
Journal de l'OSGeo

Le Journal de la Fondation Open Source Geospatial

Volume 1 / Mai 2007

Dans ce volume

Développement de logiciels Open Source

Introduction à Mapbender, deegree, openModeller ...

Comprendre les relations spatiales

Examen de la spécification du Web Processing Server (WPS)

Interaction des logiciels - GRASS-GMT, Tikiwiki, PyWPS, GRASS-R ...

Mises à jour des logiciels

Actualités, et plus ...



Éditorial

Éditorial

Notre nouveau journal

Bienvenue au numéro 1 de notre tout nouveau journal ! Cette nouvelle publication a pour vocation de mettre en lumière les nombreuses réussites des communautés Open Source et géospatiales, et en particulier celles de l'Open Source Geospatial Foundation (OSGeo)¹.

Le but de ce journal est de tenir au courant, d'informer et d'instruire tout type de lecteurs. Il fournira des informations sur les évolutions de projets, des études de cas, des tutoriaux et bien d'autres choses. Il contribuera en particulier à décrire les évolutions de projets OSGeo et les projets à venir de l'association.

Ce numéro est destiné à remplacer et à élargir les thèmes du journal GRASS-News² qui ne paraît plus, mais proposait de solides tutoriaux techniques et de nombreuses opportunités d'interaction au sein de la communauté GRASS. Ce nouveau journal va continuer à offrir des informations sur GRASS, tout comme sur de nombreux autres projets, qu'ils soient supportés par l'OSGeo ou non.

Appel à contibution

Vous pouvez contribuer en nous soumettant des articles ou des idées d'articles pour le prochain nu-

méro. Il y a aussi de la place pour ceux qui veulent nous aider pour la rédaction, la mise en page ou la relecture. Pour nous proposer votre aide, remplissez le formulaire "commentaires"³ sur le site web de l'OSGeo.

Le prochain numéro conservera cette ouverture : il publiera des tutoriaux de programmation, des présentations de nouveaux projets, des descriptions d'études de cas, des évolutions de projets, etc. Les articles seront regroupés par thème (ex : tutoriaux, actualités, études de cas...) pour proposer aux lecteurs un large choix de thèmes et d'outils.

Ce fut un plaisir de travailler avec toute l'équipe éditoriale et les rédacteurs. Ils ont travaillé dur à chercher du contenu, écrire des articles, rassembler des informations, corriger et mettre en forme. Ils ont ainsi contribué à rassembler des informations très intéressantes dans un laps de temps très court. Inutile de dire que j'attends avec impatience la parution du numéro 2 dans les mois à venir. J'espère juste à temps pour la **conférence annuelle FOSS4G**⁴.

Tyler Mitchell

Rédacteur en chef

<http://osgeo.org>

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

¹Site web de l'OSGeo (Open Source Geospatial Foundation) : <http://www.osgeo.org>

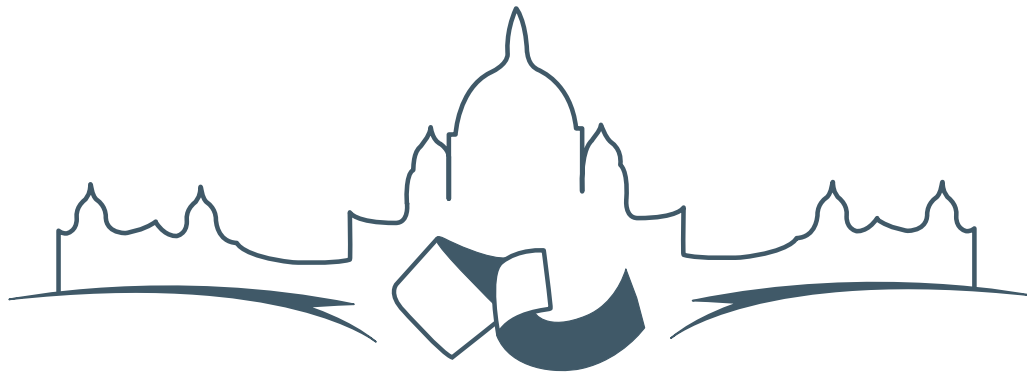
²Site web GRASS-News : <http://grass.itc.it/newsletter/index.php>

³Formulaire de soumission de commentaires sur le journal : <http://www.osgeo.org/feedback/Journal>

