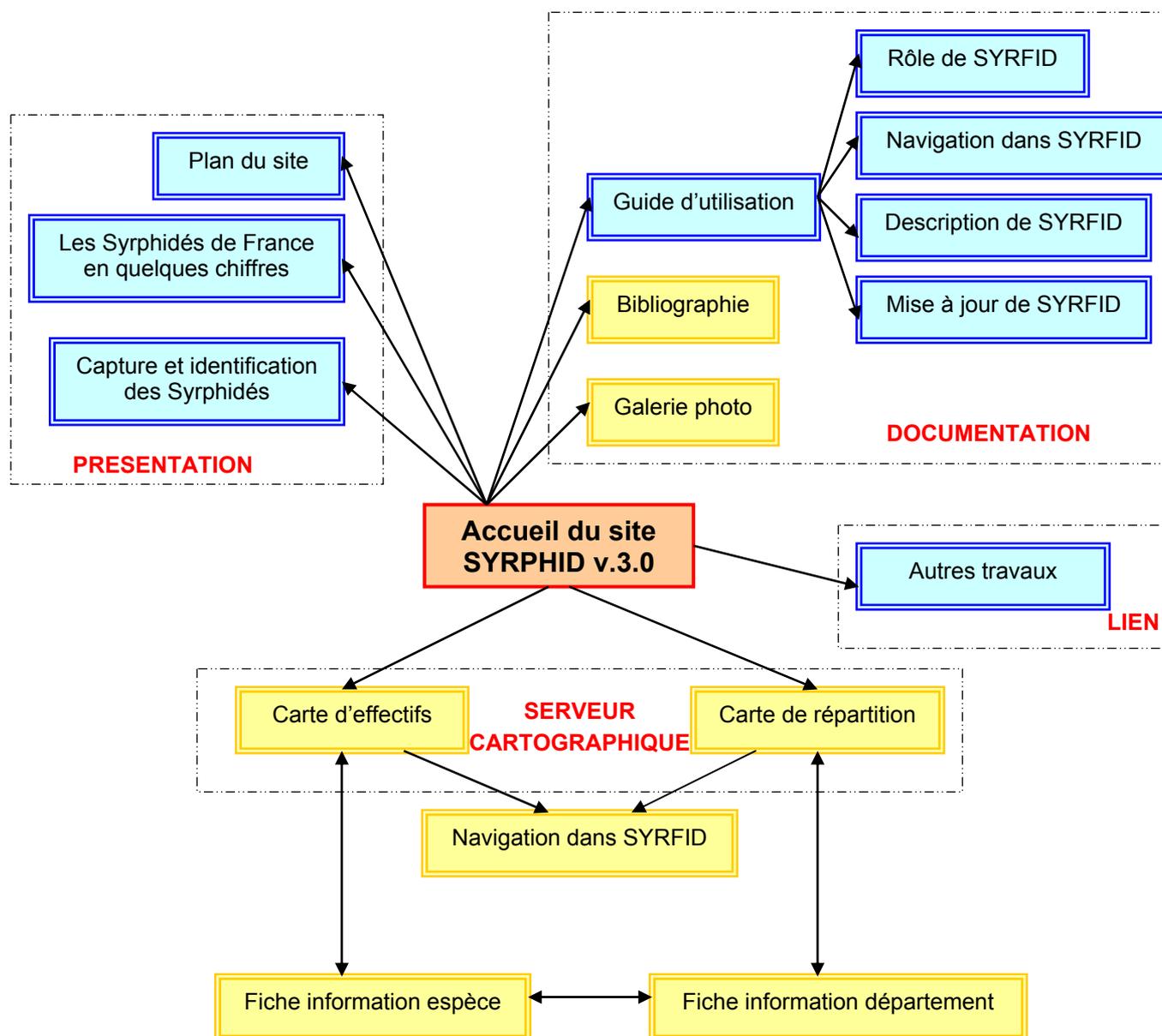
Responsables : Claude MONTEIL et Marc SOUQUES

Fevrier 2005

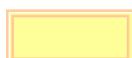
SOMMAIRE

I. DESCRIPTION GENERALE DU SITE	3
II. LE PORTAIL DU SITE	4
2.1 Présentation	4
2.2 Les menus	4
2.2.1 Présentation	4
2.2.2 Documentation	5
2.2.3 Serveur cartographique	6
2.2.4 Lien	7
III. L'INTERFACE : CARTE DE REPARTITION	8
3.1 Présentation	8
3.2 Recherche thématique	8
3.2.1 Fonctionnement et sélection	8
3.2.2 Autres informations disponibles	11
IV. L'INTERFACE : CARTE D'EFFECTIFS	13
4.1 Présentation	13
4.2 Recherche thématique	13
4.2.1 Fonctionnement	13
4.2.2 Informations complémentaires	14
V. L'INTERFACE : ADMINISTRATION ET EXPORTATION DE LA BASE	16
5.1 Présentation	16
5.2 Fonctionnalités	17
5.2.1 Ajout d'un enregistrement	17
5.2.2 Modification d'un enregistrement	18
5.2.3 Liaison des enregistrements	18
5.3 Export des données vers Excel	19

I. DESCRIPTION GENERALE DU SITE



 : ces pages n'interrogent pas la base de données. Elles sont statiques

 : ces pages interrogent la base de données. Elles sont dynamiques

TITRE : correspond aux menus se trouvant sur la barre de menu horizontale sur la page d'accueil

II. LE PORTAIL DU SITE

2.1 Présentation

L'accueil se présente sous cette forme :



L'affichage est optimisé pour une résolution écran de 1024x768 plein écran.

La compatibilité est établie avec Internet Explorer et Mozilla.

2.2 Les menus

Pour avoir accès aux différentes parties du site, nous avons choisi de créer le menu dynamique horizontal suivant :



[2.2.1 Présentation](#)

Le menu **Presentation** vous permet d'avoir accès à plusieurs sous-menus :

➤ Plan du site



Cette page vous permet de consulter tout ce qu'il est possible de faire dans ce site.

➤ Les Syrphidés en France

Ce lien vous permet d'avoir un état des lieux des Syrphidés sur le territoire national. Nous apprenons que la France possède la syrphidofaune la plus riche d'Europe avec 511 espèces de Syrphidés.

SYRFID compte à ce jour 4593 observations correspondant à un triplet espèce / département / période temporelle (avant ou depuis 1970).

➤ Capture et identification

Dans cette page, nous vous donnons quelques conseils pour la capture et l'identification des Syrphidés.

[2.2.2 Documentation](#)

Le menu **Documentation** vous permet d'avoir accès à plusieurs sous-menus :

➤ Guide d'utilisation

M Dessaix, M Monteil et M Sarthou ont souhaité expliquer à tous les utilisateurs du site SYRFID:

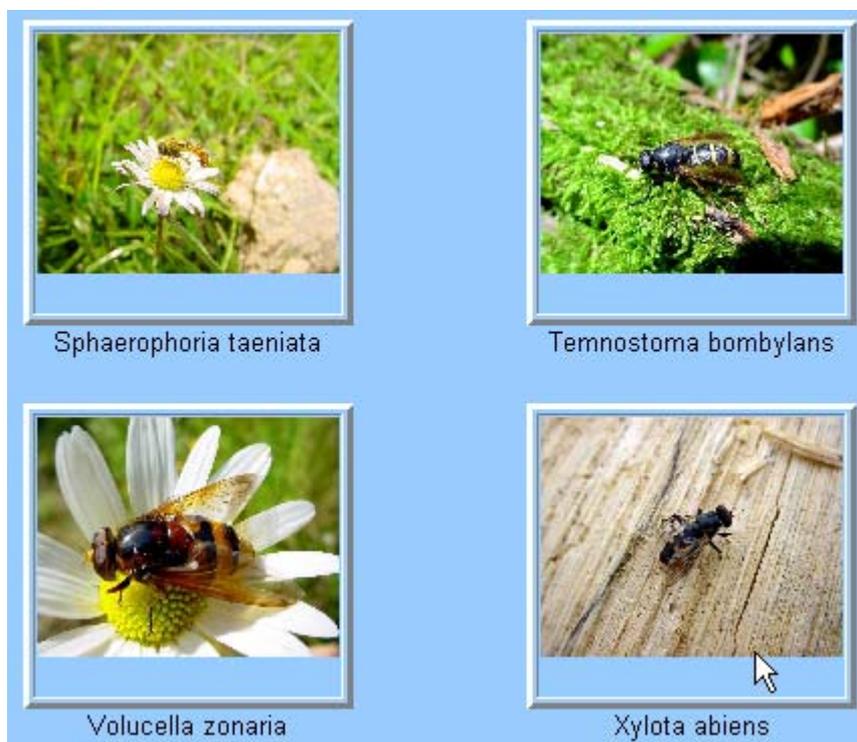
1. Le rôle du site SYRFID
2. Le contenu
3. La navigation
4. La mise à jour

➤ Bibliographie

Ce lien affiche l'ensemble des auteurs des publications où au moins une observation de Syrphidé a été citée.

Cette liste est mise à jour automatiquement lors d'un ajout d'une nouvelle publication dans la base de données.

➤ Galerie photo



En cliquant sur une photo, vous aurez l'agrandissement de la photo dans une nouvelle fenêtre. Cette page est mise à jour automatiquement lors d'un ajout d'une photo.

[2.2.3 Serveur cartographique](#)

Le menu **Serveur cartographique** vous permet d'avoir accès à plusieurs sous-menus :

➤ Carte d'effectifs

Cette interface est de type quantitatif. Elle nous permet de visualiser les répartitions globales de genre et d'espèce par département. L'accès se fait par une liste de choix à huit entrées. Pour plus d'informations consultez la suite du manuel.

➤ Carte de répartition

Cette interface est qualitative. Elle offre la possibilité de visualiser des cartes de répartition d'observations par genres, espèces ou espèces protégées. La demande s'effectue par listes de choix. Dès le choix effectué, la carte est mise à jour automatiquement.

Plus finement, ces observations sont classées selon une dimension temporelle. Trois périodes sont proposées : Pas d'observation, observations uniquement avant 1970, observations avant ET depuis 1970, observations depuis 1970 uniquement.

Pour plus d'informations consultez la suite du manuel.

2.2.4 Lien

Le menu **Lien** vous permet d'avoir accès au sous menu : Autres travaux.

Comme l'indique le nom, cette page offre un récapitulatif des sites dont le thème est : Les Syrphidés.

III. L'INTERFACE : CARTE DE REPARTITION

3.1 Présentation

L'objectif est de représenter la répartition sur l'ensemble du territoire français d'une espèce de syrphes selon qu'elle est été observée uniquement avant 1970, avant et après 1970 ou uniquement après. D'autre part cette interface permet d'obtenir pour l'espèce sélectionnée, des informations synthétiques, devant permettre de faciliter l'identification pour l'utilisateur.

Carte de répartition

Sélection genre/espece
 Epistrophe | leiophthalma

Sélection statut/espece
 en France | Statut
 en Europe | Espèce

Couches complémentaires
 + / - : Affichage des zones biogéographiques

Informations complémentaires
 🗺️ : Info Espece | ? : Aide
 Sélectionner un département sur la carte:
 Info sur ce département | 66

REPRESENTATION DE LA REPARTITION DE : Epistrophe leiophthalma

département
 avant 1970 seulement
 avant et apres 1970
 apres 1970

Auteurs: Gautier, Granjon, Gloaguen, Thieu.

Synonyme de l'espece:
 ♦ E. liophthalma

1 : à surveiller
 2 : en diminution
 3 : menacée

3.2 Recherche thématique

3.2.1 Fonctionnement et sélection

Cette interface propose deux modes de sélections d'espèce :

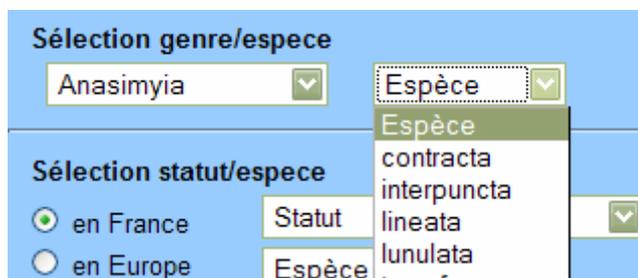
- Soit par la présélection d'un nom de genre permettant le choix d'une espèce parmi toutes celles associées à ce genre (relation 1-N entre l'espèce et le genre).
- Soit par la présélection d'un statut appliqué en France ou en Europe.

Le premier type de sélection, permet après le choix du genre, d'obtenir tous noms d'espèce appartenant à ce genre. Cette solution permet d'identifier rapidement toutes les espèces associées à un nom de genre et d'effectuer une sélection rapide de l'espèce recherchée.

1^{ère} étape : sélection du genre



2^{ème} étape : sélection de l'espèce

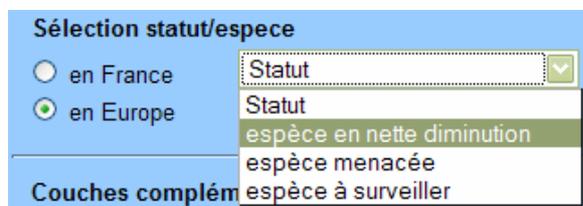


Ce premier mode de sélection est qualifié de complet, puisque toutes les espèces recensées sont présentes et disponibles à la sélection.

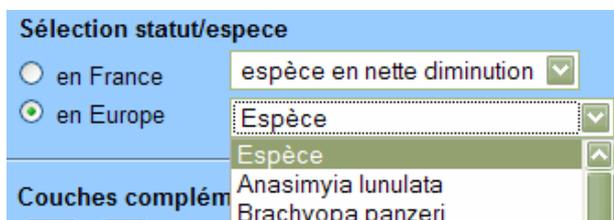
L'interface propose un deuxième type de sélection d'espèce, par la présélection d'un statut (surveillé, en diminution, menacée) appliqué à l'échelle de la France ou de l'Europe.

Si l'utilisateur s'oriente vers ce mode de sélection, il devra tout d'abord identifier le statut et sa portée française ou européenne, sachant que le bouton « France » est présélectionné par défaut. Une fois ce premier choix réalisé, l'ensemble des espèces relatives à cette présélection sont présentées. Il reste à l'utilisateur le choix de l'espèce recherchée.

