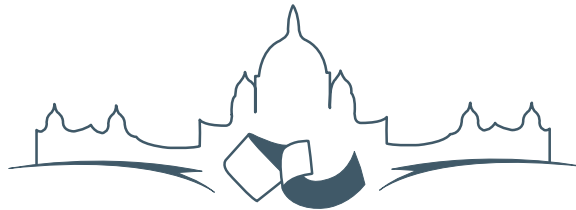

Journal de l'OSGeo

Le journal de la Fondation Open Source Geospatial

Volume 3 / Décembre 2007



2007 FREE AND OPEN SOURCE SOFTWARE
FOR GEOSPATIAL (FOS4G) CONFERENCE
VICTORIA CANADA  SEPTEMBER 24 TO 27, 2007

Compte rendu du FOS4G 2007

Intégration & Développement

- Portable GIS : SIG sur une clé USB
- Génération Automatique d'Applications SIG / Base de données sur Internet
- db4o2D - Extension de Base de données Orientée Objet pour les types géospatiaux 2D
- Google Summer of Code pour la géomatique

Intérêt thématique

- Approche générique pour la gestion de standards de métadonnées
- Vers des services web dédiés à la cartographie thématique
- Interopérabilité pour les données géospatiales 3D
- Un Service Web orienté modèle pour une interopérabilité sémantique améliorée
- Spatial-Yap : un système de base de données spatialement déductif

Études de cas

- Le Projet DIVERT : Développement de Télématiques Inter-Véhicules Fiables
 - GRASS et la Modélisation des Risques Naturels
 - Une Base de Données Spatiales pour l'Intégration des Données du Projet de Gestion des Ressources Naturelles du Rondonia
 - GeoSIPAM : Logiciel libre et Open Source appliqué à la protection de l'Amazonie brésilienne
 - Le Système de Suivi de la Déforestation Amazonienne
-

Génération Automatique d'Applications SIG / Base de données sur Internet

Nirut Chalainanont, Junya Sano and Toshimi Minoura,
traduit par Marie Silvestre

Résumé

Nous avons développé des applications Internet SIG / base de données (WebGD) qui permettent aux utilisateurs d'ajouter, interroger, mettre à jour, et supprimer des entités géographiques ainsi que leurs données attributaires à partir d'un navigateur Internet standard. Le code que partagent ces applications est organisé comme un framework WebGD. Le comportement de l'interface cartographique d'une application WebGD est défini par des fichiers de configuration. Nous avons également un générateur d'application WebGD (WebGD-Gen) qui produit automatiquement ces fichiers de configuration à partir des métadonnées de la base de données, incluant celles des tables `geometry_columns` et `spatial_ref_sys`. WebGD-Gen permet également de générer des scripts Web qui interagissent avec l'interface cartographique et la base de données. Le framework WebGD et WebGD-Gen permettent donc de réduire significativement les temps de développement et les coûts de maintenance d'une application Internet SIG / base de données complexe.

Introduction

Internet est devenu le lieu de prédilection pour le partage de l'information. Une application de cartographie dynamique sur Internet permet aux utilisateurs, sans avoir recours à un logiciel de SIG bureautique, d'effectuer des requêtes spatiales et de produire des cartes sur un navigateur Internet standard.

Cependant, une application SIG sur Internet typique permet simplement de récupérer des cartes et des données relatives à ces cartes. Un serveur web fournit des informations au client mais le client ne peut pas renvoyer des informations au serveur (3). Ce flux d'informations unidirectionnel est un problème majeur des applications cartographiques sur Internet actuelles. De plus, créer une application Internet interactive avec une interface cartographique prend du temps. Les serveurs cartographiques propriétaires et les systèmes de gestion de base de données géographiques coûtent cher. Cette situation

freine l'utilisation d'applications SIG Internet destinées à rassembler des données, à l'analyse et à la prise de décision.

Nous avons développé une série d'outils qui permettent de réduire significativement le coût des développements (14; 16; 9; 15). Le code partagé par les applications SIG Internet interactives est organisé comme un framework Internet SIG / base de données (WebGD). La plupart des fonctions complexes qui produisent des fonctions SIG sur Internet sont inclus dans ce framework. Avec le framework WebGD, en utilisant uniquement des fichiers de configuration, on peut créer une interface cartographique permettant aux utilisateurs d'ajouter, interroger, et supprimer des entités géographiques ainsi que leurs données attributaires

Une fonctionnalité importante d'une application WebGD est qu'elle intègre complètement les données spatiales et leur données attributaires, ce qui permet d'effectuer des analyses spatiales sur les données (13). Ces fonctions sont à la disposition des utilisateurs sur des zones géographiques différentes, pour des questions de gestion économique et rapide des données.

Il y a quelques années, nous avons développé un générateur de formulaires web qui génère automatiquement des scripts pour des formulaires traditionnels à partir du schéma d'une base de données relationnelle (4). Cette fonctionnalité a été récemment étendue en générateur d'applications WebGD (WebGD-Gen). Ce nouveau générateur d'applications peut produire la plupart du code pour une application WebGD à partir du schéma relationnel d'une base de données et les informations concernant les *geometry columns*. Avec l'interface cartographique et les scripts web générés automatiquement, les entités géographiques (par exemple des points, des lignes et des polygones) peuvent être ajoutés, interrogés, et supprimés. Ainsi, si un schéma relationnel et des métadonnées sur les couches sont disponibles, une application *non-personnalisée* peut rapidement être mise en place grâce au framework WebGD et au générateur d'applications WebGD-Gen.

La génération automatique d'interfaces cartographiques et de scripts pour formulaires web rend possible le développement *incrémental* et *itératif* d'applications SIG Internet. Lorsqu'une couche géographique est ajoutée ou modifiée, il suffit de créer ou

de mettre à jour le fichier de configuration concernant cette couche puis de régénérer le formulaire web pour cette couche. Lorsqu'un schéma de base de données est modifié, il est possible de régénérer intégralement le set de scripts web en quelques minutes. Ainsi, même avec un jeu de couches géographiques et schéma de base de donnée partiel, il est possible de générer un prototype fonctionnel pour de premiers tests. De plus, la plupart des bugs peuvent être corrigés en modifiant simplement le code partagé dans le framework WebGD ou dans WebGD-Gen puis en régénérant les scripts web pour chaque application.

Les applications WebGD utilisent les composants *open-source* suivants. PostgreSQL, un système de gestion de base de données *relationnel-objet*, et PostGIS permettent de gérer les données spatiales. PostGIS est une extension de PostgreSQL pour les applications SIG (11). MapServer (12) génère les cartes qui sont visualisées par un navigateur web à partir des données spatiales fournies par PostGIS. Les pages web, dont celle qui affiche les cartes, sont générées par des scripts PHP côté serveur. Le module PHP MapScript interagit avec MapServer (6; 2). Lorsqu'une requête d'ajout ou de suppression d'une entité géographique est reçue par un script PHP, celui-ci accède directement à la base de données PostgreSQL en utilisant l'extension PostGIS. Ce genre d'applications tournent sur un PC sans aucun logiciel propriétaire.