⁴Site web FOSS4G 2007 : <http://www.foss4g2007.org>

Table des matières

Éditorial	1	Étude de cas	30
FOSS4G - Ouverture des Inscriptions à la Conférence	3	MapGuide OS dans le projet de cartographie de la forêt urbaine de San Francisco	30
Actualité	4	Étude d'intégration	33
Brèves	4	Créer des cartes prêtes à être publiées avec GRASS et GMT	33
Compte rendu d'événements	8	Utiliser l'interface R— GRASS	41
Rencontres Utilisateurs OSGeo Caroline du Nord	8	Traitement géospatial sur des serveurs distants via Internet – PyWPS	44
Compte-rendu de l'OSGeo Inde au Map World Forum 2007	10	Tikiwiki un GeoCMS	48
8 ^{ème} Rencontre Annuelle des utilisateurs Ita- liens de GRASS & GFOSS	12	Étude thématique	52
Zoom sur un projet	14	Relations spatiales dans les SIG - Une introduc- tion	52
QLandkarte	14	Évaluation du Web Processing Service de l'OGC	55
Une courte introduction à Mapbender	17	Annonce des développeurs	64
deegree	21	Mise à jour des développeurs	64
Introduction à OpenModeller	24	FOSS4G - Ouverture des Inscriptions à la Conférence	77



**2007 FREE AND OPEN SOURCE SOFTWARE
FOR GEOSPATIAL (FOSS4G) CONFERENCE**
VICTORIA CANADA  SEPTEMBER 24 TO 27, 2007

FOSS4G - Ouverture des Inscriptions à la Conférence

Nous sommes heureux de vous annoncer l'ouverture des inscriptions en ligne à la Conférence Free and Open Source Software for Geospatial 2007 (FOSS4G 2007). FOSS4G est l'évènement annuel qui réunit les personnes et les sociétés qui créent, utilisent, et gèrent des logiciels géospatiaux open source. Inscrivez-vous dès maintenant en ligne.⁵

Inscrivez-vous avant la date limite du 27 Juillet, pour économiser sur les frais d'inscription! Tirez profit de l'opportunité que FOSS4G 2007 vous offre, de construire un réseau avec les autres professionnels des données géospatiales, de renouveler d'anciennes relations, et d'en créer de nouvelles.

Pour les dernières mises à jour, l'inscription et/ou la soumission d'une présentation, visitez le site web de la conférence.⁶

OPPORTUNITES D'EXPOSITION & DE SPONSORING

Concernant les opportunités d'exposition et de sponsoring, lisez la page des partenaires⁷ ou contac-

tez Paul Ramsey, Président de la Conférence par email.⁸

SOUMETTRE UNE PRESENTATION

Vous pouvez soumettre une présentation en ligne.⁹ La date limite pour les soumissions est le 29 Juin 2007.

Les présentations FOSS4G durent 25 minutes, avec 5 minutes de questions/réponses à la fin. Les présentations concernent l'utilisation ou le développement de logiciels géospatiaux opensource. Tout le monde peut soumettre une proposition de présentation et participer à la conférence comme présentateur. Plus d'informations sont disponibles sur la page des présentations sur le site web.

Nous espérons vous voir à Victoria, au Canada en Septembre!

⁵Inscription en ligne : <http://www.foss4g2007.org/register/>

⁶Site web de la conférence : <http://www.foss4g2007.org/>

⁷Page des partenaires : <http://foss4g2007.org/sponsors>

⁸Email Paul Ramsey : pramsey@foss4g2007.org

⁹Soumettez une présentation sur <http://www.foss4g2007.org/presentations/>

Actualité

Brèves

Une collection des annonces et actualités les plus récentes

Compilé par Jason Fournier, traduit par Gwenal Bachelot et Yves Jacolin

Premier anniversaire de l'OSGeo

Le 4 février 2007 fut le premier anniversaire de l'OSGeo. Ce jour là, 25 personnes se rencontrent, et beaucoup plus par téléphone et IRC, pour discuter de la possibilité de lancer une organisation généraliste. Ces participants représentaient plus de 13 projets open source différents. Les objectifs de base de l'organisation était d'aider la promotion et de continuer le développement des outils open source dans la sphère géospatiale. Depuis ce temps, beaucoup de chose sont arrivé et l'élan autour de l'OSGeo continue à se développer.