1^{ère} étape : sélection du statut



2^{ème} étape : sélection du (genre espece)

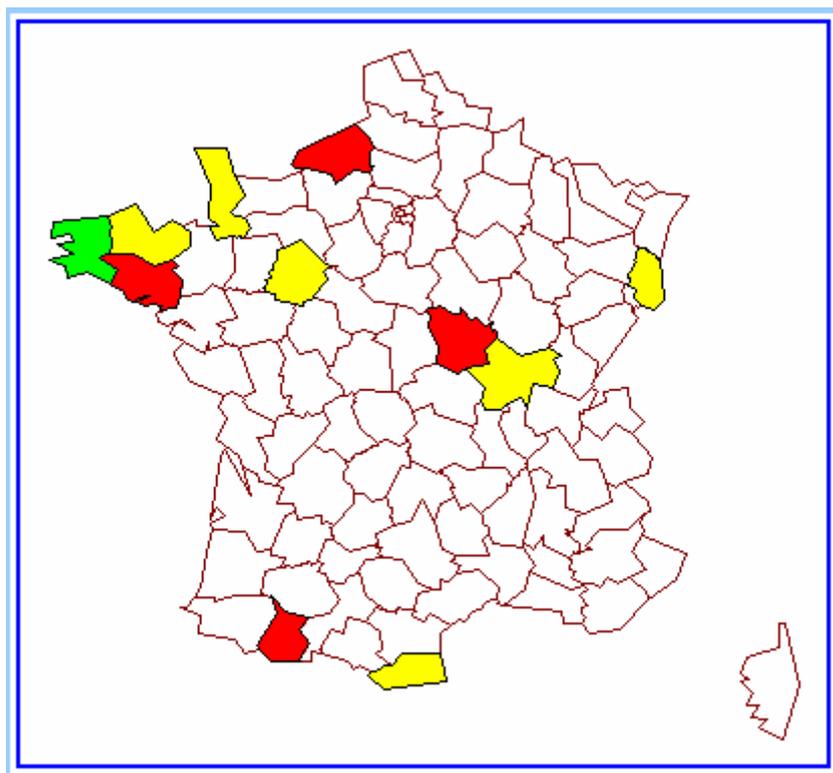


Il est évident que ce deuxième type de recherche restreint le nombre d'espèce disponible à la sélection, puisque seules les espèces qualifié par un statut sont présentées.

Quelque soit le mode de sélection utilisé, la finalité reste la sélection d'une espèce. C'est sur la base de cette sélection que l'interface affiche différentes informations :

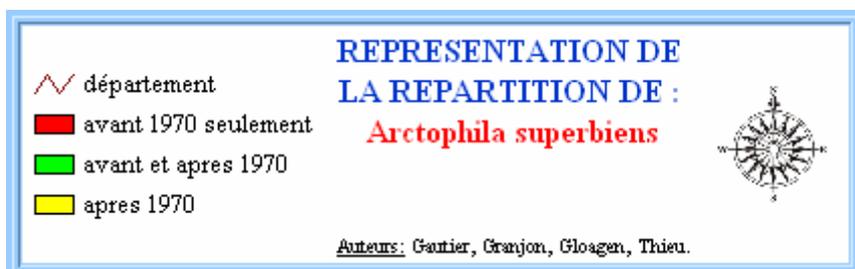
Tout d'abord la carte de répartition de l'espèce suivant les critères suivants :

- en jaune , si les observations sont uniquement antérieur à 1970 ;
- en rouge , si les observations sont uniquement postérieur à 1970 ;
- en vert , si les observations ont eu lieu avant et après 1970.



Implicitement toutes les classes ne sont représentées selon l'espèce sélectionnée, et les départements restant non colorés impliquent l'absence d'observation de l'espèce.

Un bloc légende accompagne la carte, rappelant la légende de l'analyse thématique et le nom de l'espèce sélectionnée :



Ici, le bloc légende correspondant à la carte présentée ci-dessus, c'est-à-dire à la représentation d'Arctophila superbiens.

Enfin, des informations et illustrations sont disponibles sur l'espèce sélectionnée :

Synonyme de l'espèce:

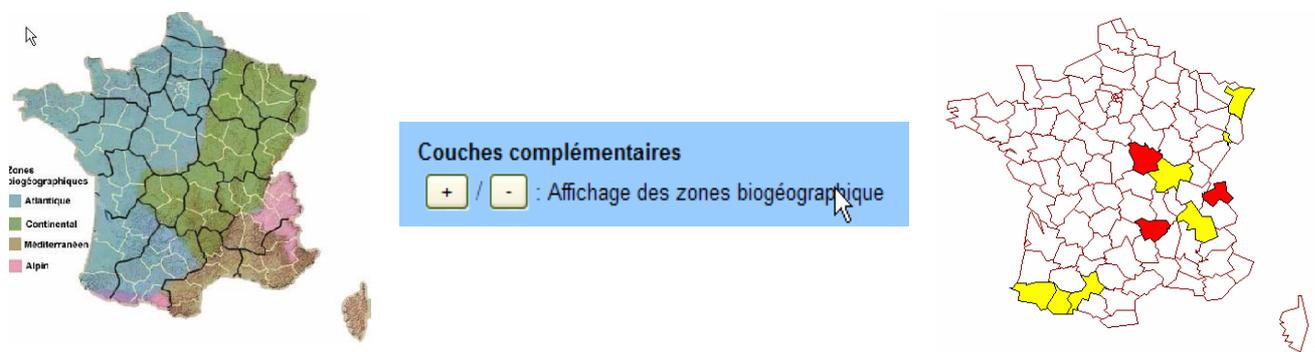
- ◆ E. liophthalma

1 : à surveiller
2 : en diminution
3 : menacée

Parmi ces informations, l'utilisateur découvrira une illustration de l'espèce, des pictogrammes indiquant le statut accompagné d'une légende, et aussi une liste des noms d'espèce synonymes de la sélection.

3.2.2 Autres informations disponibles

L'utilisateur peut superposer à la représentation de la répartition d'une espèce, l'identification des zones biogéographiques. Ceci à l'aide des contrôles suivants :



L'intérêt de cette fonction, est de pouvoir alterner rapidement (aucun rafraîchissement de la page), les deux représentations cartographiques, afin de réaliser des analyses comparatives.

L'utilisateur peut aussi obtenir plus d'informations sur l'espèce, grâce au bouton .

Une page s'ouvre dans une nouvelle fenêtre afin que l'utilisateur puisse comparer ce nouveau document et l'information de la carte précédemment créée :



:: Genre : **Epistrophe**

:: Espèce : **lelophthalma**

:: Découvreur : **Schiner & Egger**

:: Date de nommage : **1853**

Synonyme de l'espèce :

Nom	Découvreur	Date
E. liophthalma	Schiner & Egger	

Information sur l'Observation de **Epistrophe lelophthalma**

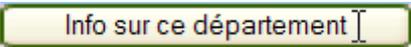
Localisation	Observée avant 1970 <small>AUTEUR(S) - Publication - (Observation)</small>	Observée après 1970 <small>AUTEUR(S) - Publication - (Observation)</small>
📍 Hautes-Alpes (5)	SPEIGHT M.C.D., 1993 - (1924, 1950) 1366	
📍 Hautes-Alpes (5)	SPEIGHT M.C.D., 1993 - (1924, 1950) 1366	
📍 Hautes-Alpes (5)		SPEIGHT M.C.D., 1993 - (1990) 1367
📍 Pyrenees-Orientales (66)	SARTHOU J.P., SPEIGHT M.C.D., 1997 - (1955) 1368	
📍 Haute-Savoie (74)	AUBERT J., AUBERT J.-J., GOELDIN DE TIEFENAU P., 1976 - (1964, 1967, 1968) 1369	
📍 Haute-Savoie (74)	AUBERT J., AUBERT J.-J., GOELDIN DE TIEFENAU P., 1976 - (1964, 1967, 1968) 1369	
📍 Haute-Savoie (74)	AUBERT J., AUBERT J.-J., GOELDIN DE TIEFENAU P., 1976 - (1964, 1967, 1968) 1369	
📍 Haute-Savoie (74)		SPEIGHT M.C.D., 1993 - (1990) 1370
📍 Haute-Savoie (74)		DIRICKX H.G., HAMON J., STEFFEN J., 1996 - (1990) 1370

Ce document appelé « fiche informative sur l'espèce », précise des informations déjà présentes au niveau de l'interface, notamment les dates et découvreurs de l'espèce et de ces synonymes.

Un tableau reprend ensuite la liste des départements où l'espèce a été observée, et identifie les auteurs des différentes publications faisant état de ces observations, tout en affichant les dates d'observations et de publication.

L'étiquette située près de chaque département permet d'ouvrir la « fiche informative sur le département » (explication partie suivante). En effet, l'utilisateur peut obtenir la liste de toutes les espèces observées dans un département.

Cette fiche est accessible de deux façons :

- soit par sélection d'un département sur la carte, puis validation de cette sélection à l'aide du bouton 
- soit par sélection d'un département dans la fiche informative sur l'espèce, à l'aide du bouton  situé à gauche du libellé département.

Cette fiche informative sur le département présente sur le même principe que la fiche informative sur l'espèce, la liste des espèces observées pour le département sélectionné.

Observations réalisées dans le département : 74		
Genre Espece	Observée avant 1970 AUTEUR(S) - Publication - (Observation)	Observée après 1970 AUTEUR(S) - Publication - (Observation)
 Anasimya lineata		DIRICKX H.G., HAMON J., STEFFEN J., 1996 - (1990-1995)
 Anasimya lineata		DIRICKX H.G., HAMON J., STEFFEN J., 1996 - (1990-1995)
 Arctophila bombiforme		DIRICKX H.G., HAMON J., STEFFEN J., 1996 - (1990-1995)
 Arctophila bombiforme		DIRICKX H.G., HAMON J., STEFFEN J., 1996 - (1990-1995)
 Baccha elongata	TIMON-DAVID J., 1937 - (1937)	
 Blera fallax	AUBERT J., AUBERT J.-J., GOELDLIN DE TIEFENAU P., 1976 - (1962, 1965)	
 Blera fallax	AUBERT J., AUBERT J.-J., GOELDLIN DE TIEFENAU P., 1976 - (1962, 1965)	
 Brachyopa scutellaris		DIRICKX H.G., HAMON J., STEFFEN J., 1996 - (1990-1995)
 Brachyopa scutellaris		DIRICKX H.G., HAMON J., STEFFEN J., 1996 - (1990-1995)
 Brachyopa testacea	AUBERT J., AUBERT J.-J., GOELDLIN DE TIEFENAU P., 1976 - (1967)	

Le bouton  permet de s'orienter vers la fiche informative de l'espèce. Ce système de navigation inter-fenêtre doit permettre à l'utilisateur d'approfondir ces recherches sans repasser par une interface principale, mais tout en gardant en arrière plan l'analyse de départ.

IV. L'INTERFACE : CARTE D'EFFECTIFS

4.1 Présentation

L'objectif est de représenter des cartes d'effectifs basées sur la fréquence d'observation de l'ensemble des genres ou de l'ensemble des espèces de syrphes actuellement recensées.

The screenshot displays the 'Carte d'effectifs' interface. On the left is a map of France with departments colored according to the number of observed species before 1970. A legend below the map shows five categories: 1-10 (lightest yellow), 11-50, 51-100, 101-200, and plus de 200 (darkest yellow/brown). The title of the map is 'CARTE D'EFFECTIFS DES espèces observées seulement avant 1970'. Below the legend, it lists authors: 'Auteurs: Garnier, Granjon, Gloaguen, Thieu.' and includes a compass rose.

On the right, the control panel includes:

- Sélection genre/espece:** A dropdown menu currently set to '--Genres ou Espèces--'.
- Couches complémentaires:** '+ / - : Affichage des zones biogéographiques'.
- Informations complémentaires:** 'Presser Ctrl pour sélectionner plusieurs départements dans la liste puis "Lancer"'. Below this is a list of departments: '1 - Ain', '2 - Aisne', '3 - Allier', '4 - Alpes-de-Haute-Provence', '5 - Hautes-Alpes'. A 'Lancer' button is next to the list.
- Sélectionner un département sur la carte:** A button labeled 'Info sur ce département'.
- Aide:** A question mark icon with the text ': Aide'.

4.2 Recherche thématique

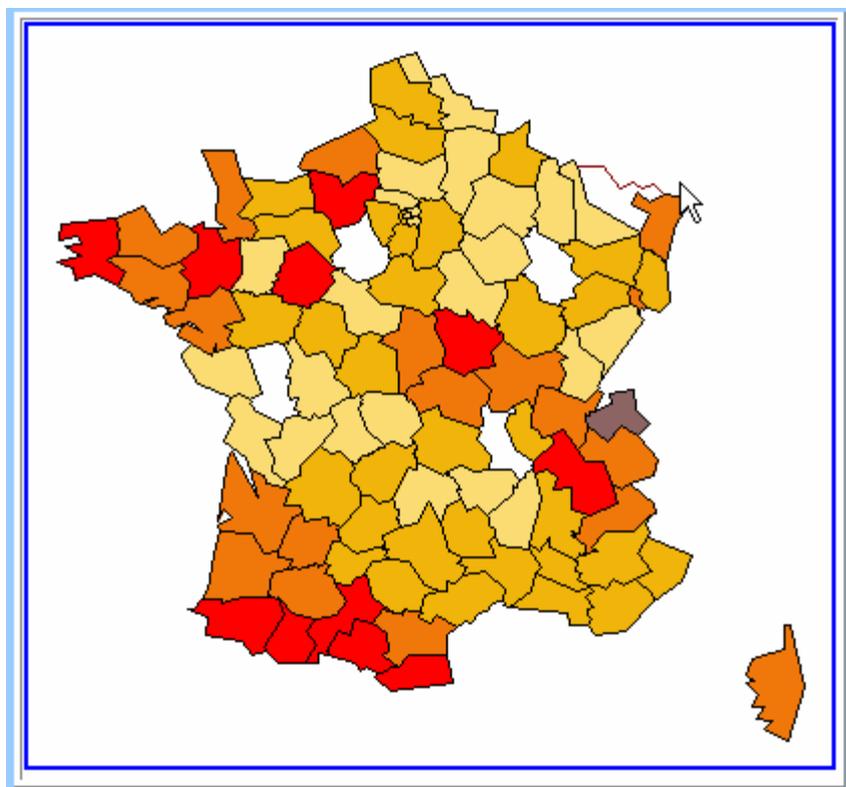
4.2.1 Fonctionnement

L'interface propose une liste déroulante permettant à l'utilisateur de choisir l'information à représenter :

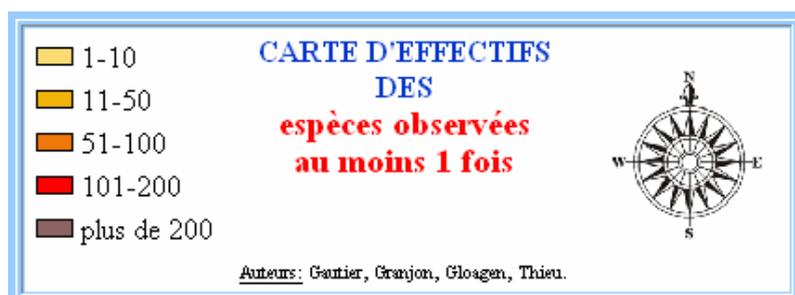
This close-up shows the dropdown menu for 'Sélection genre/espece'. The menu is open, showing the following options:

- Genres ou Espèces-- (selected)
- Genres ou Espèces---
- C Espèces observées au moins 1 fois
- C Espèces observées avant et après 1970
- C Espèces observées seulement avant 1970
- C Espèces observées seulement depuis 1970
- lr Genres observés au moins 1 fois
- lr Genres observés avant et après 1970
- lr Genres observés seulement avant 1970
- p Genres observés seulement depuis 1970

La sélection dans la liste déroulante entraîne l'interrogation de la base de données, et un comptage des genres ou espèces selon le critère d'observation spécifié (au moins une fois, avant et après 1970, seulement avant 1970, seulement depuis 1970,...).