Nous détaillons les fonctionnalités du framework WebGD dans la partie **Applications WebGD**. Le processus de génération des fichiers de configuration des couches géographiques est détaillé dans la partie **Framework WebGD** et celui de la génération des scripts web dans la partie 4. Le processus de génération automatique des fichiers de configuration des couches géographiques est expliqué dans la partie **Génération automatique des fichiers de configuration des couches géographiques**. La partie **Historique du Développement de WebGD** se consacre à un bref historique du développement du framework WebGD et de WebGD-Gen. La partie **Conclusions et travaux à venir** conclue cet article.

Applications WebGD

la figure 1 montre l'interface web d'une des applications WebGD, le "Natural Heritage Information System" de Caroline du Nord (NHIS). Cette application fournit une interface cartographique pour une copie de la base de données Biotics 4.0 maintenue par le "North Carolina Natural Heritage Program".

Biotics 4.0 est une application SIG bureautique basée sur la base de données développée par NatureServe. Les éléments clés de cette base sont les *occurrences d'éléments* (EOs) qui correspondent aux zones terrestres et/ou aquatiques dans lesquelles les espèces sont présentes ou ont été présentes (7). Les enregistrements d'EO sont composés à la fois de données *spatiales* et de données *tabulaires*, et la base de données contient environ 700 tables relationnelles citeFogelsong :2002. L'extension Biotics Mapper implémentée dans ArcView par NatureServe fournit une interface cartographique qui permet de représenter, créer, mettre à jour et supprimer les EOs et les données attributaires (8). Dans notre implémentation, nous pouvons réaliser toutes ces opérations à partir d'un simple navigateur Internet. De plus, des formulaires web, environ 3500 au total, sont générés pour toutes les tables de la base de données.

L'application NHIS permet des transferts *bi-directionnels* de données spatiales aussi bien que de données ordinaires. Les scientifiques ou autres personnes disposant des droits nécessaires peuvent *ajouter, interroger et supprimer* des entités géographiques telles que des polygones EO, des lignes, des points ainsi que des données attributaires. Des requêtes peuvent être effectuées en *sélectionnant des zones* sur la carte ou en utilisant un formulaire web traditionnel. De plus, des images *orthorectifiées* ou des photos aériennes dont la résolution est d'un mètre sont disponibles en tant que couche. Lorsque ces images sont combinées avec d'autres couches cartographiques telles que les autoroutes, les limites de comtés, le réseau hydrographique et les rues, des lieux peuvent être facilement localisés grâce à la superposition des couches (16).

Les opérations principales proposées par l'interface cartographique d'une application WebGD sont les suivantes :

1. Pour récupérer des informations concernant des entités géographiques d'une zone d'intérêt, l'utilisateur peut zoomer ou dézoomer sur cette zone en utilisant les outils de navigation cartographique. Si l'utilisateur zoome suffisamment, les images aériennes d'un mètre de résolution deviennent visibles. L'utilisateur peut également se déplacer vers une nouvelle zone en la sélectionnant dans la liste des **raccourcis**.
2. Pour obtenir des informations sur une entité géographique, l'utilisateur peut sélectionner la couche dans la légende et le bouton **Information** dans la barre d'outils puis cliquer sur les limites de cette entité.
3. La fonction **Insert** permet d'ajouter une entité

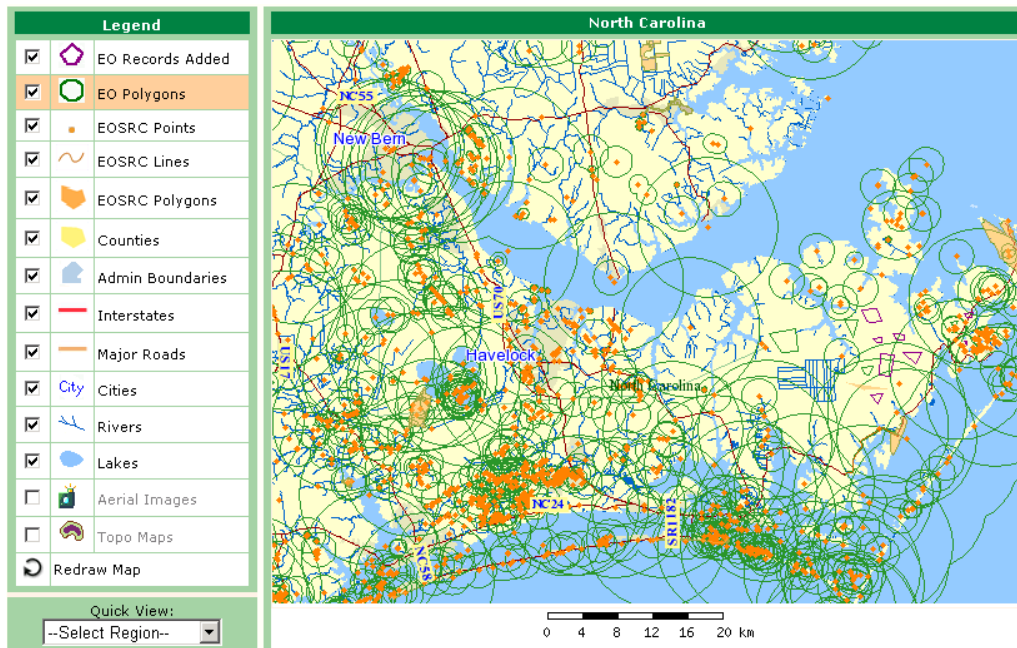


FIG. 1: Interface de l'application NHIS de Caroline du Nord

géographique par clic sur la carte. Le bouton **Done** doit être pressé une fois que tous les points ont été rentrés.

4. La fonction **Search by Area** permet aux utilisateurs de récupérer une liste d'entités comprises dans un *rectangle d'emprise* spécifié sur la carte et qui satisfont les critères de recherche. Ces entités sont mises en *surbrillance* sur la carte. De plus, l'utilisateur peut sélectionner des entités dans la liste en cochant les cases qui leur sont associées. Ensuite, la carte est actualisée et les entités sélectionnées sont mises en surbrillance.
5. L'interface d'administration des données peut être activée en cliquant sur **Database** dans la barre de menu sous la bannière. On peut cliquer sur un icône pour afficher une *arborescence* que l'on peut parcourir. L'arborescence pour les **taxons** est la principale. Comme certaines données sur les espèces menacées sont confidentielles, l'utilisateur doit s'authentifier avec un mot de passe pour accéder aux données de cette application.

Plusieurs applications WebGD créées avec le framework WebGD et WebGD-Gen sont accessibles sur cette page.³

³Exemples d'applications WebGD : http://yukon.een.orst.edu/index_webgd.html Vous pouvez y ajouter/interroger/mettre à jour/supprimer des données. Cependant, NE DISTRIBUEZ PAS ces données. Si un login est demandé, utilisez le nom d'utilisateur cs540 et le mot de passe CSxyz540. Il est possible que vous ayez à vous authentifier deux fois, une fois pour le serveur puis une fois pour une application.