Pour un aperçu des nombreuses actualités depuis la première année, visitez s'il vous plait le blog de Tyler.¹⁰

¹⁰Tyler's anniversary blog article : http://www.osgeo.org/tyler/osgeo_1st_anniversary

¹¹Idée de projet pour le Summer of code : http://wiki.osgeo.org/index.php/2007_SoC_Merged_Ideas

¹²Site du SOC : <http://code.google.com/soc/>

¹³Annonce en ligne du SOC : <http://osgeo.org/news/soc2007>

OSGeo accepté pour le Google Summer of Code 2007

OSGeo est heureuse de vous annoncer que Google a accepté l'OSGeo comme organisation tuteur pour le programme du Google Summer of Code 2007. Ce programme fournit un financement pour des étudiants pour travailler sur des projets open source sous la direction de tuteurs expérimentés. Les projets participants à travers l'OSGeo sont GDAL, GeoServer, GeoTools, GRASS, MapGuide, PostGIS, uDig, OpenJUMP et MapServer. Les étudiants intéressés pour une participation peuvent trouver plus d'informations sur les idées des projets de l'OSGeo sur le wiki ¹¹ et plus d'informations sur le programme dans son ensemble sur le site du SOC de Google ¹².

Source : Liste OSGeo-Announce et site de l'OSGeo ¹³

Actualité des sponsors

En plus de la poursuite de la Sponsorisation par Autodesk, de nouveaux Sponsors Associés incluant Lizardtech, 1Spatial et First Base Solutions :

- Autodesk - <http://www.autodesk.com>
- Lizardtech - <http://lizardtech.com>
- 1Spatial - <http://www.1spatial.com>
- First Base Solutions -
<http://www.firstbasesolutions.com>

Sponsorisation de GDAL

Il y a également plusieurs sponsors qui ont fourni un support pour le développement de GDAL/OGR à travers le programme de sponsorisation de l'OSGeo.

ACTUALITÉS DES PROJETS

Cette section présente des brèves sur divers projets. Pour des informations plus détaillées voyez la section Annonces des développeurs à la page 64.

MapGuide

Mars 2007 - MapGuide a passé le processus d'incubation de l'OSGeo, en montrant le niveau requis de participation, de liberté sur la propriété du code source, et d'usage des méthodologies standard de développement open source. MapGuide Open Source en est à la version 1.1.0.

MapServer

L'équipe de développement de MapServer a prévu de sortir la version 5.0 pour la fin de l'été 2007. La planification de la version 5.0 présente un certain nombre de mise à jour prévues :

- Gestion de la bibliothèque AGG, un nouveau moteur de rendu d'image comme alternative à GD
- Mise à jour de la gestion du WMS pour la version 1.3.0
- Amélioration de la gestion du serveur SOS de l'OGC
- Nouvelle gestion de l'OWS commun de l'OGC version 1.0.0
- Possibilité de création de graphique dynamique
- Prioritisation des étiquettes
- Correction de la gestion de la mémoire de MapScript

- Restructuration du mécanisme de rendu de LOG/DEBUG

La planification de la version de MapServer¹⁴ pour la version 5.0 est en ligne.

Mapbender

Mapbender a récemment annoncé une série de nouvelles fonctionnalités ainsi qu'une future planification des évolutions pour de nouveaux développements, parmi eux :

- Intégration d'OpenLayers comme un module Mapbender optionnel prévu pour 2007
- Amélioration de la numérisation / WFS-T : prise en charge de géométries complexes
- Utilisation plus large d'AJAX /JSON et de la technologie Orienté Objet : amélioration des interfaces de modules pour rendre justice à l'audience du développement plus large.
- Évaluation des algorithmes : dû à une minimisation du trafic, il y a très peu de commentaires du code source. JSDoc et PHPDoc seront utilisés pour les commentaires dans le futur. Les problèmes de trafic seront résolus en récupérant les commentaires avant le déploiement et en dépolyant JS-code.

GDAL/OGR

GDAL/OGR a annoncé sa version 1.4.1 au dernier trimestre. Il a lancé un programme de sponsorship par l'OSGeo. Les premiers sponsors sont Analytical Graphics, Applied Coherent Technology, SRC, Safe Software, Cadcorp, Waypoint, MicroImages et i-Cubed.