Ci-contre le type d'analyse thématique obtenue pour les espèces observées au moins une fois. Par défaut, c'est la sélection retenue à l'ouverture de l'interface cartographique.



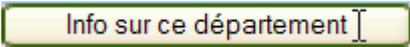
Ci-contre la légende qui rappelle le type de sélection effectué, ainsi que les classes choisies pour représenter la thématique.

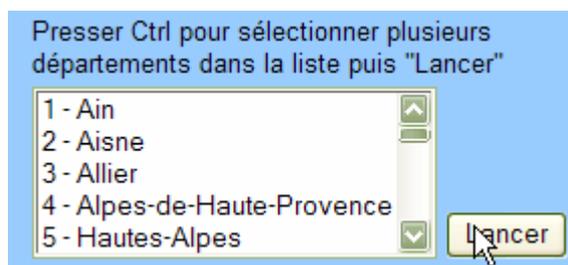
On remarquera que ces classes ont été définies et ne sont pas modifiable par l'utilisateur.

4.2.2 Informations complémentaires

On retrouve des fonctionnalités communes avec l'interface précédente, tel que l'affichage des zones biogéographiques.

Logiquement, seule la fiche informative sur le département est accessible depuis cette interface. Cependant l'originalité est de pouvoir choisir deux types de sélection :

- soit une sélection unique de façon similaire à l'interface précédente, c'est à dire par sélection sur la carte puis validation par le bouton .
- Soit une sélection multiple, par choix dans la zone de texte déroulante (la sélection multiple doit être effectuée par « ctrl+clic ») et valider par le contrôle prévu à cet effet..



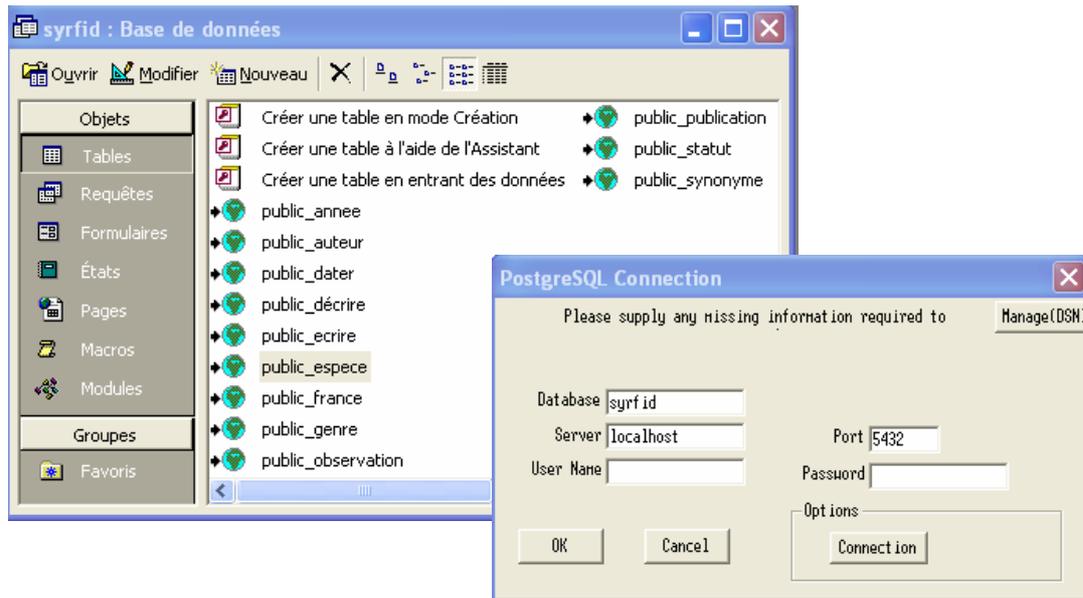
La structure de la fiche informative sur le département, reste identique quelque soit la sélection (simple ou multiple), et la navigation vers les fiches informatives sur les espèces restent disponible.

V. L'INTERFACE : ADMINISTRATION ET EXPORTATION DE LA BASE

5.1 Présentation

Le rôle du SGBD consiste à saisir et entretenir les données thématiques intégrées dans le système. Maintenant que les données sont intégrées à la base sous PostgreSQL, il faut gérer l'administration des données.

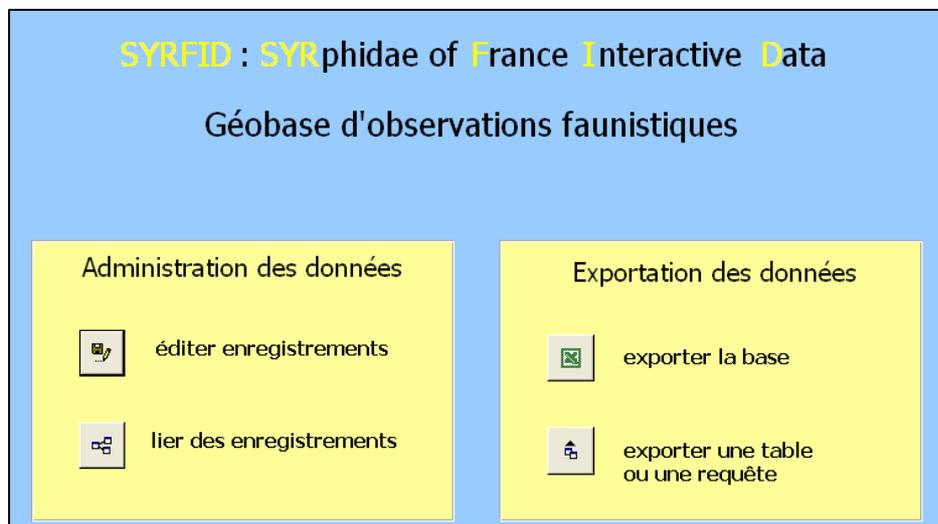
La gestion des données se fait par l'intermédiaire d'une interface sous le SGBD Microsoft Access, la base syrfid.mdb étant liée en permanence avec la base PostgreSQL grâce à une connexion ODBC. Pour se connecter, l'utilisateur doit entrer son nom d'utilisateur et son mot de passe.



L'interface est inspirée des pages web avec une arborescence des formulaires hiérarchisée (page d'accueil, bouton retour pour la navigation, etc). Elle fonctionne au clic-bouton et dès qu'une action est réalisée (ajout, modification, exportation), l'utilisateur est informé par une boîte de message.

L'interface est divisée en deux parties bien distinctes :

- une partie administration de la base qui permet d'ajouter ou de modifier un enregistrement et de lier les enregistrements
- une partie exportation des données qui propose à l'utilisateur d'exporter la base ou seulement certaines tables ou requêtes



5.2 Administration des données

Le formulaire intitulé *Administration des données* sur la page d'accueil concerne la gestion des données. L'activation du premier bouton de ce formulaire fait découvrir un sous-formulaire, qui laisse le choix à l'utilisateur d'ajouter un nouvel enregistrement ou de modifier les données déjà présentes dans la base. Le second permet de lier des enregistrements entre eux.

5.2.1 Ajout d'un enregistrement

Cliquer sur *éditer enregistrements* puis sur *ajouter un enregistrement*. Dans un premier temps, l'utilisateur doit choisir le type d'enregistrement qu'il souhaite ajouter. Il accède ensuite à un formulaire de saisie où il doit remplir les champs.

Pour le champ numéro, qui correspond à la clé primaire de chaque table, la numérotation automatique n'est pas conservée sous Access. L'utilisateur doit rentrer manuellement l'identifiant du nouvel enregistrement, il est donc impératif pour lui de connaître le nombre d'enregistrement de la table concernée.

Certains champs apparaissent sous forme de liste déroulante, il s'agit des clés étrangères de la table, dans ce cas l'utilisateur doit choisir la valeur du champ parmi une liste de valeur prédéfinie.

Si l'ajout du nouvel enregistrement s'effectue avec succès, un message d'information prévient l'utilisateur.

The image shows two screenshots of a web application interface. The top screenshot, titled "Ajouter un enregistrement", displays a grid of six buttons, each with a right-pointing arrow and a plus sign, labeled: "un genre", "une espèce", "un synonyme", "une observation", "une publication", and "un auteur". Below this grid is a button with a downward arrow and a plus sign. The bottom screenshot, titled "Ajouter une espèce", shows a form with the following fields: "numéro:" (text input), "codeE(4):" (text input), "nom:" (text input), "genre:" (dropdown menu), "découvreur:" (text input), "date de nommage:" (text input), "photo:" (text input), "miniature:" (text input), "statut français:" (dropdown menu), "statut européen:" (dropdown menu), "importance internat*:" (text input), and "UEF:" (text input). At the bottom of the form are two buttons: "valider" and "annuler".

5.2.2 Modification d'un enregistrement

Cliquer sur *éditer enregistrements* puis sur *modifier un enregistrement*. Comme dans le menu d'ajout, l'utilisateur doit choisir le type d'enregistrement qu'il souhaite modifier. Il accède ensuite à une liste des enregistrements où il peut modifier chacun des champs.

Modifier un enregistrement

 un genre

 une espèce

 un synonyme

 une observation

 une publication

 un auteur



numéro	codeE(4)	nom_espece	decouvreur	date nommage	photo	miniature	UEF	importance	inter.	statut fra
292	Equ2	equestris	Sack	1913	MeroEqu2.jpg	XyloSegn_t.jpg	1			
311	Crab	crabroniformis	Fabricius	1775	MleCrab.jpg	XyloAbie_t.jpg	1	1		S
313	Flor	florea	L.	1758	MyalFlor.jpg	VoluZona_t.jpg	1			
337	Bico	bicolor	Fabricius	1794	Par1Bico.jpg	VoluInfl_t.jpg	1			
429	Frost	rostrata	L.	1758	RhinFrost.jpg	volubomb_t.jpg	1			M
434	Meco	mecogramma	Bigot	1860	ScaeMeco.jpg	TemnBomb_t.jpg	1			D
436	Sele	selenica	Meigen	1822	ScaeSele.jpg	SphaTaen_t.jpg	1			
437	Lapp	lappona	L.	1758	SenLapp.jpg	SphaScri_t.jpg	1			
452	Scri	scripta	L.	1758	SphaScri.jpg	SenLapp_t.jpg	1			
454	Taen	taeniata	Meigen	1822	SphaTaen.jpg	ScaeMeco_t.jpg	1			
482	Bomb	bombylans	Fabricius	1805	TemnBomb.jpg	RhinFrost_t.jpg	1			S

Enr : 500 sur 511

5.2.3 Liaison des enregistrements

Cliquer sur *lier les enregistrements*. Ce menu offre la possibilité à l'utilisateur de relier des enregistrements entre eux. On peut lier une publication à un auteur et une observation à une publication et à une année. Une publication peut avoir été écrite par plusieurs auteurs, dans ce cas là, il faut répéter l'action pour chacun des différents auteurs. De même, une observation peut avoir fait l'objet de plusieurs publications, il faut alors lier l'observation à chacune des publications la concernant. Si la liaison s'effectue avec succès, un message d'information s'affiche pour prévenir l'utilisateur.

Lier des enregistrements

 lier une publication à un auteur

 lier une observation à une publication

 lier une observation à une année

Lier une publication à un auteur

code de publication :

nom de l'auteur :

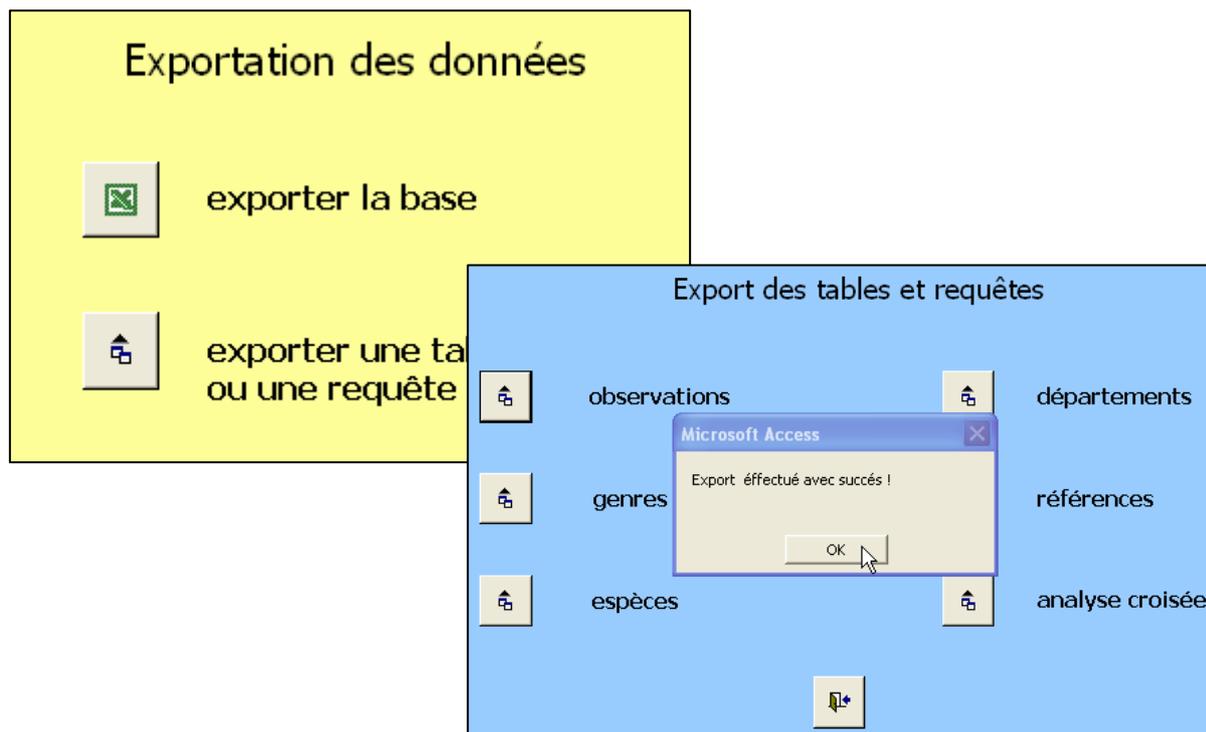
position de l'auteur :



5.3 Export des données vers Excel

La partie exportation des données propose à l'utilisateur de travailler avec les données de la base sous Excel. Cette export offre à l'utilisateur de nouvelles possibilités de travail sur les données. Il peut analyser les données, effectuer des calculs, générer des graphiques, ce qu'il ne pourrait pas faire sous un SGBD.

Il a le choix d'exporter les données dans leur globalité en cliquant sur *exporter la base* dans la fenêtre d'accueil ou de récupérer seulement un genre précis de données en choisissant *exporter une table ou une requête*. Si l'exportation des données se passe bien, une boîte de message prévient l'utilisateur. Le fichier syrfid.xls, résultant de l'exportation, se trouve au chemin d'accès suivant C:\Access.



Guide de maintenance

SYRFID

Presentation Documentation Serveur cartographique Lien

SYRFID v3.0

SYRphidae of France Interactive Data

Données interactives sur les observations des espèces
de DIPTERA SYRPHIDAE et MICRODONTIDAE en France

INP ENSAT

Site réalisé par Mlle Gautier, Mr Gloaguen, Mr Granjon et Mr Thieu dans le cadre du projet de fin d'étude du DESS SIGMA, en collaboration avec Mr Monteil, Mr Sarthou et Mr Souques.