La première application proposée est le "Natural Heritage Information System". Bien que cette application couvre tout les USA ou le monde, les données actuellement disponibles ne couvrent que la Caroline du Nord. La deuxième application fournit une interface cartographique pour une application qui suit les pratiques de conservation sur des parcelles de territoire. La troisième est une application cartographique sur Internet pour la collection de plantes germplasm maintenue à la "Western Regional Plant Introduction Station" (USDA-ARS). La quatrième application permet de récupérer les informations sur le sol à l'endroit sélectionné par clic sur l'interface cartographique.

Une caractéristique remarquable du framework WebGF actuel est la possibilité de *changer dynamiquement de système de référence spatiale*. Typiquement, chaque région géographique ou localité a des *projections recommandées* de manière à éviter les distorsions sur les cartes créées (1). Le framework permet de couvrir le monde entier avec des cartes différentes à différentes échelles, par exemple la carte du monde, celle d'un continent ou d'une région. L'interface cartographique sélectionne automatiquement la projection la mieux adaptée à la zone géographique

affichée. Par exemple, pour le monde on peut utiliser un *système de coordonnées géographiques*, pour les USA la *projection équivalente Albers* ou pour l'Oregon la *projection conique conforme Lambert*. Les analyses spatiales peuvent donc se faire en utilisant la projection la mieux adaptée pour une région particulière. Le *changement dynamique de référence spatiale*, le *fichier de configuration de la carte*, la *légende* et les *raccourcis* supportés par le framework WebGD permettent de couvrir n'importe quelle partie du monde ayant sa propre échelle et son propre système de référence spatiale, y compris les régions couvertes par des images aériennes d'un mètre de résolution. Il s'agit d'une fonctionnalité très importante d'autant plus qu'aujourd'hui le coût de stockage d'images aériennes sur les Etats-Unis entiers devient abordable (les 10 téraoctets nécessaires pour stocker des images aériennes sur tous les USA coûtent actuellement environ \$10,000). De plus, beaucoup d'Etats reversent leurs images aériennes dans le domaine public.

Framework WebGD

Le framework WebGD propose les fonctionnalités de base de l'interface cartographique d'une application WebGD telles que le zoom, le déplacement sur la carte et les opérations d'ajout/interrogation/mise à jour/suppression sur les entités géographiques.

Une opération sur la carte qui ne manipule pas la géométrie des entités, par exemple, le *zoom* ou le *déplacement sur la carte* est traitée de la manière suivante :

1. L'action de l'utilisateur sur la carte est transmise par le navigateur web au serveur web par une requête HTTP GET.
2. Le serveur web active le script PHP qui gère l'action de l'utilisateur.
3. Dans le script PHP, plusieurs méthodes PHP MapScript sont appelées pour préparer les nouveaux paramètres de la carte tels que la projection cartographique et l'étendue de la carte ainsi que les noms des couches à afficher. La méthode de dessin de la carte dans PHP MapScript est ensuite appelée pour demander la création de l'image de la carte à MapServer.
4. MapServer demande les données relatives aux couches géographiques stockées dans la base de données PostgreSQL.
5. Les données pour les couches sont renvoyées.
6. L'image de la carte est renvoyée au script PHP.
7. La page HTML générée par le script PHP est renvoyée au serveur web.

8. Le serveur web transmet la page HTML au navigateur web en incluant l'image de la nouvelle carte .

Une opération cartographique qui accède ou manipule la géométrie d'entités, par exemple la *recherche par zone*, l'*ajout* ou *déplacer un point*, est traitée de la même manière que ci-dessus, excepté qu'entre les étapes 2 et 3, les opérations suivantes sont réalisées :

- a Le script PHP se connecte à la base de données PostgreSQL pour réaliser une opération spatiale via une requête SQL adaptée pour PostGIS.
- b Le résultat de l'opération spatiale est renvoyée au script PHP par requête SQL.

Le framework WebGD fait beaucoup appel à des *fichiers de configuration* pour la personnalisation d'une application. Les fichiers de configuration principaux sont les *fichiers de configuration des régions*, les *fichiers de configuration des couches géographiques* et un *fichier de configuration des raccourcis*.

Fichiers de Configuration de Région

Afin de limiter les distorsions, différentes zones géographiques peuvent être affichées en utilisant différents systèmes de référence spatiale. La *plus petite zone* qui contient l'étendue de la carte à générer détermine la *région actuelle*. La *référence spatiale* est ensuite automatiquement changée pour correspondre à celle de la nouvelle région. Enfin la légende des couches, la liste des commandes de navigation sur la carte et de manipulation des données ainsi que la liste des *raccourcis* sont reconfigurés pour cette nouvelle région. Toutes ces reconfigurations sont nécessaires car il se peut que différentes régions nécessitent différents jeux de couches géographiques, commandes et listes des raccourcis.

Pour chaque région, le *fichier de configuration de région* inclut les éléments suivants :

1. le système référence spatiale spécifique à la région,
2. le *nom de la région* qui est affiché sur l'interface cartographique,
3. *l'unité de mesure des distances*,
4. le *nom du fichier de configuration des couches géographiques* de la région,
5. le *nom du fichier de configuration des raccourcis*.

Par exemple le fichier de configuration pour le monde correspond à :

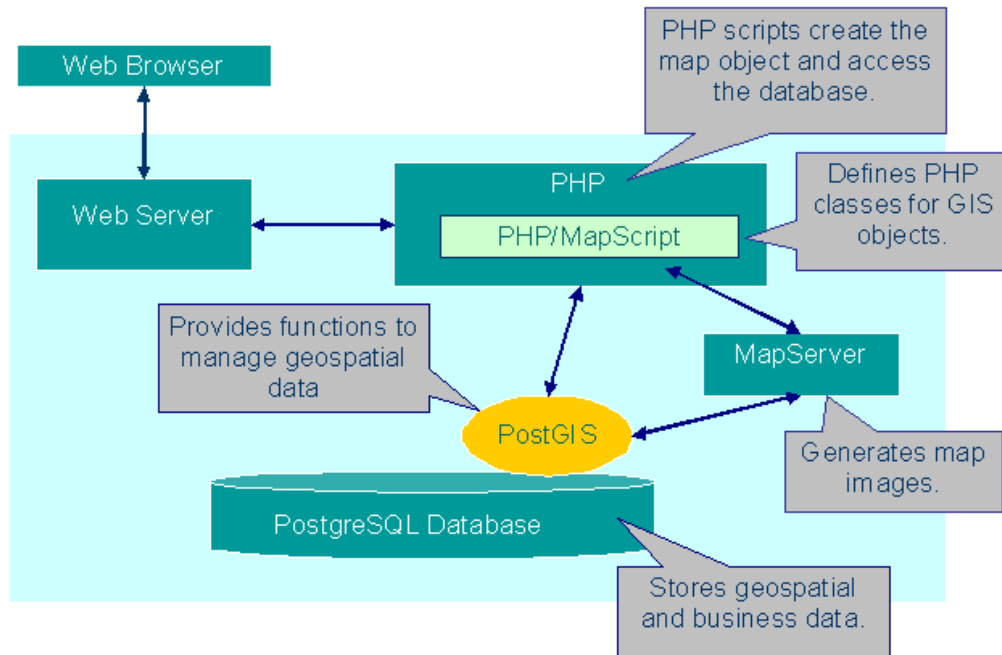


FIG. 2: Organisation d'une Application WebGD

```
$region = array(
    "gid" => 1,
    "name" => "world",
    "display_name" => "World",
    "srid" => 4326,
    "rank" => 1,
    "units" => "MS_DD",
    "mapfile" => "gmap75_world.map",
    "quickview" => "world_gview.php",
    "legend" => "world_maplayers.php",
    "proj4text" => "+proj=longlat +ellps=WGS84
    +datum=WGS84"
);
```