GeoNetwork

Mise à jour sur les développements du premier trimestre 2007. Récemment, le travail s'est focalisé sur quelques aspects du projet. Ceux-ci sont :

- Développement de la version 2.1 de GeoNetwork opensource résultant à un vertion bêta.
- Mise à jour du site communautaire
- Travail lié au processus d'incubation de l'OSGeo

¹⁴Planification de la version de MapServer : http://mapserver.gis.umn.edu/development/release_plans/mapserver_5_0

GRASS GIS

Mise à jour du projet en 2007 - Trim. 1 :

- Rencontre des utilisateurs Italien de GRASS et GFOSS - La rencontre des utilisateurs de GRASS et GFOSS, Palerme (Italie), 14-16 Fév. 2007
- 12 Fév. 2007 : nouveau tracker de bug et de fonctionnalité pour GRASS - basé sur Gforge
- 10 Fév. 2007 : binaire pour GRASS GIS 6.2.1 winGRASS/Cygwin disponible

Quantum GIS

Le projet QGIS est entré officiellement en incubation en mars 2007. Cela a démarré le processus d'incubation pour prendre en compte QGIS comme un membre à part entière de l'OSGeo. L'OSGeo incubé un grand nombre de projet et QGIS rejoint GRASS et OSSIM dans la catégorie des applications de bureau.

ÉVÉNEMENTS PROCHES

Voici quelques uns des événements à venir où l'OSGeo et ses membres seront présents pour les prochains mois.

III Rencontre National des Utilisateurs de MapServer, Brésil

9-11 mai 2007 Ce sont des conférences en géomatiques open source et sur MapServer à Brasilia, Brésil. Voyez <http://www7.univali.br/elis2/enum07/> pour des informations en anglais et portugais.

Where 2.0, San-José

29-30 mai 2007 La conférence Where 2.0 d'O'Reilly situé à San-José, Californie, USA. Pour plus d'information : <http://conferences.oreillynet.com/where2007/>.

OSCON 2007, Portland

23-27 juillet 2007 La Convention Open Source d'O'Reilly situé à Portland, Oregon, USQ. Plus d'informations sur : <http://conferences.oreillynet.com/os2007/>.

Map Asia 2007, Kuala Lumpur

14-16 août 2007 6ème Salon et Conférence Annuelle Internationale sur les technologies et applications de l'Information géographique. Le thème de la conférence est *Maponomics — Croissance économique par la technologie de la géo-information*. Map Asia a mis en place une initiative international dont le but est de fournir une plateforme pour la convergence, le partage et l'utilisation de technologies géospatiales. Fondé durant l'année 2002, Map Asia est le plus grand Salon et Conférence Annuel Asiatique dans le domaine de SIG, GPS, photographie aérienne et récepteurs distants. Situé à Kuala Lumpur, Malaisie. Pour plus d'informations : <http://www.mapasia.org>.

FOSS4G 2007, Victoria

24-27 septembre 2007 La Conférence Annuelle des Logiciels Libres et Open Source pour la Géomatique (FOSS4G) rassemble les personnes qui crée, utilise, et supporte les logiciels ouverts en géomatique. Aucun autre événement ne rassemble autant de membres du développement open source, création de données ouvertes, et les communautés de promotion des standards ouverts. L'événement est situé à Victoria, Colombie-Britannique, Canada. Pour plus d'information visitez s'il vous plait la page officiel du FOSS4G 2007 : <http://www.foss4g2007.org/>.

INTERGEO, Leipzig

25-27 septembre 2007 Conférence et foire commerciale pour la géodésie, la géoinformation et la gestion des sols situé à Leipzig, Germany. Ils espèrent 17 000 visiteurs de plus de 80 pays. Plus d'information sur : <http://www.intergeo.de>.

BUREAU

Résumé des rencontres du Bureau des Directeurs de l'OSGeo :

- Confirmation d'Howard Butler comme nouveau SAC chair
- Approbation de la formation du local chapter de New Mexico USA
- Approbation de la motion pour MapGuide OS du succès de l'incubation
- Approbation de la motion pour que QGIS et FDO entre en phase d'incubation

-
- | | |
|--|--|
| <ul style="list-style-type: none">- Gestion de la participation de l'OSGeo au Google Summer of Code- Nomination de Tyler comme secrétaire de l'OSGeo pour remplacer Rich, et comme Chief Returning Officer pour les élections- Approbation des procédures d'élection des | <ul style="list-style-type: none">membres et du Bureau- Demande du CRO de démarrer la procédure d'élection des membres de la charte (Charter Membership)- Examen financier |
|--|--|

Compte rendu d'événements

Rencontres Utilisateurs OSGeo Caroline du Nord

Après la Conférence SIG de Caroline du Nord (NC) 2007

par Julie Harrell, traduit par Sidonie Christophe

Deux fois par an, les professionnels du SIG de Caroline du Nord organisent une conférence SIG à l'échelle de l'Etat pour présenter leurs multiples projets modèles et leurs initiatives. Cette année, c'était les 20 ans de la Conférence SIG de Caroline du Nord, un des plus anciens événements en SIG d'un Etat. Il y a eu plus de 950 participants, 159 présentateurs et 58 exposants.