MapServer PostgreSQL PostGIS

Référence à utiliser pour citer SYRFID
SARTHOU J.P., MONTEIL C. et DUSSAIX C., 2005
SYRFID v3.0 : Syrphidae of France Interactive Data [On-Line URL : <http://syrfid.ensat.fr/>].
Dernière mise à jour : 20 février 2005
© SYRFID 2005. INP/ENSAT

GAUTIER Bénédicte
GLOAGUEN Jérémy
GRANJON Ludovic
THIEU Vincent



Responsables : Claude MONTEIL et Marc SOUQUES

Fevrier 2005

SOMMAIRE

I. INTRODUCTION	3
II. LE SERVEUR CARTOGRAPHIQUE	4
2.1 Quels logiciels « open source » utilisés ?	4
2.2 Installation	4
2.3 Création de cartes dynamiques	6
III. LA BASE DE DONNEES	9
3.1 Le Modèle Conceptuel de Données : modification	9
3.2 Migration de la base de données Excel	10
3.3 Actualisation	10
3.4.1 Choix de la solution	10
3.4.2 Mise en place de la connexion ODBC	11
3.4.3 Création de l'outil de gestion de données	12
3.4.4 Export au format Excel	14
IV. LES INTERFACES CARTOGRAPHIQUES : retour sur le fonctionnement	15
4.1 Fonctionnement et relation entre les contrôles de sélection	15
4.2 Les zones d'affichages d'informations dynamiques	17
4.3 Les liens statiques et dynamiques	17
4.4 Les éléments javascript	18
V. LES PERSPECTIVES	19
5.1 La sélection	19
5.2 Interface : carte de répartition	19
5.3 La partie administration : Access	20
ANNEXES	
Le dictionnaire de données	21
Le diagramme des classes de PhpMapScript	25
Exemple de script pour la création d'un fichier map	30

I. INTRODUCTION

Ce guide de maintenance a pour but d'expliquer, de diriger l'utilisateur qui souhaite mettre à jour la base de données, les interfaces Web et/ou l'interface administration sur ACCESS.

L'utilisateur doit de disposer d'informations suffisantes pour modifier, créer efficacement de nouvelles fonctionnalités.

Dans le développement d'un outil, d'un logiciel ou d'une base de données, la phase d'actualisation reste un des éléments fondamentaux.

C'est pourquoi, vous trouverez dans ce guide,

1. la présentation de chaque point essentiel de ce projet de « Base de données biogéographiques sur Internet »
2. la méthodologie utilisée
3. les possibilités et moyens de mise à jour.

Partant d'une base de données complexe, créée sous Excel, et d'un site Web – prototype : [http://syrfid.ensat.fr/ Syrfid.htm](http://syrfid.ensat.fr/Syrfid.htm), notre travail a donc été important pendant ces quelques semaines de projet.

Vous trouverez dans ce guide de maintenance, les points que nous avons gérés et dans une dernière partie, les possibilités pour améliorer le site, les interfaces cartographiques réalisées.

II. LE SERVEUR CARTOGRAPHIQUE

Pour que le site SYRFID v 3.0 soit disponible sur Internet, nous avons besoin d'installer un certain nombre de composants sur un ordinateur sur lequel on pourra accéder à distance.

Nous avons choisi d'installer un serveur sur un ordinateur disposant d'un système d'exploitation Windows Server 2003. Cette solution a été retenue pour sa facilité d'installation des composants en comparaison d'une installation sous Linux.

2.1 Quels logiciels « open source » utilisés ?

Sur cet ordinateur, nous avons dû installer les composants suivants :

- Un **serveur Web** : Apache 2.0.50
- Un **langage de script** pour le Web dynamique : PHP 4.3.7 avec une extension PHP/MapScript 4.4.1
- Un **serveur cartographique** : Mapserver 44
- Une **Base de données** : PostgreSQL 8.0.0 beta 2
- Une **Base de données spatiale** : PostGIS 0.8 (extension ou cartouche spatiale de PostgreSQL)
- Une **interface pour PostgreSQL** : phpPgAdmin 3.5.1

2.2 Installation

- Dans un premier temps, nous allons installer **Apache**, **PHP** (et **PHP MapScript**) et **Mapserver**.

Ces trois composants ont été installés à partir d'un package développé par DM Solutions Group (<http://www.dmsolutions.ca/>) dont le nom est ms4w. Celui-ci est disponible à l'adresse suivante : http://www.maptools.org/dl/ms4w/ms4w_1.0.3.zip.

Après l'avoir téléchargé, il faut le « dézipper » sur la racine du disque dur afin d'obtenir une arborescence de ce type : c:\ms4w\.

Dans le dossier ms4w, il faut cliquer sur le fichier `apache-install.bat` pour installer ce package.

Pour pouvoir utiliser PHP avec PostgreSQL, il faut activer l'extension `php_pgsql.dll`. Pour cela, il faut enlever le point-virgule «;», qui se trouve dans le fichier `php.ini`, situé dans `C:\ms4w\Apache\cgi-bin` devant la ligne `extension=php_pgsql.dll`

- Dans un deuxième temps, nous allons installer **PostgreSQL** et **PostGIS**. Pour cela, il faut télécharger le fichier PostgreSQLWin32 à l'adresse suivante : <http://www.01map.com/download/>, puis le décompresser sur la racine du disque dur afin d'obtenir une arborescence du type `c:\PostgreSQLWin32\`.

Il faut ensuite **créer les variables d'environnement** (clique droit sur le poste de travail et Propriétés puis sur l'onglet Avancé et Variables d'environnement).

Dans la partie Variables systèmes, faites « nouveau » puis définissez successivement :

- `PGHOME = c:\PostgreSQLWin32`
- `PGDATA = c:\PostgreSQLWin32\data`
- `PGHOST = localhost`
- `PGPORT = 5432`

Ensuite, recherchez la variable d'environnement nommée `Path` et la modifier en insérant au début :
`c:\PostgreSQLWin32\bin;c:\PostgreSQLWin32\lib;...` (les « ... » correspondent à la suite de la variable `Path`).

Vous devez maintenant redémarrer votre ordinateur afin que ces variables soient prises en compte. Ces variables d'environnement permettent à PostgreSQL de savoir où se trouve la racine de la distribution (`PGHOME`). Il saura également que le répertoire pour vos futures bases est `PGDATA` et que le port de connexion est le 5432 (`PGPORT`).

PostgreSQL permet l'accès aux données à partir d'un compte utilisateur et non d'un compte administrateur.

Il faut donc **créer un nouveau compte utilisateur avec un nom et un mot de passe**. Ce nom et ce mot de passe serviront par la suite dans la connexion vers la base de données (dans notre cas : nom d'utilisateur = `syrfid` et mot de passe = `syrfid`).

- Nous pouvons donc maintenant **créer notre base de données**. Il faut ouvrir une session administrateur et lancez l'invite de commandes (Démarrer/ Tous les programmes/ Accessoires/ Invite de commandes).

Taper comme tel : `runas /noprofile /user:nomducompteutilisateur cmd` (pour nous `user:syrfid`) puis tapez le mot de passe. Une nouvelle fenêtre s'affiche vous permettant de travailler en tant qu'utilisateur `syrfid`, donc utilisateur, à partir d'une session administrateur.

A partir de là, plusieurs étapes sont nécessaires. A la ligne de commandes, il faut écrire les instructions suivantes :

- `cd \postgresqlwin32` pour vous rendre dans le dossier PostgreSQLWin32
- `set_env.bat`
- `initdb -A trust`
- `pg_ctl start` pour lancer postgresql
- `createdb syrfid`
- `createlang plpgsql syrfid`
- `psql -d syrfid -c "\i share/postgresql/postgis.sql"`
- `psql -d syrfid -c "\i share/postgresql/spatial_ref_sys.sql"`
- `psql syrfid<c:/postgresqlwin32/syrfid.sql` (dans ce cas, le fichier de définition de la base de données se trouve dans `c:/postgresqlwin32` et se nomme `syrfid.sql`)
- `pg_ctl stop` pour arrêter postgresql

Maintenant, la base de données `syrfid` existe et contient l'ensemble des tables et des vues, vous pouvez l'exporter en tapant la ligne de `pg_dump nomdelabasededonnées>emplacementdel'export/nom.sql` après avoir lancé `pg_ctl start`

Pour **accéder à la base de données**, il suffira de lancer l'invite de commande, de taper le nom et mot de passe choisi et de taper dans la nouvelle fenêtre directement : `postmaster`.

```
Microsoft Windows XP [version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\WINDOWS\system32>postmaster
LOG: database system was interrupted at 2005-02-22 16:22:14 Paris, Madrid
LOG: checkpoint record is at 0/3C95720
LOG: redo record is at 0/3C95720; undo record is at 0/0; shutdown TRUE
LOG: next transaction ID: 35858; next OID: 259897
LOG: database system was not properly shut down; automatic recovery in progress

LOG: record with zero length at 0/3C95760
LOG: redo is not required
LOG: database system is ready
```

- Maintenant que la base de données est créée, nous allons installer **phpPgAdmin** afin de pouvoir visualiser la base de données de façon plus attractive qu'à la ligne de commande. Pour cela, il faut décompresser le fichier `phpPgAdmin-3.5.1.zip`, disponible à l'adresse <http://phppgadmin.sourceforge.net/>, dans le répertoire `C:\ms4w\Apache\htdocs`.

Pour accéder à la base de données « syrfid », dans notre cas, choisissez le navigateur Internet de votre choix (Internet Explorer, Mozilla,...), tapez <http://localhost/phpPgAdmin/>, puis insérer votre nom d'utilisateur et mot de passe (ATTENTION les majuscules comptent).

L'installation du serveur cartographique est terminée.

Attention : il faut modifier la ligne 325 du fichier `carte_effectif.php` et la ligne 244 du fichier `carte_repartition.php` (`width."px' height=".$map->height."px" ;?> USEMAP=#ImageMap' name="MON_IMAGE"/>`), en remplaçant `localhost` par le numéro IP du serveur : `194.254.144.40`.

2.3 Création de cartes dynamiques

Pour permettre de réaliser des cartes de façon dynamique, nous avons utilisé MapServer couplé à PostgreSQL/PostGis.

Le principe de fonctionnement de MapServer est basé sur des fichiers `map`. Ces fichiers contiennent l'ensemble des informations nécessaires à MapServer pour créer une carte.

Ci-dessous un exemple de fichier `map` (source : *DESS Sigma : Module 233, Cartographie et SIG interactifs en ligne, Séance 2 : Introduction aux scripts MapServer MapScript et PHPMapScript, JEGOU L.*)

```

Le fichier map :

NAME "Europe en Bleu"           Nom de la carte
SIZE 400 400                    Taille de l'image à générer
STATUS ON                       Statut d'activation
EXTENT -31.2467 27.6065 40.1199 80.762  Extension du fond, coords. mini/maxi
UNITS METERS                    Unité : mètres
SHAPEPATH "."                  Chemin du répertoire contenant le fichier shape
WEB                              Section Web
    IMAGEPATH 'C:/ms4w/tmp/ms_tmp/'  Chemin complet du répertoire de l'image
    IMAGEURL  '/ms_tmp/'            Chemin relatif de l'image
END                              Fin section

LAYER                            Début section layer (couche)
    NAME "Europe"                Nom de la couche
    TYPE POLYGON                 Type d'objets
    STATUS ON                    Affichage : oui
    DATA "europe"               Source des données : nom du fichier .shp
    CLASS                        Début section Class
        STYLE                    Début section Style
            COLOR 20 10 110      Couleur de fond des polygones
            OUTLINECOLOR 200 200 200  Couleur contour
        END                      Fin section Style
    END                          Fin section Class
END                              Fin section Layer

END

END                              Fin mapfile
    
```

Ce fichier map est ensuite appelé par un fichier html/php qui aura pour but de dessiner la carte à partir des informations du fichier map et de l'enregistrer dans un répertoire connu pour pouvoir par la suite l'afficher à l'écran.

Voici un exemple de ce type de fichier (*source : DESS Sigma : Module 233, Cartographie et SIG interactifs en ligne, Séance 2 : Introduction aux scripts MapServer MapScript et PHPMapScript, JEGOU L.*):

Le fichier html/php :

```

1 <?php                          1 <? Début de script php
2 dl('php_mapscript_44');        2 chargement du module phpMapScript
3                                3
4 $map = ms_newMapObj("europe.map");  4 variable de l'objet map : init et chargement
5 $image=$map->draw();            5 variable d'image remplie par dessin fichier map
6 $image_url=$image->saveWebImage();  6 sauvegarde de la variable image dans un fichier
7 ?>                              7 fin de script php.
8 <HTML>                          8 début document HTML
9   <HEAD>                          9   en-tête
10   <TITLE>Exemple : Afficher un shp</TITLE>  10   titre doc HTML
11 </HEAD>                          11   fin en-tête
12 <BODY>                           12   début corps
13   <IMG SRC=<?php echo $image_url; ?> >  13   balise image et source variable php
14 </BODY>                           14   fin corps
15 </HTML>                          15 fin document HTML
    
```

Le fichier map vu précédemment ne permet pas de créer des cartes dynamiques, les informations contenues ne sont pas dynamiques.

Dans notre cas, nous souhaitons créer des cartes de manière dynamique. Pour cela, nous allons écrire la quasi-totalité du fichier map à partir d'un fichier php contenant des variables.

Le principe est basé sur le langage PhpMapscript, qui nous permet de générer dans du php la définition complète d'un fichier map (fichier utilisé par Mapserver pour afficher une carte), à partir du diagramme des classes de PhpMapScript (cf. annexe).

Pour cela, il est nécessaire de créer un fichier map très simple, contenant ces informations de base :

```

MAP
    EXTENT 0 0 1 1
    SIZE 1 1
    IMAGECOLOR 255 255 255
    IMAGETYPE PNG
END
    
```

Ce fichier est nécessaire à PhpMapscript qui va pouvoir à partir de ce fichier écrire un fichier map complet. (Voir exemple de script en annexe)

Nous allons donc nous connecter à la base de données puis récupérer toutes les informations nécessaires pour la création de notre carte.

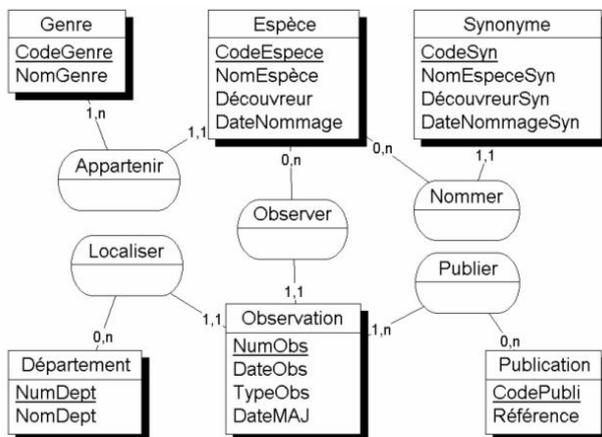
Il faut prédéfinir les différentes couches de la carte (layer) qui seront également modifiées selon les requêtes effectuées via le site internet.