Le options suivantes peuvent être spécifiées dans un fichier de configuration :

- gid** – l'identifiant unique associé à chaque région, il s'agit de la clé primaire de chaque enregistrement représentant une région dans la table regions de la base de données.
- name** – le nom de la région pour le programme.
- display_name** – le nom de la région utilisé sur l'interface cartographique. Ce nom peut inclure des capitales, des espaces et autres signes de ponctuation.
- srid** – l'identifiant de la référence spatiale pour la projection cartographique associée à la région.
- rank** – le chiffre représentant la priorité, utilisé pour déterminer la région à afficher sur la carte. La région avec le chiffre le plus élevé est sélectionnée parmi les régions dont l'étendue est com-

- plètement comprise dans celle de la carte à afficher. Les régions avec les rangs les plus élevés recouvrent celles dont les rangs sont inférieurs.
- units** – l'unité de mesure associée à la région.
- mapfile** – le fichier de configuration de la carte à charger lorsque la région est sélectionnée.
- quickview** – le nom du fichier de configuration du raccourci pour la région.
- legend** – le nom du fichier de configuration des couches géographiques pour la région.
- proj4text** – la chaîne de caractères de définition de la projection utilisée pour la région.

Dès que la zone cartographique change, suite à une action de l'utilisateur sur l'interface cartographique, le fichier de configuration de la nouvelle région est chargé automatiquement. Ensuite l'interface cartographique est modifiée selon le fichier de configuration de cette nouvelle région.

Par exemple, le fichier de configuration pour l'Oregon correspond à :

```
$region = array(
    "gid" => 150137,
    "name" => "oregon",
    "display_name" => "Oregon",
    "srid" => 6010,
    "rank" => 150137,
    "units" => "MS_FEET",
    "mapfile" => "gmap75_oregon.map",
    "quickview" => "oregon_gview.php",
```

```

"legend" => "oregon_maplayers.php",
"proj4text" => "+proj=lcc +lat_1=43.0
+lat_2=45.5 +lat_0=41.75
+lon_0=-120.5
+x_0=400000.00000 +y_0=0.0"
);

```

Fichier de Configuration des Couches Géographiques

Le fichier de configuration des couches géographiques pour une région spécifie les couches à inclure dans la légende ainsi que leurs caractéristiques. Plusieurs régions peuvent partager un même fichier de configuration des couches géographiques.

Par exemple, le fichier de configuration des couches géographiques pour la région de l'Oregon est défini comme suit :

```

$layer_groups = array (
  'grp_eo_py' => array(
    'geom_type' => 'polygon',
    'table' => 'eo_py',
    'layer_selectable' => true,
    'gid_column' => 'gid',
    'geom_col' => 'the_geom',
    'legend_label' => 'EO Polygons',
    'search_script' =>
      'forms/eo/eo_py_eo_search.phtml',
    'select_script' =>
      'forms/eo/eo_py_eo_select.phtml',
    'edit_script' =>
      'forms/eo/eo_py_edit.phtml',
    'normal_layer' => 'eo_py',
    'searched_layer' => 'eo_py_searched',
    'checked_layer' => 'eo_py_checked',
    'selected_layer' => 'eo_py_selected',
    'img_src' => 'images/eo_poly.png',
    'img_width' => 26,
    'img_height' => 26,
    'onclick' => 'activate_layer("grp_eo_py")',
    'data_srid' => 32119
  ),
  'grp_eosrc_pt' => array(
    'geom_type' => 'point',
    'table' => 'eosrc_pt',
    'layer_selectable' => true,
    'gid_column' => 'gid',
    'geom_col' => 'the_geom',
    'legend_label' => 'EOSRC Points',
    'search_script' =>
      'forms/eo/eosrc_pt_search.phtml',
    'select_script' =>
      'forms/eo/eosrc_pt_select.phtml',
    'edit_script' =>
      'forms/eo/eosrc_pt_edit.phtml',
    'normal_layer' => 'eosrc_pt',
    'searched_layer' => 'eosrc_pt_searched',
    'checked_layer' => 'eosrc_pt_checked',
    'selected_layer' => 'eosrc_pt_selected',
    'img_width' => 5,
    'img_height' => 5,
    'onclick' => 'activate_layer("grp_eosrc_pt")',
    'data_srid' => 32119
  )
);

```

```

),
...

```

La légende de la carte sur la figure 3 a été produite avec l'exemple de fichier de configuration présenté ci-dessus.

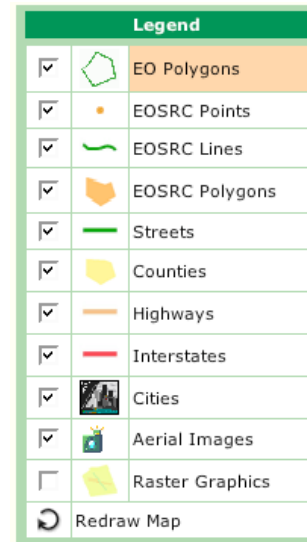


FIG. 3: Légende de la carte de l'application WedGD NHIS

Nous allons détailler chaque option utilisée dans ce fichier de configuration :

geom_type – le type de géométrie des entités contenues dans la couche. Il peut s'agir de polygon (polygone), multipolygon (multi-polygone), linestring (ligne), multilinestring (poly-ligne), point (point) ou multipoint (multi-point).

table – le nom de la table dans la base de données qui contient le champ géométrie de la couche.

layer_selectable – une valeur booléenne qui détermine si des opérations spatiales peuvent être effectuées sur la couche ou pas. Certaines couches, telles que les comtés et les routes principales de l'exemple ci-dessus, sont statiques et ne sont pas sélectionnables pour les opérations spatiales.

gid_column – le nom du champ qui contient les identifiants des entités géométriques de la couche.

geom_col – le nom du champ géométrie pour la géométrie des entités contenues dans la couche. Le nom par défaut est the_geom.

legend_label – le nom de la couche dans la légende.

search_script – le nom et la localisation du formulaire de recherche associé à la couche.

select_script – le nom et la localisation du formulaire de sélection associé à la couche.

edit_script – le nom et la localisation du formulaire d'édition associé à la couche.

normal_layer – le nom de la couche affichée lorsqu'elle n'est pas en surbrillance.

searched_layer, checked_layer, selected_layer – les noms des couches utilisées pour mettre en surbrillance les entités géométriques respectivement *cherchées, cochées* ou *sélectionnées*.

img_src, img_width, img_height – le nom du fichier, la largeur et la hauteur de l'icône dans la légende.

onclick – le nom du gestionnaire d'événements javascript activé lorsque l'utilisateur sélectionne la couche dans la légende.

data_srid – l'identifiant de la référence spatiale (*srid*) qui correspond à la projection cartographique utilisée pour les entités géométriques de la couche. Si le *srid* est différent de celle de la région, les entités géométriques de la couche sont reprojctées avant d'être affichées sur la carte.

Fichier de Configuration des Raccourcis

Le mécanisme de *raccourcis* permet à l'utilisateur de sélectionner une zone d'intérêt. C'est-à-dire que l'utilisateur peut rapidement passer à une nouvelle zone en la sélectionnant dans la liste des raccourcis.

Chaque entrée dans un *fichier de configuration des raccourcis* décrit le nom de la zone représentée par cette entrée, la projection cartographique utilisée sur l'étendue de la zone et l'étendue. Par exemple, le fichier de configuration du raccourci pour la région de l'Oregon présentée sur la figure 4 pourrait être comme suit :

```
$qview = array(
  array(
    'name' => 'World',
    'srid' => 4326,
    'extent' => '-180,-90,180,90'
  ),
  array(
    'name' => 'United States',
    'srid' => 4326,
    'extent' => '-125,13,-65,53'
  ),
  array(
    'name' => 'United States, East',
    'srid' => 4326,
    'extent' => '-102,22,-60,50'
  ),
  array(
    'name' => 'United States, West',
    'srid' => 4326,
    'extent' => '-135,30,-105,50'
  ),
)
```

```
array(
  'name' => 'Whole Oregon',
  'srid' => 6010,
  'extent' => '46461.662375,-43912.968464,
    2487069.754184,1785176.331408',
),
...
```

Voici la liste des raccourcis des régions telle qu'elle apparaît sur l'interface cartographique correspondant au fichier de configuration présenté ci-dessus.

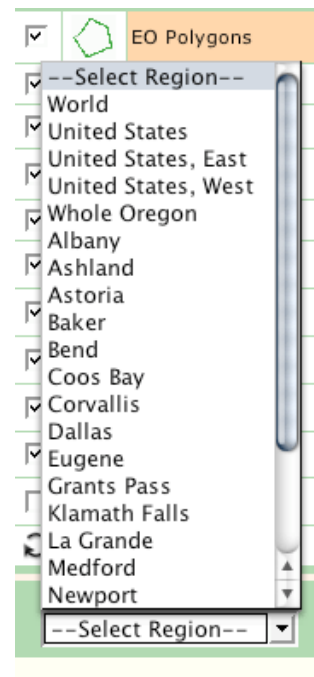


FIG. 4: Choix des raccourcis pour l'application WebGD NHIS

Pour chaque entrée dans le fichier de configuration des raccourcis, les options suivantes peuvent être spécifiées :

name – le nom de la zone.

srid – l'identifiant de la référence spatiale pour l'étendue détaillée ci-dessous.

extent – les coordonnées *xmin, ymin, xmax* et *ymax* décrivant la position des coins inférieur gauche et supérieur droit de la zone géographique.

Génération automatique de scripts de formulaires web

Plusieurs outils ont été développés pour enrichir le framework WebGD et simplifier le développement d'une application. Le générateur de site In-

Internet WebGD (WebGD-Gen) peut créer une application WebGD entière, incluant une interface cartographique web. WebGD-Gen génère *automatiquement* un jeu de scripts de formulaires web cohérents à partir des *fichiers de configuration* qui sont également générés automatiquement à partir du schéma relationnel d'une base de données. La génération de formulaires étant automatique, les coûts de développement d'une application en sont grandement réduits. Pour une base de données telle que Biotics qui contient environ 700 tables, programmer l'ensemble des 3 500⁴ formulaires nécessaires manuellement serait extrêmement coûteux voir infaisable.

WebGD-Gen correspond à une collection de *templates*. Chaque template, combiné avec le *fichier de configuration* correspondant, génère un des six types de scripts web : les scripts *recherche*, *sélection*, *édition*, *informations*, *action* et *arborescence*. Les templates et les fichiers de configuration sont écrits en PHP. Les scripts web qu'ils génèrent sont également en PHP. Ces scripts générés sont exécutés par un serveur web via un interpréteur PHP. Chaque script, excepté le script action, crée un formulaire web qui est affiché par l'ordinateur client via un navigateur Internet. La figure 5 illustre les interactions entre les scripts web et les formulaires.

De plus, si les lignes suivantes données en exemple sont ajoutées à un fichier de configuration, WebGD-Gen permet de générer automatiquement les requêtes pour l'ajout, la recherche et la suppression d'*entités géographiques* :

```
// type of geographical features
$web_gd = 'MULTIPOLYGON';
// layer group in legend
$layer_name = 'grp_eo_py';
// geometry column containing shapes
$geometry_column = 'the_geom';
//geographical feature IDs
$gid_column = 'gid';
// epsg spatial reference
$db_table_srid = 32119;
```

Les formulaires générés pour les entités géographiques ont les fonctionnalités supplémentaires suivantes, comparé aux formulaires des tables ordinaires d'une base de données :

1. Un formulaire de recherche peut être activé à partir de l'interface cartographique. Dans ce cas, l'étendue du rectangle de recherche spécifié sur la carte est passé en tant que paramètre additionnel de recherche.
2. Un formulaire de sélection comprend du code Javascript supplémentaire pour mettre en sur-

brillance les entités géographiques récupérées ou sélectionnées par l'utilisateur.

3. Un formulaire d'édition peut insérer un enregistrement pour une entité géographique, après transformation des coordonnées depuis le système de référence spatiale de l'interface cartographique actuelle vers celui utilisé par le champ géométrie de l'enregistrement.

Les formulaires reliés sont automatiquement associés les uns aux autres. Par exemple, la figure 6 montre le formulaire d'*édition* d'une table Etudiant. Depuis ce formulaire d'édition, l'utilisateur peut ouvrir les formulaires du département et des cours associés à l'étudiant. Les informations nécessaires pour créer ces liens sont extraites des *relations clé primaire-clé étrangère* entre les tables dans la base de données.

Afin de générer le formulaire d'édition d'un Etudiant de la figure 6, les paramètres suivants doivent être rentrés dans le fichier de configuration.

```
$edit_fields=array(
  array("column"=>"student_id",
    "label"=>"Student Id", "type"=>"numeric",
    "maxlen"=>"40", "size"=>"40"),
  array("column"=>"first_name",
    "label"=>"First Name", "type"=>"text",
    "maxlen"=>"40", "size"=>"40"),
  array("column"=>"age", "label"=>"Age",
    "type"=>"text", "maxlen"=>"40", "size"=>"40"),
  array("column"=>"gender", "label"=>"Gender",
    "type"=>"text", "maxlen"=>"40", "size"=>"40"),
  array("column"=>"department_id",
    "label"=>"Department ID", "type"=>"to_one",
    "linked_table"=>"departments", "maxlen"=>"40",
    "size"=>"40"),
  array("label"=>"Courses Taken", "type"=>"to_many",
    "linked_table"=>"courses", "maxlen"=>"40",
    "size"=>"40"),
  array("column"=>"other_info",
    "label"=>"Other Information",
    "type"=>"textarea", "rows"=>"4",
    "cols"=>"32",
  );
```