Ainsi nous pouvons à la fois créer des cartes en fonction de requêtes pilotées par le site Internet mais ces requêtes sont en plus mises à jour au fur et à mesure que la base de données est mise à jour.

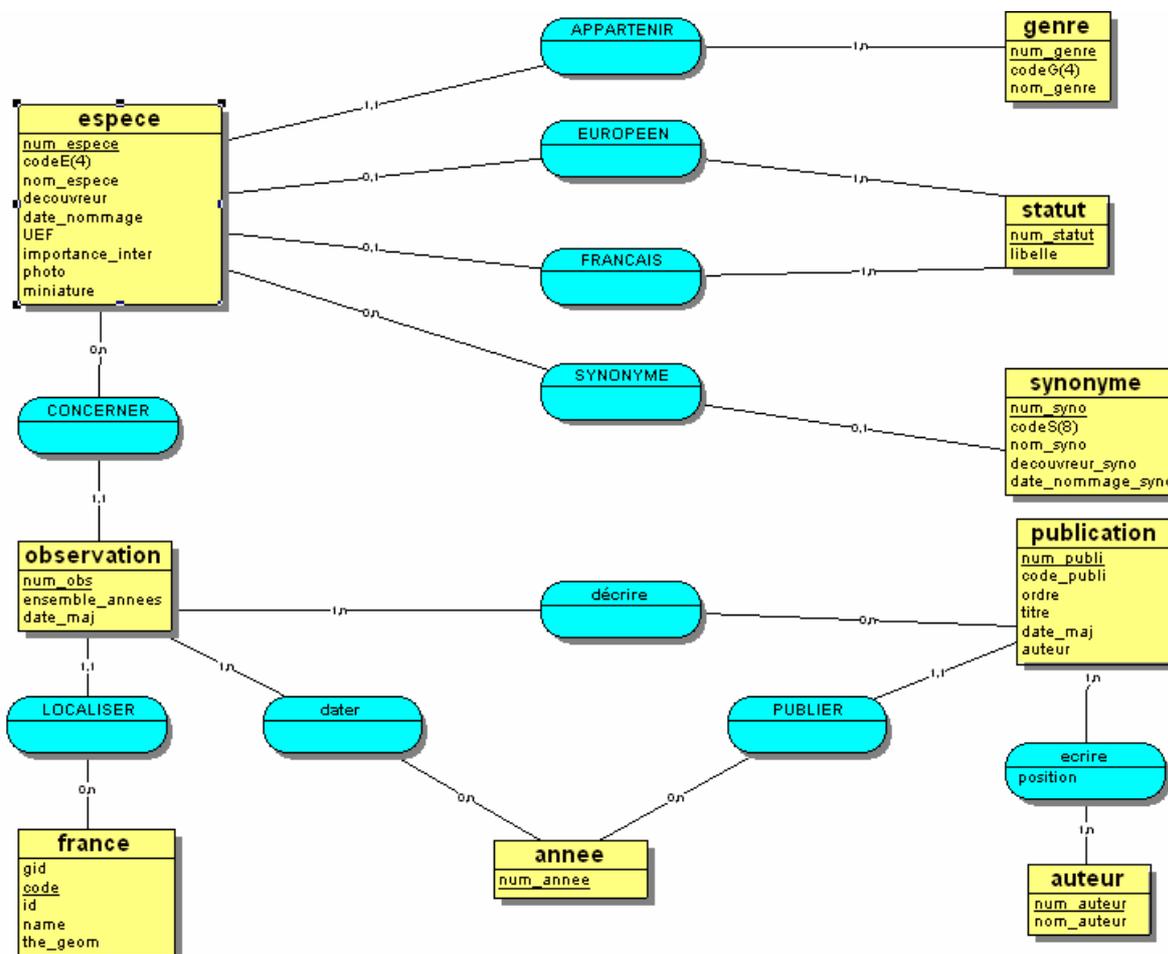
III. LA BASE DE DONNEES

3.1 Le Modèle Conceptuel de Données : modification

A l'origine, les données étaient structurées de manière à répondre au Modèle Conceptuel de Données suivant :



Pour conserver l'ensemble des informations et pouvoir répondre aux spécifications du site Web, nous avons dû modifier le MCD, pour aboutir à celui-ci :



3.2 Migration de la base de données Excel

En réalité, les données de départ n'étaient pas structurées dans un Système de Gestion de Base de Données, elles étaient gérées sous Excel dans différentes feuilles.

Cela nous a posé des problèmes pour la récupération des données et l'intégration dans PostgreSQL. Cette intégration s'est réalisée en plusieurs étapes :

- Tout d'abord, nous avons **traité les feuilles Excel** pour lesquelles correspondent aux tables de notre MCD. Pour cela, nous avons :
 1. récupérer les feuilles Excel quasiment à l'identique ou en sélectionnant qu'une partie,
 2. pour certaines tables, nous avons du créer des applications simples en Visual Basic pour Application, afin d'automatiser en partie la création des tables. Par exemple, nous avons utilisé cette méthode pour la table dater, en dissociant les dates de la feuilles Excel afin d'affecter chaque date à un numéro d'observation.
- Puis, nous avons pu **intégrer nos nouvelles tables dans Access** par les fonctions habituelles d'importation pour avoir une même visualisation des données et des relations entre les tables.
- Pour **transférer les tables Access en tables PostgreSQL**, nous avons utilisé un utilitaire gratuit : ODBC Converter avec le pilote ODBC de PostgreSQL. Cet utilitaire est disponible à l'adresse suivante : http://membres.lycos.fr/eazygem/ODBC_HTML/Francais/ODBC_Converter.zip et le pilote est à télécharger sur : <http://www.postgresql.org/ftp/odbc/versions/dll/>
- Enfin, nous avons défini dans PostgreSQL les **clés primaires et étrangères des tables à partir de phpPgAdmin** en créant des contraintes soit de type clé primaire, soit de type clé étrangère.

L'ensemble des tables est donc implémenté dans PostgreSQL afin de vérifier le MCD que nous avons conçu et les contraintes d'intégrité qui en découlent.

Le dictionnaire de données associé à notre de base de données est à consulter dans les ANNEXES.

3.3 Actualisation

3.4.1 Choix de la solution

Le rôle du SGBD consiste à saisir et entretenir les données thématiques intégrées dans le système. Maintenant que les données ont été intégré à la base sous PostgreSQL, il nous faut gérer l'administration des données.

Il n'est pas question d'utiliser l'interface phpPgAdmin pour administrer la base de données, en effet l'ajout et la modification des enregistrements se fais ligne par ligne. Ainsi, la gestion des données par l'administrateur de l'ENSAT doit être simplifiée par la création de formulaire de saisie.

Pour cela, deux solutions s'offrent à nous :

- La création de requête SQL d'ajout, de modification et de suppression (INSERT, UPDATE et DELETE) à partir des données POSTées depuis un formulaire en PHP, puis l'utilisation du tableau associatif \$_POST[] global au site pour récupérer les valeurs ajoutées ou modifiées.

- La mise en place d'une connexion ODBC entre PostgreSQL et Access pour permettre la création de formulaire de saisie directement sous Access.

Nous avons penché pour la seconde solution, bien qu'elle ne soit pas optimale.

La création des formulaires sous Access est beaucoup moins lourde, en terme de temps, que leur développement en PHP. De plus, il nous est demandé de permettre l'exportation des données de la base locale vers le format Excel et le SGBD Microsoft Access l'autorise.

Cependant, la mise en place de la connexion ODBC entre PostgreSQL et Access pose des problèmes d'intégrité référentielle au niveau des liens entre les tables, mais nous verrons cela plus précisément lors de la liaison des tables.

3.4.2 Mise en place de la connexion ODBC

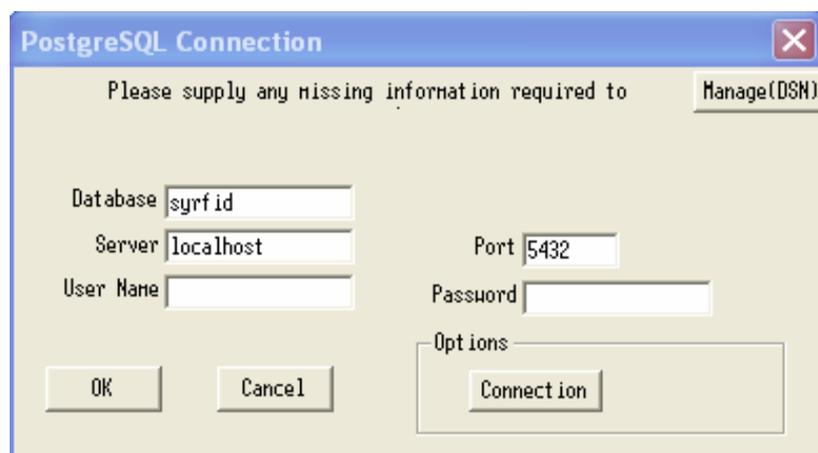
Le pilote ODBC qui permet de faire le lien entre Postgre et Access n'est pas installé par défaut sur les PC, il faut le télécharger sur Internet (pilote psqLODBC distribué par Insight Distribution Systems).

On accède au gestionnaire de source de données fichier en cliquant sur *Fichier / Données externes / lier les tables / Type de fichiers : ODBC Databases()*

On crée une nouvelle source de données fichier que l'on nomme « syrfid », on sélectionne le pilote PostgreSQL qui correspond au fichier PSQLODBC.DLL, puis on enregistre la connexion dans les sources ODBC de la machine.

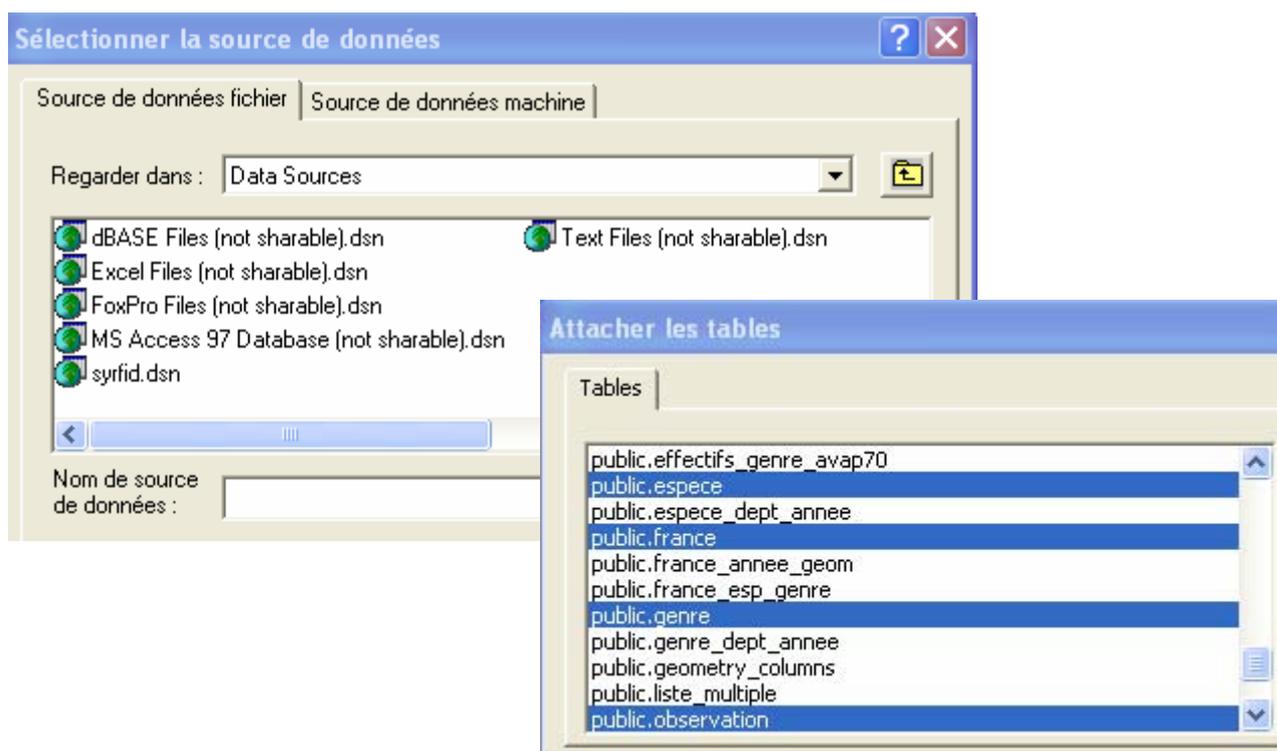
Enfin il nous reste à préciser les éléments nécessaires à la connexion :

- le nom d'utilisateur et le mot de passe
- le nom de la base
- la localisation du serveur
- le port



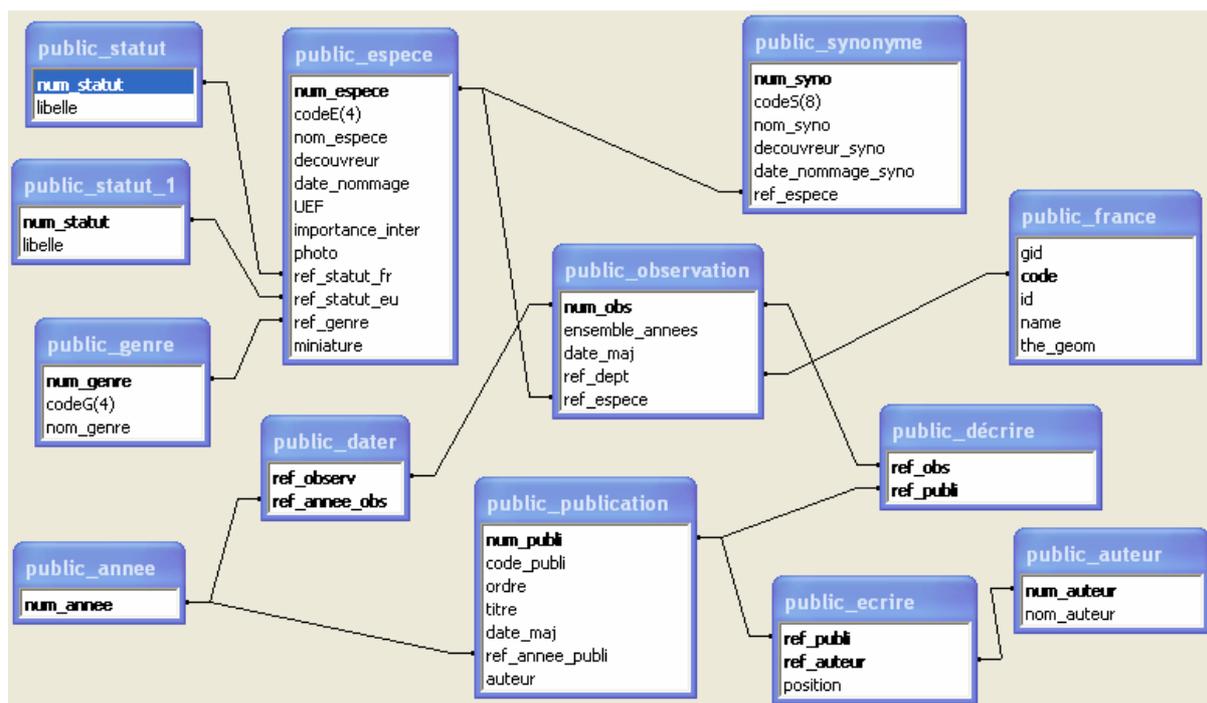
Pour établir le lien, on sélectionne la nouvelle dsn ; on appelle « dsn » la déclaration de la source de données qui sera accessible par l'intermédiaire d'ODBC. Il nous suffit ensuite d'attacher les tables souhaitées.