Chaque élément du tableau \$edit_fields représente un champ dans un formulaire d'édition avec les options column (champ), label (étiquette), table (table), maxlen (longueur maximale), size (taille), et type (type). L'option type peut être numeric, text, time, date, email, phone, textarea, to_one ou to_many.

textarea : il s'agit d'une chaîne de caractère affichée dans une zone de texte.

to_one : Un champ de type to_one permet à l'utilisateur de visualiser ou de modifier un enregistrement lié à l'actuel via une relation du type

⁴700 x 50

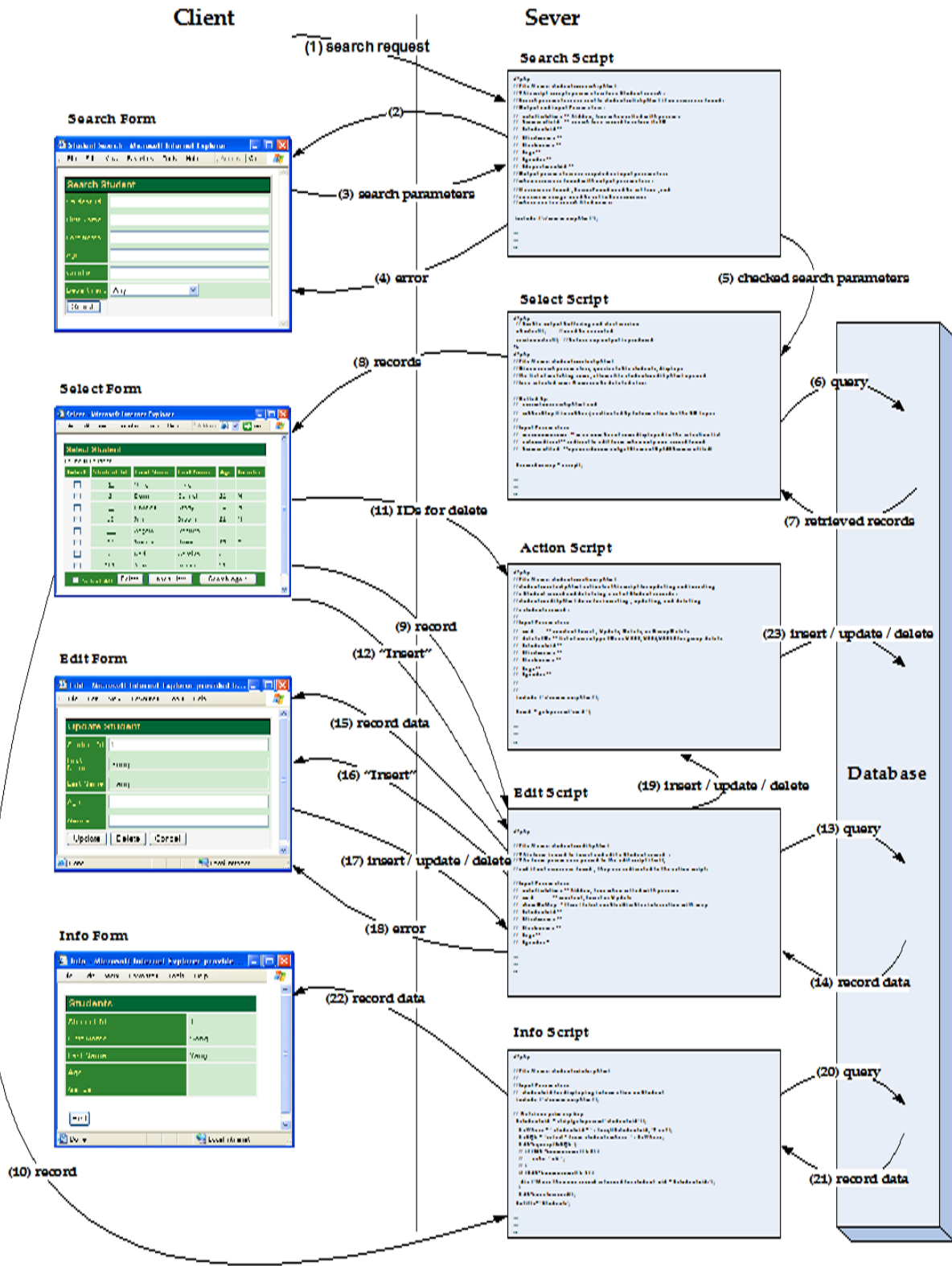


FIG. 5: Interactions entre les scripts web et les formulaires

Department Select Form

Select	Department ID	Department Name	Description
<input type="checkbox"/>	1	Computer Science	Department of Computer Science
<input type="checkbox"/>	2	Mathematics	Department of Mathematics

Department Search Form

Student Edit Form

Department Edit Form

Course Select Form

Select	Student ID	Course ID	Course Name	Grade
<input type="checkbox"/>	21	1	Databases	D
<input type="checkbox"/>	21	2	Programming	C
<input type="checkbox"/>	21	3	Operating System	B
<input type="checkbox"/>	21	4	Software Engineering	C

FIG. 6: Formulaire d'édition d'un Etudiant

un à un ou *plusieurs à un*. Dans notre exemple, le champ `department_id` de la table `students` est spécifié comme étant de type `to_one`. Lorsque l'on clique sur le bouton `View`, le formulaire d'édition montrant l'enregistrement correspondant au département de l'étudiant est affiché. De plus, un utilisateur peut chercher et sélectionner un nouveau département à lier. L'option `linked_table` désigne le nom de la table qui stocke l'enregistrement associé. L'option `child_column` indique le champ à utiliser pour lier dans la table associée. Si cette option est omise le nom du champ utilisé pour le lien est identique à celui du champ clé étrangère de l'enregistrement actuel.

to_many : L'objectif de ce type est de lister les enregistrements liés à l'actuel via une relation du type *un à plusieurs* ou *plusieurs à plusieurs*. Dans notre exemple, le champ étiqueté `Cours` suivi est de type `to_many`. Lorsque l'on clique sur le bouton `Show`, tous les cours pris par l'étudiant sont affichés dans le formulaire de sélection des cours. Ce type nécessite la présence des options `linked_table`, `parent_column` et `child_column`. Les options `linked_table` et `child_column` peuvent être omises lorsque le nom du champ de la clé étrangère de la table associée est identique à celui du champ de la clé primaire de l'enregistrement actuel. L'option `parent_column` indique le champ de la clé étrangère dans la table actuelle. Si cette option est omise, le champ de la clé étrangère correspond au champ de la clé primaire de la table actuelle.

Génération automatique des fichiers de configuration des couches géographiques

On peut créer un fichier de configuration des couches géographiques de manière semi-automatique en utilisant les informations de la table `geometry_columns` qui stocke les métadonnées liées au SIG et le nom de la couche dans le fichier de configuration de la carte.

1. Le *méta-fichier de configuration des couches géographiques* utilisé par défaut et qui liste tous les groupes de couches, `xxx_maplayers.mconfig`, peut être créé avec **map_mconfgen**. Les noms des groupes de couches sont générés à partir de des noms des tables stockés dans la table

`geometry_columns` et à partir des noms des groupes de couches définis dans le fichier de configuration de la carte.

2. On peut modifier le fichier de méta-configuration des couches utilisé par défaut et dont le format est identique à un fichier de configuration des couches, en sélectionnant et en changeant l'ordre des groupes de couches. On peut également fournir des valeurs personnalisées pour les paramètres de chaque groupe de couches. Le fichier de méta-configuration a uniquement besoin de garder les définitions des paramètres personnalisés.
3. Enfin, le fichier de configuration des couches `xxx_maplayers.config` est généré par **map_mconfgen**. Les informations nécessaires à cette étape sont récupérées du méta-fichier de configuration, de la table `geometry_columns` et du fichier de configuration de la carte. Les valeurs par défaut des paramètres, déterminées par la table `geometry_columns` et du fichier de configuration de la carte, sont écrasées si des valeurs personnalisées sont présentes dans le fichier de méta-configuration.

Le fichier de méta-configuration des couches permet de définir n'importe quel paramètre accepté dans le fichier de configuration des couches. Cependant, le fichier que **map_mconfgen** génère par défaut contient uniquement, pour chaque groupe de couches, les définitions des paramètres suivants :

layer_selectable – Une option booléenne qui détermine si la couche peut être sélectionnée pour des opérations spatiales ou pas.

legend_label – Le nom de la couche à afficher sur la légende de la carte.

Voici un exemple de fichier de méta-configuration généré par défaut :

```
$layer_groups = array(
  'grp_eo_py' => array(
    'layer_selectable' => true,
    'legend_label' => 'Eo Py',
  ),
  'grp_eosrc' => array(
    'layer_selectable' => true,
    'legend_label' => 'Eosrc Pt',
  ),
  ...
)
```

L'utilisateur peut modifier les valeurs de ces définitions et en ajouter d'autres. Par exemple le fichier de méta-configuration peut être modifié de la manière suivante :

```
$layer_groups = array(
```

```

'grp_eo_py' => array(
  'layer_selectable' => true,
  'legend_label' => 'EO Polygons',
  'search_script' => 'forms/eo/eo_py_eo_search.phtml',
  'select_script' => 'forms/eo/eo_py_eo_select.phtml',
  'edit_script' => 'forms/eo/eo_py_edit.phtml',
  'img_src' => 'images/eo_poly.png',
),
'grp_eosrc_pt' => array(
  'layer_selectable' => true,
  'legend_label' => 'EOSRC Points',
  'search_script' => 'forms/eo/eosrc_pt_search.phtml',
  'select_script' => 'forms/eo/eosrc_pt_select.phtml',
  'edit_script' => 'forms/eo/eosrc_pt_edit.phtml',
  'img_src' => 'images/eosrc_pt.png',
),
...

```

Les modifications suivantes ont été effectuées :

1. legend_label a été changé pour le nom souhaité de la couche dans la légende de la carte.
2. Les noms et la localisation des formulaires de recherche, de sélection et d'édition sont spécifiés respectivement dans search_script, select_script et edit_script.
3. Le nom du fichier source de l'icône dans la légende de la carte a été spécifié dans img_src.

L'ordre des groupes de couches peut également être changé pour correspondre à celui souhaité dans la légende de la carte.

Ensuite le fichier de configuration des couches montré précédemment peut être généré par `map_confgen`.

Historique du Développement de WebGD

Le framework WebGD et WebGD-Gen ont été développés de manière *incrémentielle* et *itérative* au cours des quatre dernières années. En 2000, nous avons tout d'abord implémenté une application qui permettait d'ajouter des entités de type ponctuel sur une carte en utilisant ASP avec ArcIMS et ArcSDE. En 2001, nous avons re-développé cette application avec ASP.NET, vu qu'ASP.NET propose des contrôles web qui constituent de meilleures briques pour la construction de pages web. Sur la base de cette application, la première version du framework WebGD a vu le jour en 2002, afin de supporter de multiples applications.⁽¹⁵⁾

Début 2003, nous avons re-développé une application appelée Motels Oregon avec MapServer, PostGIS et PostgreSQL⁽⁹⁾. Cette version Linux était non seulement plus fiable et rapide que la précédente

mais également basée sur des logiciels libres. En développant l'application MapServer suivante, un système de gestion de la ressource en germplasm (GEM-GIS), la première version du framework WebGD pour MapServer a été créée. Ce framework a ensuite été amélioré pour qu'il intègre la gestion des entités de type polygone aussi bien que celles de type ponctuel.

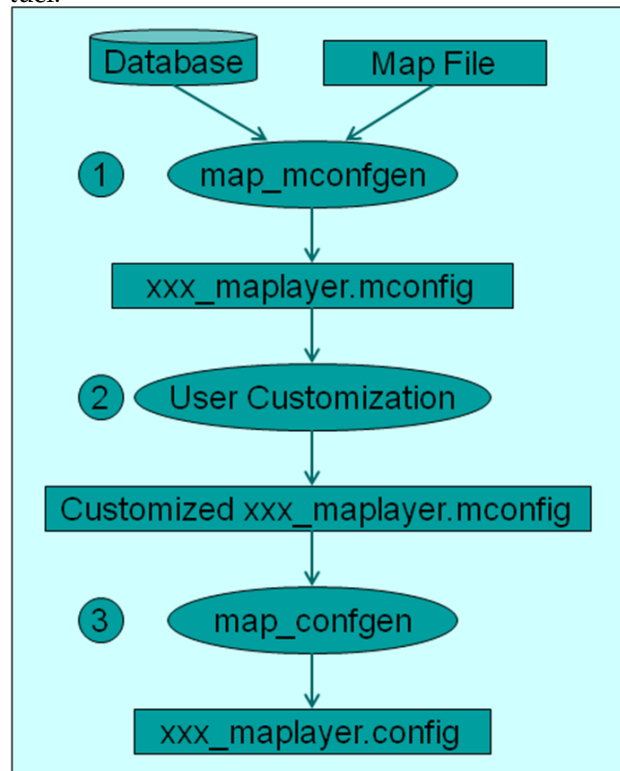


FIG. 7: Processus de création d'un fichier de configuration des couches

Deux améliorations majeures ont été faites en 2004 sur le framework WebGD : le *changement dynamique* de référence spatiale pour différentes régions et la *génération automatique* de formulaires web utilisés pour les ajouts, les interrogations et les suppressions d'entités géographiques. Comparé à une application qui affiche simplement des entités géographiques telles que des points sur une carte, le framework actuel est environ 20 fois plus complexe en terme de temps passé à développer les fonctionnalités nécessaires.

Conclusions et travaux à venir

Nous avons développé le framework WebGD et le générateur d'applications WebGD-Gen pour développer rapidement des applications SIG/base de

données Internet.