Précision : La connexion ODBC permet aussi d'attacher les vues créées sous PostgreSQL.



3.4.3 Création de l'outil de gestion de données

Le lien permanent établie entre les deux SGBD conserve les clés primaires et étrangères mais pas les liens entre les tables. Il faut donc commencer par relier les tables. On se rend compte qu'il est impossible d'appliquer l'intégrité référentielle aux relations créées, la mise à jour en cascade des champs et la suppression en cascade des enregistrements ne pourra donc pas se faire automatiquement.



Création des formulaires d'ajout et de modification

On crée des formulaires d'ajout et de modification pour les tables genre, espèce, synonyme, observation, publication et auteur (cf. cours Access de Mr Souques).

Saisie des clés étrangères :

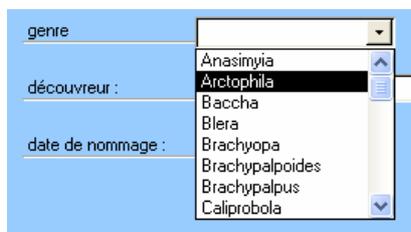
Pour faciliter à l'administrateur de données la saisie dans les clés étrangères des tables, on crée des listes déroulantes. Par exemple, dans la table espèce, on compte trois clés étrangères, ref_statut_fr, ref_statut_eu et ref_genre. Pour la clé étrangère ref_genre qui pointe vers la table genre, on crée une zone de liste modifiable avec les propriétés suivantes :



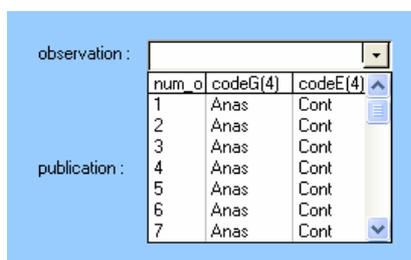
Parmi les deux colonnes (num_genre et nom_genre), la colonne liée est celle qui est gardée dans le contrôle (ici la 1 donc num_genre est stocké dans ref_genre). C'est aussi cette première colonne qui n'est pas affichée dans la liste (0 cm). Dans la propriété Contenu, on crée une requête dont la syntaxe SQL est la suivante :

```
SELECT public_genre.num_genre, public_genre.nom_genre
FROM public_genre
ORDER BY public_genre.nom_genre;
```

On obtient finalement pour la saisie de la clé étrangère ref_genre :



Il nous reste maintenant à créer des formulaires pour les tables d'association décrire, dater et écrire. On utilise la même méthode des listes déroulantes, c'est-à-dire que pour lier une observation à une publication et à une année, on utilise une requête dans la liste déroulante qui va chercher pour un numéro d'observation, le codeG(4) du genre et le codeE(4) de l'espèce.



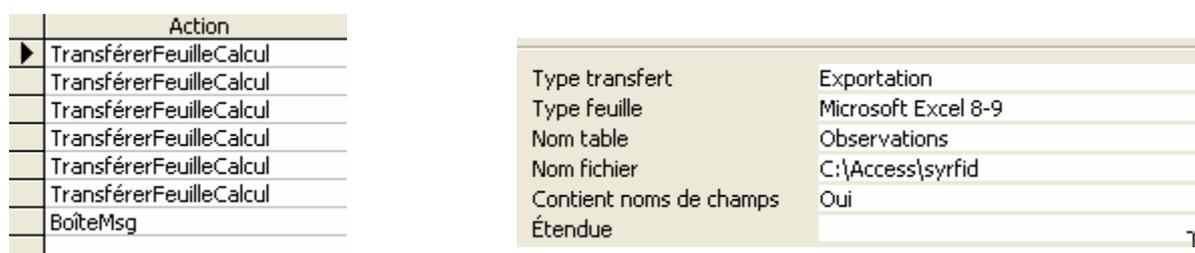
3.4.4 Export au format Excel

Pour obtenir l'information demandée dans les feuilles Excel, on doit dans un premier temps créer des requêtes. On crée ainsi cinq requêtes de sélection, Observations, Genres, Espèces, Départements, Références, et une analyse croisée CptEspDept qui correspondent au feuille Excel à exporter.

On automatise ensuite l'exportation de la base sous Excel en précisant :

- le type de transfert souhaité
- le type de feuille de calcul dans lequel on souhaite exporter les données
- le nom de la table Microsoft Access (dans notre cas il s'agit du nom de la requête) sur laquelle effectuer l'exportation
- le chemin d'accès complet de la feuille de calcul à utiliser pour l'exportation

Pour le dernier arguments de l'action « Contient noms de champs », on sélectionne « oui » pour utiliser les données de la première ligne de la feuille de calcul en tant que nom de champs lors de l'exportation.



On obtient finalement un fichier syrfid.xls :

	A	B	C	D	E	
1	CodeRef	Ordre	Auteur	Année	Auteur principal	Ref_complète
2	Agu1988		Aguilar & Coutin	1988	COUTIN R.	Cheilosia caerulescens Meigen, ravageur
3	And1998		André-Ruiz	1998	ANDREI-RUIZ M. C.	Stratification verticale et diversité des peup
4	Aub1976		Aubert et al.	1976	GOELDLIN DE TIEFENAU P.	Douze ans de captures systématiques de :
5	Bau1993		Bausenwein	1993	BAUSENWEIN D.	Schwebfliegen-Nachweise aus Südfranke
6	Bea1963		Beaucourou & Matile	1963	MATILE L.	Contribution à l'inventaire faunistique des c
7	Bec1910		Becker	1910	BECKER T.	Dipterologische Sammelreise nach Korsici
8	Bra1991		Bradescu	1991	BRADESCU V.	Les Syrphides de Roumanie (Dipt., Syrph.)
9	Bru1990		Brunel & Cadou	1990	CADOU D.	Description de la larve et de la pupa de Ch
10	Bru1998		Brunel et al.	1998	LE VIOL I.	Les Dolichopodidés et les Syrphidés des i
11	Cad1988		Cadou & Brunel	1988	CADOU D.	In : Inventaire analytique des Invertébrés de
12	Cad1991		Cadou	1991	CADOU D.	Les Diptères Syrphides de Bretagne. Prem
13	Cad2001		Cadou	2001	CADOU D.	Ils profitaient d'un beau soleil ce jour-là! Bu
14	Car2000a	a	Cartier	2000	CARTIER G.	Contribution à l'expertise du patrimoine nat
15	Car2000b	b	Cartier	2000	CARTIER G.	Addenda aux insectes de Rueil-Malmaisor
16	Che2001		Chevin et al.	2001	SAGOT P.	Une nouvelle enquête proposée aux natur
17	Cla1991		Claussen	1991	CLAUSSEN C.	Eine neue Orhonevra von Korsika (Diptera
18	Cla1993		Claussen & Kassebeer	1993	KASSEBEER C.F.	Eine neue Art der Gattung Cheilosia Meige
19	Cla1994		Claussen et al.	1994	LUCAS J.A.W.	Zur Identität von Pipizella heringii (Zetterste
20	Cla1995		Claussen & Vujić	1995	VUJICA.	Eine neue Art der Gattung Cheilosia Meige
21	Coc1990		Cocquempot & Chambon	1990	COCQUEMPOT C.	Contribution à l'étude de la faune entomolo
22	Cou1941		Coupin	1941	COUPIN L.	Contribution à l'étude des Syrphidés et Cor
23	Dau1999		Dauphin	1999	DAUPHIN P.	Données entomologiques sur la Réserve c

Cependant, la mise en place de la connexion ODBC entre PostgreSQL et Access pose quelques problèmes :

- impossibilité d'appliquer l'intégrité référentielle aux relations entre les tables (cf. §3.4.3)
- perte de la numérotation automatique pour les identifiants de chaque table

IV. LES INTERFACES CARTOGRAPHIQUES : Retour sur le fonctionnement

Afin de mieux comprendre leur fonctionnement, l'exemple de l'interface cartographique de répartition des syrphes, disposant d'un nombre plus significatif de contrôles et d'illustrations, servira d'exemple.

Il est important de distinguer :

- des zones disposant de formulaires interactifs permettant à l'utilisateur de sélectionner ou d'agir sur l'information, c'est-à-dire des contrôles de sélection (vert).
- les zones d'affichages d'informations dynamiques (rouge),
- Les liens statiques (jaune) et dynamiques (bleu).
- Les éléments utilisant le javascript (violet).

The screenshot shows a web interface titled 'Carte de répartition' for the species *Epistrophe leiophthalma*. On the left is a map of France with colored regions. On the right is a control panel with several sections:

- Sélection genre/espece:** Two dropdown menus showing 'Epistrophe' and 'leiophthalma'.
- Sélection statut/espece:** Radio buttons for 'en France' (selected) and 'en Europe', and dropdown menus for 'Statut' and 'Espèce'.
- Couches complémentaires:** A section with '+' and '-' buttons and the text 'Affichage des zones biogéographique'.
- Infos complémentaires:** A section with an 'Info' button and a '66' button.

At the bottom, there is a legend for 'REPRESENTATION DE LA REPARTITION DE : Epistrophe leiophthalma' with three categories: 'département avant 1970 seulement' (red), 'avant et apres 1970' (green), and 'apres 1970' (yellow). To the right of the legend is a compass rose and a section for 'Synonyme de l'espece:' showing 'E. leiophthalma' with a status of '1 : à surveiller'.

4.1. Fonctionnement et relation entre les contrôles de sélection

Les zones de contrôles telles que les listes déroulantes présentent de l'information à l'utilisateur qui pourra agir par sélection sur cette liste. Si on prend l'exemple de la première liste déroulante présentant les noms de genre, son contenu est obtenu par interrogation de la base de données de la façon suivante :

```
SELECT nom_genre FROM genre
```

Ainsi, **tous les genres existants** sont présentés dans la liste déroulante, qui est contenue dans ce qui est appelé un formulaire (`<FORM>` `</FORM>`) qu'il est important de nommer, ici « s_genre ».

La deuxième liste déroulante quand à elle doit afficher uniquement **les noms d'espèces associés au nom de genre sélectionné** par l'utilisateur dans le formulaire « s_genre ».

Pour cela, la valeur choisi dans la liste genre est enregistré dans une variable ici nommé \$genre_select (\$ indiquant la présence d'une variable en langage php). On spécifie ensuite au niveau de la deuxième liste déroulante que \$genre_select a pour valeur, la valeur choisi dans le formulaire « s_genre » avec la syntaxe suivante :

```
$genre_select = $_POST ['s_genre'] ;
```

Cette expression veut dire : \$genre_select reçoit pour valeur, celle postée par le formulaire « s_genre ».

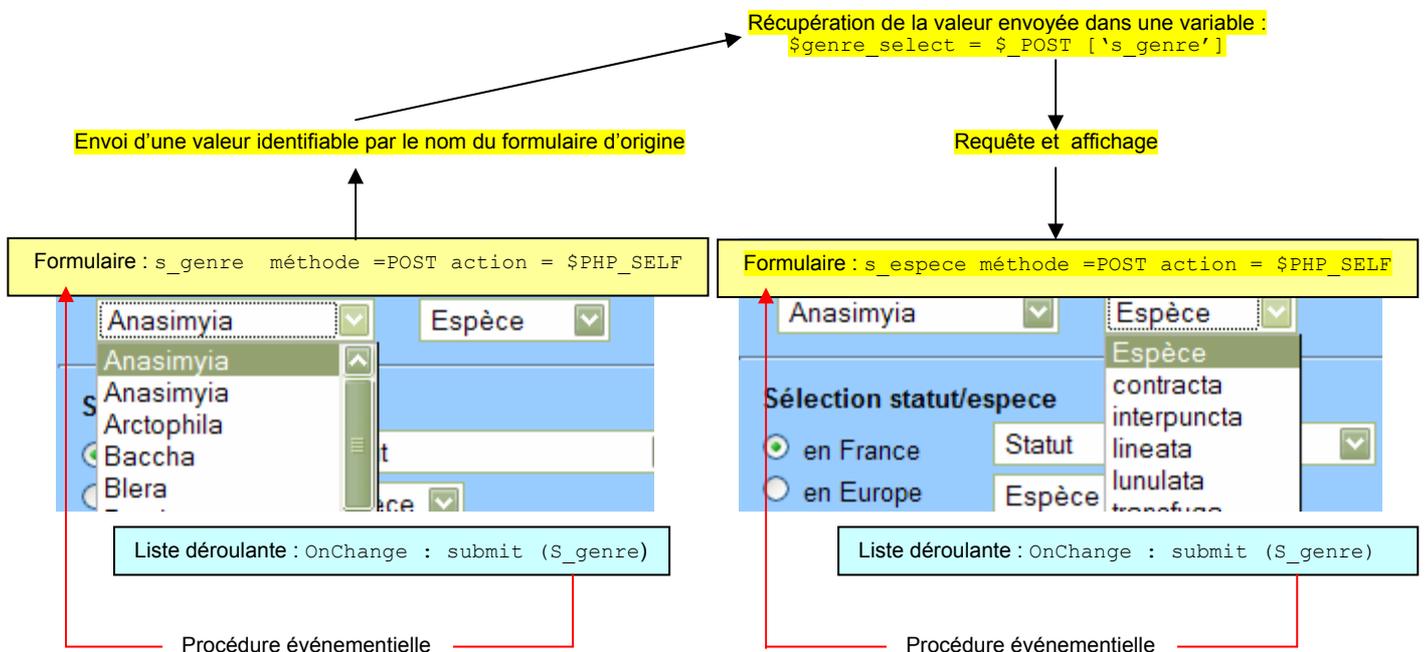
POST est une méthode d'envoi d'information vers une autre page, en spécifiant l'adresse de destination ex : adresse.php. Dans notre cas, l'information est envoyée vers la même page qui va donc s'actualiser, pour cela on remplace l'adresse de destination par <?php \$PHP_SELF ?>

\$genre_select étant maintenant défini, la requête de sélection des noms d'espèce peut être lancée avec la syntaxe suivante :

```
SELECT distinct nom_espece, num_espece FROM espece, genre
WHERE ( ref_genre like num_genre ) and ( nom_genre like '$genreselect' )
```

La seconde liste déroulante présente ainsi les **noms d'espèce associés à un nom de genre**. Le formulaire associé à 2^{ème} liste est « s_espece ».

La procédure « OnChange : submit('Un_formulaire') » est définie au niveau de chaque liste déroulante, afin que toute nouvelle sélection soit accompagnée d'un rafraîchissement de la page, c'est-à-dire nouvelle interrogation de la base, permettant la présentation de nouvelles données.