1. Les entités géographiques, telles que les habitats de plantes ou d'animaux, les zones de chantiers routiers ou des tronçons hydrographiques, peuvent être insérées, interrogées et supprimées grâce à l'interface cartographique et les formulaires web qui s'affichent sur un navigateur web standard.
2. Une application peut être créée sans programmer une seule ligne de code. L'interface cartographique et les scripts web pour l'accès aux données peuvent être générés automatiquement à partir de fichiers de configuration et ces fichiers de configuration peuvent eux-même être générés à partir du schéma de base de données et les métadonnées sur les données SIG, par exemple, les informations stockées dans la table `geometry_columns`. La génération automatique d'une application SIG Internet ne réduit pas seulement les coûts de développement de manière significative mais facilite également le développement incrémentiel et itératif de l'application.
3. Le changement dynamique de système de référence spatial permet à une application de couvrir différentes régions ayant des fichiers de configuration de la carte, des projections, des légendes de carte et des listes de raccourcis différents. Il s'agit d'une fonctionnalité importante pour une application qui couvrirait par exemple l'ensemble des Etats-Unis ou du monde.
4. Nous avons créé le framework WebGD en utilisant uniquement des logiciels libres et open source. Les outils logiciels que nous utilisons, tels que MapServer de l'Université du Minnesota, le SGBD PostgreSQL, PostGIS, Apache et PHP sont tous disponibles gratuitement. Les données SIG utilisées provenant de l'USGS, de TIGER/LINE et de Digital Chart of the World (DCW) font également partie du domaine public. Le framework est disponible à tous en utilisation libre.
5. Les coûts pour faire tourner nos applications sont extrêmement bas. Nous pouvons mettre des copies de grosses bases de données telles que Biotics, les données sur les sols de SSURGO2 et une partie du Système d'Information National sur la Ressource en Germplasm sur un PC à \$800.

La génération automatique du code d'une application WebGD permettra d'économiser beaucoup d'efforts dans le développement de systèmes spatiaux

d'aide à la décision. Bien que quelques personnalisations sont nécessaires, le temps à y accorder peut être diminué à des semaines ou des mois comparé aux années nécessaires pour développer un *système spatial d'aide à la décision* depuis le début.

Le framework WebGD et WebGD-Gen sont actuellement disponibles sur demande. Nous cherchons des collaborateurs pour le rendre publique en tant que logiciel libre et open source. Nous sommes également en train de re-développer l'interface cartographique en utilisant OpenLayers afin de supporter les déplacements fluides sur la carte.

Bibliographie

- [1] P. H. Dana Map Projection Overview http://www.colorado.edu/geography/gcraft/notes/mapproj/mapproj_f.html
- [2] DM Solutions Group Inc. PHP MapScript <http://www.maptools.org>
- [3] R. Kingston (1998) Web-based GIS for public participation decision making. In *Procs of NCGIA PPGIS Meeting*, Santa Barbara, California. Retrieved Map 2003 from <http://www.ncgia.ucsb.edu/varenius/ppgis/papers/kingston/kingston.html>
- [4] Eum D. and Minoura T. (June 2003) Web-based database application generator. *IEICE Transactions on Information and Systems*, Vol. E86-D, No. 6.
- [5] Fogelson, C. (2002) Biotics 4.0 data model version 1.0. Retrieved January 5, 2004, from <http://whiteoak.natureserve.org/hdms/HDMS-DataModel.shtml>
- [6] McKenna, Jeff MapServer PHP/MapScript Class Reference - Versions 3.6, 4.0 & 4.2 DM Solutions Group Inc.
- [7] NatureServe (February 2002) Element Occurrence Data Standard. Retrieved January 4, 2004, from <http://whiteoak.natureserve.org/eodraft/all.pdf>
- [8] NatureServe (December 2003) Biotics 4.0 Getting Started Guide. Retrieved January 5, 2004, from <http://whiteoak.natureserve.org/eodraft/all.pdf>

- naturereserve.org/hdms/biotics-learn-more.shtml (now obsolete).
- [9] Sano J., Wanalertlak N., Maki A., and Minoura T. (July 2003) Benefits of web-based GIS/database applications. In *Proc. of 2nd Annual Public Participation GIS Conference* Portland, Oregon.
- [10] USDA-ARS Western Regional Plant Introduction Station, USDA - Agricultural Research Service, Pullman, Washington <http://www.ars-grin.gov/ars/PacWest/Pullman/>
- [11] Ramsey, Paul PostGIS Manual Refractions Research Inc.
- [12] University of Minnesota (2003) MapServer <http://mapserver.gis.umn.edu>
- [13] Sharma, A. (December 2003) Web-based analysis module for a germplasm collection. Master of Science report, School of Electrical Engineering and Computer Science, Oregon State University
- [14] Wangmutitakul P, Li L, and Minoura T. (March 2003) User Participatory Web-Based GIS/Database Application. *Proc. of Geotec Event Conference*
- [15] Wangmutitakul Paphun et al. (2004) Framework for Web-based GIS/database Applications *Journal of Object Technology* 3, 209-225
- [16] Wuttiwat T., Minoura T. and Steiner J. (May 2003) Using Digital Orthographic Aerial Images as User Interfaces *Proc. of ASPRS Annual Conference*, Anchorage, Alaska

Toshimi Minoura, Nirut Chalainanont, Junya Sano
Oregon State University
<http://enr.oregonstate.edu/~minoura/research/creeda/>
minoura AT eecs.orst.edu

The [Open Source Geospatial Foundation](#), or OSGeo, is a not-for-profit organization whose mission is to support and promote the collaborative development of open geospatial technologies and data. The foundation provides financial, organizational and legal support to the broader open source geospatial community. It also serves as an independent legal entity to which community members can contribute code, funding and other resources, secure in the knowledge that their contributions will be maintained for public benefit. OSGeo also serves as an outreach and advocacy organization for the open source geospatial community, and provides a common forum and shared infrastructure for improving cross-project collaboration.

Publié par l'OSGeo, le Journal de l'OSGeo a pour objectif de publier les résumés des conférences, étude de cas et introduction, et les concepts liés à l'open source et aux logiciels en géomatique.

Équipe éditorial :

- Angus Carr
- Mark Leslie
- Scott Mitchell
- Venkatesh Raghavan
- Micha Silver
- Martin Wegmann

Rédacteur en Chef :

Tyler Mitchell - [tmitchell AT osgeo.org](mailto:tmitchell@osgeo.org)

Remerciements

Divers relecteurs & le project GRASS News

Le *Journal de l'OSGeo* est une publication de la *Fondation OSGeo*. La base de ce journal, le source des styles L^AT_EX 2_ε a été généreusement fournie par le bureau d'édition de GRASS et R.



Ce travail est sous licence Creative Commons Attribution-No Derivative Works 3.0 License. Pour lire une copie de cette licence, visitez : creativecommons.org.



All articles are copyrighted by the respective authors — contact authors directly to request permission to re-use their material. See the OSGeo Journal URL, below, for more information about submitting new articles.

Journal en line : <http://www.osgeo.org/journal>

OSGeo Homepage : <http://www.osgeo.org>

Courrier postal : OSGeo

PO Box 4844, Williams Lake,

British Columbia, Canada, V2G 2V8

Téléphone : +1-250-277-1621



ISSN 1994-1897

This PDF article file is a sub-set from the larger
OSGeo Journal. For a complete set of articles
please the Journal web-site at:

<http://osgeo.org/journal>