Ainsi, le premier choix du nom de genre va conditionner l'affichage des espèces associées. La sélection dans la deuxième liste va conditionner la procédure événementielle « OnChange : submit (s_espece) ». C'est sur la base de cette valeur numérique postée par s_espece que va être réalisée l'affichage de toutes les zones d'affichages d'informations dynamiques.

4.2 Les zones d'affichages d'informations dynamiques

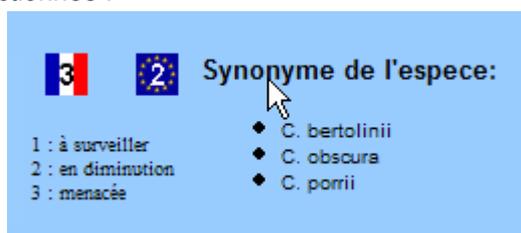
Les zones d'affichages d'informations dynamiques sont le résultat de l'interrogation de la base de donnée.

Cette interrogation est en faite une requête sur une ou plusieurs tables.

La majorité des zones informatives sont issues de requêtes basées sur la sélection d'une espèce identifiée par une variable numérique (clé-primaire, ici `num_espece`). La requête retourne en fonction de la sélection spécifiée, le contenu des champs d'une ou de plusieurs table(s).

Cette information issue de la base de donnée peut être soit :

- directement exploitable, c'est-à-dire affichable sur la page web, exemple les noms d'espèces synonymes de l'espèce sélectionnée :



- conditionner pour afficher une information d'un autre type, par exemple l'affichage d'image correspondant au statut Français ou Européen de l'espèce (information stockée sous forme de lettre S, D, M, dans la base de données).

L'utilisateur ne peut en aucun cas agir (modifier, cliquer, sélectionner) l'information contenue dans ces zones d'affichage.

A titre d'exemple, on présente la requête simple qui sur la base de la sélection d'une espèce permet d'obtenir les espèces synonymes :

```
SELECT distinct nom_syno
FROM synonyme, espece, genre
WHERE (ref_genre like num_genre) and (ref_espece like num_espece)
and (num_espece like '$num_especeselect')
```

On remarque la présence de la variable `$num_especeselect` qui correspond à la valeur identifiant l'espèce sélectionnée.

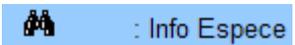
4.3. Les liens statiques et dynamiques

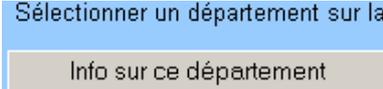
Les liens statiques ouvrent une page html, sans qu'il y ait passage de valeur, autrement l'ouverture et le contenu de la nouvelle page ne dépendent d'aucune sélection préalable dans la page d'origine.

C'est notamment le cas du lien vers l'aide [? : Aide](#) qui ne dépend en aucun cas de l'espèce ou du département sélectionnés.

A l'inverse les liens vers les fiches informatives sur l'espèce ou le département sont dynamiques.

En fonction de la sélection courante par les liste déroulantes pour l'espèce, ou par présélection sur la carte), l'action sur le lien entraînera l'envoi des valeurs courantes vers une destination spécifiée, soit :

- l'envoi de l'identifiant espèce de la sélection courante vers `fiche_info_espece.php` par action sur  : Info Espece .

- l'envoi de l'identifiant département de la présélection réaliser sur la carte, vers `fiche_info_dept.php` par action sur  Sélectionner un département sur la carte: Info sur ce département .

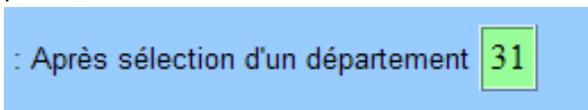
4.4. Les éléments javascript

Enfin des formulaires utilisant des fonctions javascript sont utilisés pour l'affichage d'information particulière. L'avantage est d'obtenir cet affichage de façon instantané et sans avoir à recharger la page web utilisée (ce qui permet notamment de ne pas perdre l'affichage de la sélection courante).



Ce premier formulaire fonctionne comme un interrupteur et permet l'affichage d'une image superposé à la carte thématique courante.

Sur le même principe cette zone affichage verte permet à l'utilisateur de connaître le **code du département** sur lequel il pointe, avant de réaliser une sélection.



Cette fonction en javascript est issue de la version première du site consacré aux syrphes.

V. LES PERSPECTIVES

5.1 La sélection

Sélections simples

Actuellement, il est possible de faire une sélection sur un département et de demander les informations correspondantes à ce département. Ceci est possible en utilisant une image réactive correspondant à la découpe des départements de France.

Le problème réside dans le fait que l'on est obligé de cliquer sur un département (ce clique permet la mise à jour d'une variable contenue dans un formulaire qui sera par la suite transmise pour l'interrogation de la base de données) puis de cliquer sur un bouton qui va lancer l'ouverture d'une nouvelle page contenant les informations demandées.

Il serait intéressant de faire évoluer ce système afin que le lien entre le clique sur un département et l'ouverture de la nouvelle page se fasse sans l'intermédiaire d'un bouton de validation.

Sélections multiples

Une des demandes concernant le site Web était la possibilité de pouvoir sélectionner de façon graphique plusieurs départements par rectangle de sélection ou par clique successif sur les départements. Cette fonctionnalité n'a pu être mise en œuvre, ainsi nous avons laissé la possibilité d'une sélection multiple non graphique par sélection de départements dans une liste déroulante.

Plusieurs pistes sont possibles pour améliorer le système et permettre une sélection graphique multiple :

- La première, peut être la plus rapide mais sûrement la moins intéressante, consisterait à conserver le système actuel et à trouver un moyen pour que le clique sur un département mette à jour une variable tableau qui rajouterai une ligne de tableau à chaque clique. L'inconvénient de cette méthode est l'impossibilité de zoomer à cause de l'image réactive qui ne pourra pas être redimensionnée.
- La deuxième solution (la plus intéressante et peut être la plus complexe) consisterait à utiliser MapServer en mode Query et de le coupler avec du script JavaScript (ou un applet Java comme rosa) qui permettrait d'avoir des outils de sélection graphique, de zoom, de déplacement,... Ces outils permettraient la récupération des coordonnées de la sélection et pourrait interroger la base de données en conséquence. Cette solution serait la plus élégante et permettrait l'utilisation des fonctionnalités de MapServer. De la même manière, cette solution permettrait également de régler le problème de la sélection simple vu précédemment.

5.2 Interface : carte de répartition

- La représentation de **la répartition du genre**, n'a pas été prise en compte dans l'interface cartographique de répartition. En effet la version antérieure réalisait cette analyse, mais cette dernière ne fonctionnait que sous Internet Explorer. Or durant les phases de développement, c'est Mozilla Firefox qui a été utilisé de façon majoritaire, ne nous permettant pas de découvrir cette fonctionnalité. Cependant si le projet doit être repris, l'ajout de cette fonctionnalité semble tout à fait réalisable. En effet, la création de la carte est induite par la sélection de nom d'espèce, soit pour le programme, l'existence de la variable « \$numespece_select ».

Autrement si cette variable n'existe pas : (if [! \$numespece_select]), alors la carte n'est pas créée.

Si on reprend le cheminement effectué par l'utilisateur, il y a d'abord **sélection d'un nom de genre**, puis **sélection des espèces relatives à ce genre**.

Il suffit donc de spécifier une création de carte supplémentaire basée sur le numéro identifiant, le nom de genre choisi, et sélectionner, dans la table « espèce », toutes les espèces liées à ce genre.

Il reste ensuite à conditionner l'affichage de la carte pour que si la variable « \$numespece_select » n'existe pas (pas de sélection réalisé sur l'espèce) mais que la variable « \$numgenre_select » existe (sélection du genre effectué), la requête de sélection de toutes les espèces du genre soit lancée.

- Une fonctionnalité intéressante qui n'a pu être réalisée faute de temps, est **l'affichage et la cartographie d'une espèce après sélection d'un genre**, dans l'hypothèse où le genre ne dispose que d'une seule espèce.

- Enfin, il semblerait intéressant dans le long terme de pouvoir comme **date charnière une variable plutôt qu'une date fixe**, ici 1970. Cette option n'est pas compliquée, puisqu'il s'agit simplement d'avoir une variable prenant une valeur spécifiée, et d'utiliser cette variable dans les requêtes de sélection et les requêtes de cartographie.

Le problème majeur lié à cette fonction est la représentation des observations. En effet dans notre cas, avec 1970, toute observation portant sur plusieurs années se situait entièrement avant ou après 1970. Or si on considère que l'utilisateur peut faire varier cette date charnière, il risque d'y avoir un problème pour représenter les observations multi année englobant la date faisant référence.

5.3 La partie administration : Access

L'ajout d'enregistrements sous Access peut être simplifiée en reprogrammant en VBA la numérotation automatique des identifiants de chaque table. A chaque fois que l'utilisateur souhaitera ajouter un nouvel enregistrement, le programme devra récupérer la plus grande valeur du champ id présente dans la table, l'incrémenter de 1 et l'insérer dans le formulaire d'ajout du nouvel enregistrement.

ANNEXES

DICTIONNAIRE DES DONNEES DE LA BASE

➤ **LES TABLES**

Table	Nombre d'enregistrements estimés
annee	2011
auteur	76
dater	6293
décrire	4742
ecrire	144
espece	511
france	95
genre	81
geometry_columns	16
observation	4593
publication	93
spatial_ref_sys	1785
statut	3
synonyme	163

○ **annee**

Colonne	Type	Not Null
num_annee	integer	NOT NULL

○ **auteur**

Colonne	Type	Not Null
num_auteur	integer	NOT NULL
nom_auteur	character varying(254)	

○ **dater**

Colonne	Type	Not Null
ref_observ	integer	NOT NULL
ref_annee_obs	integer	NOT NULL

○ **décrire**

Colonne	Type	Not Null	Defaut
ref_obs	integer	NOT NULL	nextval('public."décrire_ref_obs_seq"::text)
ref_publi	integer	NOT NULL	

○ **ecrire**

Colonne	Type	Not Null	Default
ref_publi	integer	NOT NULL	nextval('public.ecrire_ref_publi_seq'::text)
ref_auteur	integer	NOT NULL	
position	integer		

o **espece**

Colonne	Type	Not Null	Default
num_espece	integer	NOT NULL	nextval('public.espece_num_espece_seq'::text)
codeE(4)	character varying(50)		
nom_espece	character varying(254)		
decouvreur	character varying(254)		
date_nommage	character varying(100)		
UEF	smallint		
importance_inter	character varying(1)		
photo	character varying(254)		
ref_statut_fr	character varying(1)		
ref_statut_eu	character varying(1)		
ref_genre	integer		

o **france**

Colonne	Type	Not Null	Default
gid	integer	NOT NULL	nextval('public.france_gid_seq'::text)
code	integer	NOT NULL	
id	character varying		
name	character varying		
the_geom	geometry		

o **genre**

Colonne	Type	Not Null
num_genre	integer	NOT NULL
codeG(4)	character varying(4)	
nom_genre	character varying(50)	

o **geometry columns**

Colonne	Type	Not Null
f_table_catalog	character varying(256)	NOT NULL
f_table_schema	character varying(256)	NOT NULL
f_table_name	character varying(256)	NOT NULL
f_geometry_column	character varying(256)	NOT NULL
coord_dimension	integer	NOT NULL
srid	integer	NOT NULL
type	character varying(30)	NOT NULL

○ **observation**

Colonne	Type	Not Null
num_obs	integer	NOT NULL
ensemble_annees	character varying(254)	
date_maj	character varying(50)	
ref_dept	integer	
ref_espece	integer	

○ **publication**

Colonne	Type	Not Null	Defaut
num_publi	integer	NOT NULL	nextval('public.publication_num_publi_seq'::text)
code_publi	character varying(254)		
ordre	character varying(254)		
titre	character varying(254)		
date_maj	character varying(254)		
ref_annee_publi	integer		

○ **spatial ref sys**

Colonne	Type	Not Null
srid	integer	NOT NULL
auth_name	character varying(256)	
auth_srid	integer	
srttext	character varying(2048)	
proj4text	character varying(2048)	

○ **statut**

Colonne	Type	Not Null
num_statut	character varying(1)	NOT NULL
libelle	character varying(50)	

○ **synonyme**

Colonne	Type	Not Null	Defaut
num_syno	integer	NOT NULL	nextval('public.synonyme_num_syno_seq'::text)
codeS(8)	character varying(8)		
nom_syno	character varying(254)		
decouvreur_syno	character varying(254)		
date_nommage_syno	integer		
ref_espece	integer		

➤ **LES VUES**

Vue	Commentaire
effectifs_espece	donne par département le nombre d'espèce distinctes observées au moins une fois
effectifs_espece_ap70	donne par département le nombre d'espèce distinctes observées seulement après 1970
effectifs_espece_av70	donne par département le nombre d'espèce distinctes observées seulement avant 1970
effectifs_espece_avap70	donne par département le nombre d'espèce distinctes observées avant ET après 1970
effectifs_genre	donne par département le nombre de genres distincts observé au moins une fois
effectifs_genre_ap70	Donne par departement le nombre de genres distincts vus seulement après 1970
effectifs_genre_av70	Donne le nombre de genres distincts par département observés seulement avant 1970
effectifs_genre_avap70	Donne par département le nombre de genres distincts observés avant ET après 1970
espece_dept_annee	Requête pour avoir la liste des especes différentes par départements et leur annee min et max
france_annee_geom	jointure permet d'obtenir code, geom, num esp, num genre, num obs, annee min et max
france_esp_genre	
genre_dept_annee	Requête pour avoir la liste de genre différents par départements et leur annee min et max
liste_multiple	Permet d'obtenir le num espece, le nom de l'espece, le nom du genre, le numero de publication, l'annee d'observation et le code du departement
observation_annee_minmax	

Diagramme de Classe PHP MapScript

Diagramme de Classe PHP MapScript

RectObj	← Classe
RectObj ms_newRectObj()	← Fonction
double minx	← Attributs
int set(string property_name, new_value)	← Méthodes

Les fonctions permettent de générer ou manipuler les objets. Il s'agit principalement de constructeurs qui doivent être utilisés à la place du mot-clé "new" de php :

```
$oMap=ms_newMapObj($sMapFile);
```

Les attributs sont toujours en lecture seule. La seule façon de mettre à jour un attribut est d'utiliser une méthode telle que set(). Quand un attribut est suivi de la mention "read only" cela signifie que la méthode set() ne peut pas être utilisé avec cet attribut.

```
$iWidth=$oMap->width;
```

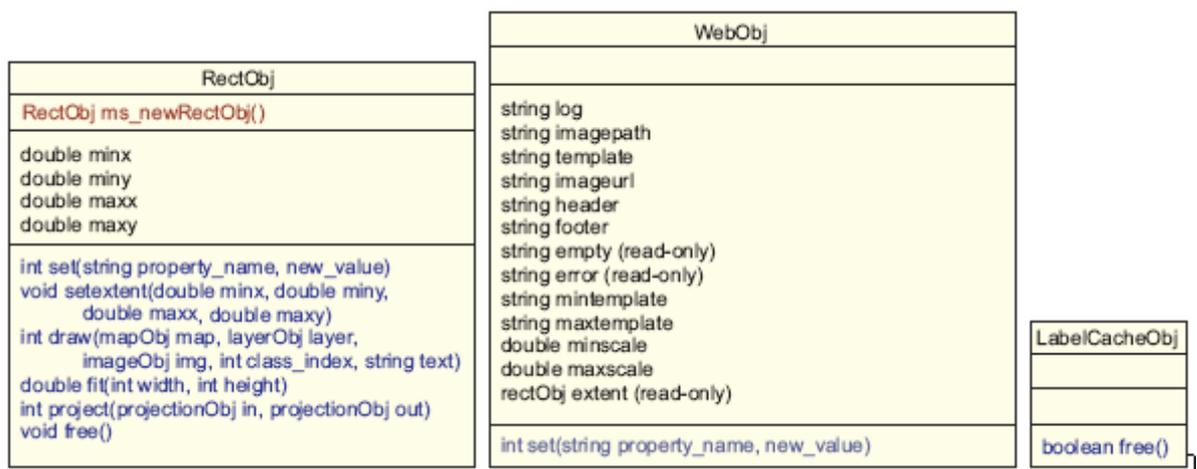
Syntaxe d'utilisation des méthodes :

```
$oMap->set(width,300);
```

Le délimiteur ">" remplace le traditionnel "." des langages à objets.




Conçu par Olivier Gayte VEREMES
 version 0.1 - 2004-05-25 à partir de
 Mapserver PHP/MapScript Class Reference - Version 4.0
 Jeff McKenna
 DM Solutions Group Inc.



LineObj	GridObj
LineObj ms_newLineObj() int numpoints (read-only) int add(pointObj point) int addXY(double x, double y [, double m]) PointObj point(int i) int project(projectionObj in, projectionObj out) void free()	double minsubdivide double maxsubdivide double minarcs double maxarcs double mininterval double maxinterval string labelformat int set(string property_name, new_value)

LegendObj	ScalebarObj
int height int width int keysizeX int keysizeY int keyspacingX int keyspacingY colorObj outlinecolor int status int position int transparent int interface int postlabelcache labelObj label colorObj imagecolor string template int set(string property_name, new_value)	int height int width int style int intervals colorObj color colorObj backgroundcolor colorObj outlinecolor int units int status int position int transparent int interface int postlabelcache labelObj label colorObj imagecolor int set(string property_name, new_value) int setimagecolor(int red, int green, int blue)

LabelObj	ClassObj
string font int type colorObj color colorObj outlinecolor colorObj shadowcolor int shadowsizeX int shadowsizeY colorObj backgroundcolor colorObj backgroundshadowcolor int backgroundshadowsizeX int backgroundshadowsizeY int size int minsize int maxsize int position int offsetX int offsetY double angle int autoangle int buffer int antialias int wrap int minfeaturesize int autominfeaturesize int mindistance int partials int force int set(string property_name, new_value)	ClassObj ms_newClassObj(layerObj layer, classObj class) string name string title int type int status double minscale double maxscale string template labelObj label int numstyles int set(string property_name, new_value) int setexpression(string expression) string getexpression(string expression) int settext(string text) int drawLegendIcon(int width, int height, imageObj im, int dstX, int dstY) imageObj createLegendIcon(int width, int height) styleObj getStyle(int index) classObj clone() int movestyleup(int index) int movestyledown(int index)

StyleObj	ShapefileObj
styleObj ms_newStyleObj(classObj class, styleobj style)	shapefileObj ms_newShapefileObj(string filename, int type)
int symbol string symbolname int size int minsize int maxsize int offsetx int offsety colorObj color colorObj backgroundcolor colorObj outlinecolor	int numshapes (read-only) int type (read-only) string source (read-only) rectObj bounds (read-only)
int set(string property_name, new_value) styleObj clone()	shapeObj getShape(int i) shapeObj getPoint(int i) shapeObj getTransformed(mapObj map, int i) rectObj getExtent(int i) int addShape(shapeObj shape) int addPoint(pointObj point) void free()

ReferenceMapObj	OutputFormatObj
string image int width int height int status rectObj extent ColorObj color ColorObj outlinecolor	string name string mimetype string driver string extension int renderer int imagemode int transparent
int set(string property_name, new_value)	boolean setformatoption(string property_name, new_value) string getformatoption(string property_name)

ProjectionObj
ProjectionObjObj ms_newprojectionobj(string projectionstring)

ShapeObj	PointObj
ShapeObj ms_newShapeObj(int type)	PointObj ms_newPointObj()
string text int classindex int type (read-only) int numlines (read-only) int index (read-only) int tileindex (read-only) rectObj bounds (read-only) int numvalues (read-only) array values (read-only)	double x double y double m
int set(string property_name, new_value) int add(lineObj line) LineObj line(int i) int draw(mapObj map, layerObj layer, imageObj img) boolean contains(pointObj point) boolean intersects(shapeObj shape) int project(projectionObj in, projectionObj out) pointObj getpointusingmeasure(double m) pointObj getmeasureusingpoint(pointObject point) void free()	int setXY(double x, double y [, double m]) int draw(mapObj map, layerObj layer, imageObj img, int class_index, string text) double distanceToPoint(pointObj poPoint) double distanceToLine(pointObject p1, pointObject p2) double distanceToShape(shapeObj shape) int project(projectionObj in, projectionObj out) void free()

MapObj
<pre>mapObj ms_newMapObj(string map_file_name [, string new_map_path])</pre>
<pre>int numlayers (read-only) string name int status int debug int width int height int maxsize int transparent int interface int imagetype (read-only) int imagequality double resolution rectObj extent double cellsize int units double scale (read-only) string shapepath int keysizeX int keysizeY int keyspacingX int keyspacingY webObj web referenceMapObj reference colorObj imagecolor scalebarObj scalebar legendObj legend string symbolsetfilename (read-only) string fontsetfilename (read-only) outputformatObj outputformat (read only) labelcacheObj labelcache</pre>
<pre>mapObj clone() int set(string property_name, new_value) int getsymbolbyname(string symbol_name) void preparequery() imageObj prepareImage() imageObj draw() imageObj drawQuery() imageObj drawLegend() imageObj drawReferenceMap() imageObj drawScaleBar() int embedLegend(imageObj image) int embedScalebar(imageObj image) int drawLabelCache(imageObj image) layerObj getLayer(int index) layerObj getLayerByName(string layer_name) colorObj getColorbyindex(int iColorIndex) void setextent(double minx, double miny, double maxx, double maxy) void zoompoint(int nZoomFactor, pointObj oPixelPos, int nImageWidth, int nImageHeight, rectObj oGeorefExt) void zoomrectangle(rectObj oPixelExt, int nImageWidth, int nImageHeight, rectObj oGeorefExt) void zoomscale(double nScale, pointObj oPixelPos, int nImageWidth, int nImageHeight, rectObj oGeorefExt) int queryByPoint(pointObj point, int mode, double buffer) int queryByRect(rectObj rect) int queryByShape(shapeObj shape) int queryByFeatures(int slayer) int queryByIndex(int layerindex, int tileindex, int shapeindex, int addtoquery) int savequery(filename) int loadquery(filename) void freequery(layerindex) int save(string filename) string getProjection()</pre>

```

int setProjection(string proj_params,
    boolean bSetUnitsAndExtents)
int setWKTProjection(string proj_params,
    boolean bSetUnitsAndExtents)
int getMetaData(string name)
int setMetaData(string name, string value)
int removeMetaData(string name)
array getLayersIndexByGroup(string groupname)
array getAllGroupNames()
array getAllLayerNames()
boolean moveLayerUp(int layerindex)
boolean moveLayerDown(int layerindex)
array getLayersDrawingOrder()
boolean setLayersDrawingOrder(array layerindex)
string processTemplate(array params,
    boolean generateImages)
string processQueryTemplate(array params,
    boolean generateImages)
string processLegendTemplate(array params)
int setSymbolSet(string fileName)
int getNumSymbols()
int setFontSet(string fileName)
int selectOutputFormat(string type)
int saveMapContext(string filename)
int loadMapContext(string filename)
int applySLD(string sldxml)
int applySLDURL(string sldurl)
string generateSLD()
    
```

LayerObj	
layerObj ms_newLayerObj(MapObj map, layerObj layer)	
int numclasses (read-only) int index (read-only) int status int debug string classitem string name string group string data int type int dump double tolerance int toleranceunits double symbolscale double minscale double maxscale double labelminscale double labelmaxscale int maxfeatures colorObj offsite int annotate int transform int labelcache int postlabelcache string labelitem string labelsizeitem string labelangleitem string tileitem string tileindex string header string footer string connection int connectiontype string filteritem string template string styleitem gridObj grid int num_processing	int set(string property_name, new_value) int draw(imageObj image) int drawQuery(imageObj image) classObj getClass(int classIndex) int queryByPoint(pointObj point, int mode, double buffer) int queryByRect(rectObj rect) int queryByShape(shapeObj shape) int queryByFeatures(int slayer) int queryByAttributes(string qitem, string qstring, int mode) int setFilter(string expression) getProjection() int setProjection(string proj_params) int setWKTProjection(string proj_params) int getNumResults() resultCacheMemberObj getResult(int index) int open() void close() shapeObj getShape(int tileindex, int shapeindex) int addFeature(shapeObj shape) int getMetaData(string name) int setMetaData(string name, string value) int removeMetaData(string name) string getWMSFeatureInfoURL(int clickX, int clickY, int featureCount, string infoFormat) array getItems() boolean setProcessing(string) array getProcessing boolean clearProcessing string executeWFSGetfeature() int applySLD(string sldxml, string namedlayer) int applySLDURL(string sldurl, string namedlayer) string generateSLD() int moveclassup(int index) int moveclassdown(int index)

Exemple de Script pour la création d'un fichier map

```

1:  <?php
2:  dl('php_mapscript_44');
3:  //
4:  // Paramètres pour le serveur PostgreSQL
5:  //
6:  $hote = "localhost";
7:  $base_de_donnees = "syrfid";
8:  $utilisateur = "syrfid";
9:  $mot_de_passe = "syrfid";
10: $france = "france";
11: //
12: // Paramètres pour PhpMapScript
13: //
14: //
15: // nom du fichier map contenant les informations de base
16: $Nom_Mapfile = "test.map";
17: // longueur de l'image à générer. La hauteur de l'image
   sera calculée selon la longueur spécifiée
18: $longueur_image_mapfile = "400";
19: // Chemin d'accès absolue vers le répertoire contenant
   l'image à générer
20: $mapserver_imagepath =
   "c:/ms4w/Apache/htdocs/syrfid/site/tmp/";
21: // URL locale selon le chemin d'accès spécifié pour l'image
22:
23: $mapserver_imageurl = "/syrfid/site/tmp/";
24: // nom de la donnée attributaire à afficher
25: // Celle-ci correspond à une colonne dans la table choisie
26: $mapserver_labelitem = "code";
27: //
28: // Connexion au serveur PostgreSQL
29: //
30: $db_handle = pg_connect("host=".$hote."
   dbname=".$base_de_donnees." user=".$utilisateur."
   password=".$mot_de_passe.");
31:
32: //
33: // Commençons par déterminer le nom de la colonne
34: // géométrique associée à la table choisie
35: //
36: $pgsql_colonne_geometrique = "select f_geometry_column,type
   from geometry_columns
   where f_table_name like
   ' ".$table_choisie." '";
37:
38: $pgsql_resultat =
   pg_exec($db_handle,$pgsql_colonne_geometrique);
39: $colonne_geometrique = pg_result($pgsql_resultat,0,0);
40: $type_donnee_geometrique_pour_mapserver =
   str_replace("LINESTRING",MS_LAYER_LINE,str_replace("MULTILINESTR
   ING",MS_LAYER_LINE,

```

```

42:     str_replace("MULTIPOLYGON",MS_LAYER_POLYGON,
43:     pg_result($pgsql_resultat,0,1)));
44:     //
45:     // Calcul de l'extent pour MapServer
46:     // et de la hauteur de l'image
47:     //
48:     $pgsql_extent = "select
49:     xmin(extent(".$colonne_geometrique."),
50:     ymin(extent(".$colonne_geometrique."),
51:     xmax(extent(".$colonne_geometrique."),
52:     ymax(extent(".$colonne_geometrique.)) from
53:     ".$france;
54:     $resultat_extent = pg_exec($db_handle,$pgsql_extent);
55:     $xmin = pg_result($resultat_extent,0,0);
56:     $ymin = pg_result($resultat_extent,0,1);
57:     $xmax = pg_result($resultat_extent,0,2);
58:     $ymax = pg_result($resultat_extent,0,3);
59:     $longueur_extent = abs($xmax-$xmin);
60:     $hauteur_extent = abs($ymax-$ymin);
61:     $rapport_extent = $longueur_extent / $hauteur_extent;
62:     $hauteur_image_mapfile = $longueur_image_mapfile /
$rapport_extent;
63:     //
64:     // Création de la mapfile
65:     //
66:     $map = ms_newMapObj($Nom_Mapfile);
67:     $map->setextent($xmin,$ymin,$xmax,$ymax);
68:     $map->web->set("imagepath",$mapserver_imagepath);
69:     $map->web->set("imageurl",$mapserver_imageurl);
70:     $map->set("width",$longueur_image_mapfile);
71:     $map->set("height",$hauteur_image_mapfile);
72:     //
73:     // Ajout du layer
74:     //
75:     $layer = ms_newLayerObj($map);
76:     // Nom du layer conforme à celui de la base de données
77:     $layer->set("name","Départements");
78:     $layer->set("status",MS_DEFAULT);
79:     // Connexion à la base de données PostGIS
80:     $layer->set("connectiontype",MS_POSTGIS);
81:     $layer->set("connection","user=".$utilisateur."
dbname=".$base_de_donnees." host=".$hote);
82:     // Requête spatiale utilisée
83:     $layer->set("data",$colonne_geometrique." from
".$france." as foo using unique geometry_columns.srid using
SRID=27582");
84:     //
85:     // Précision sur le type de donnée géométrique à prendre
en compte: LINE, POINT...
86:
87:     $layer->set("type",$type_donnee_geometrique_pour_mapserver);
88:     // Nom de la colonne de données attributaires à afficher
89:     $layer->set("labelitem",$mapserver_labelitem);

```

```
89:         //
90:         // Ajout de la class
91:         //
92:         $class = ms_newClassObj($layer);
93:         //
94:         // Ajout du style
95:         //
96:         $style = ms_newStyleObj($class);
97:         // Précision sur l'affichage des données
géométriques
98:         // couleur de bordure en code 128 0 0 pour RGB
99:         $style->set("size",4);
100:        $style->outlinecolor->setRGB(128,0,0);
101:
102:        //
103:        // Utilisation de PhpMapScript
104:        // pour générer l'image associée à la mapfile au format PNG
105:        //
106:        $image = $map->draw();
107:        $image_url = $image->saveWebImage(MS_PNG,1,1,0);
108:
109:
110:
111:
112:        //
113:        // Fermeture de la connexion à PostgreSQL
114:        //
115:
116:        pg_close($db_handle);
117:
118:        ?>
119:
120:
```