Le 1er Mars un petit groupe dédié de professionnels SIG ouverts, mené par le DR. Helena Mitsova (du très réputé projet GRASS), a tenu la première rencontre utilisateurs OSGeo affiliée Rencontre utilisateurs SIG Open-Source de Caroline du Nord, pendant la Conférence SIG NC 2007 à Winston Salem, NC, USA.

Il y a eu approximativement 20 participants, dont une majorité fait partie du groupe de l'Université qui s'est engagé pour l'existence d'une future liste de discussion. Il y a eu aussi une poignée de représentants des Gouvernements Local, Fédéral et de l'Etat. Deux personnes d'un important fournisseur commercial de logiciels SIG étaient aussi présentes.



FIG. 1 – Helena introduit l'OSGeo et présente le site web wiki de l'OSGeo

La rencontre informelle de fin de journée s'est tenue après la principale conférence SIG, et les sujets suivants ont été traités :

Message de bienvenue (Julia Harrell, Coordinatrice SIG, NC Département de l'Environnement et des Ressources Naturelles - NC DENR)

Courte introduction de l'organisation de l'OSGeo et discussion sur le groupe utilisateurs NC affilié OSGeo et ses missions (Dr. Helena Mitsova, Professeur de Recherche Associé au Département des Sciences Marines, Terrestres et Atmosphériques de l'Université de l'Etat de

Caroline du Nord)

Courte présentation et démo de QGIS et uDig SIG Open Source type Client lourd *Doug Newcomb, Administrateur Système au US Fish and Wildlife Services Raleigh, NC Office*)

Discussion sur les potentialités d'offrir un cursus SIG Open Source et une journée pendant l'été, consacrée à un atelier SIG Open Source (*Rodney Jackson, Directeur du Geospatial Technology Corporate et du Programme de Formation Continue au Central Piedmont Community College in Charlotte, NC*)

Annnonce de la Deuxième Conférence Annuelle sur les SIG Open Source à venir (OSG-SD) à l'Université de l'Etat de San Diego, et rappel des événements de l'année passée (*Dana Nibby, Analyste Base de Données SIG au NC Department of Transportation GIS Unit*)

Démo de GAIA3 Une application libre et interopérable pour gérer les réponses aux incidents créée par un "FGDC CAP 2006" (Federal Geographic Data Committee) concédé par The Carbon Project, NC DENR, la ville de Charlotte, et les Départements SIG de Wake County (*Jeff Harrison, Président & CEO, The Carbon Project*)

Annnonce de la prochaine journée Open Source dédiée aux présentations de la Conférence SIG (*Julia Harrell, Coordinatrice GIS, NC DENR*)

Texas Mesonet : Traitement de masses de données en temps réel (*Gerry Creager, Texas A&M University*)

Services Web géospatiaux et déroulements des opérations scientifiques pour la Distribution et la Modélisation de Données sur les Mammifères Marins (*Dr. Patrick Halpin, Duke University*)

Développement d'un Visualisateur de données côtières NCCOOS : Challenges des SIG marins en temps réel (*Jesse Cleary, UNC Chapel Hill*)

Services Web OpenGIS et "Service Orienté Architecture" (*Tobin Bradley, GISP, Mecklenburg County, NC*)



FIG. 2 – Participants du groupe d'utilisateurs OSGeo NC

Cartographie sur le Web de grande valeur et à prix réduit

Un tourbillon de solutions SIG Open source et libres

Liste des présentations :

PostgreSQL/PostGIS : Bases de données relationnelles intégrant le spatial (*Tobin Bradley, GISP, Mecklenburg County NC*)

PrimaGIS : Système coopératif de cartographie pour le CMS Plone (*Chris Calloway, UNC Chapel Hill*)

Clients Web : Communauté MapBuilder & MapBender (*Julia Harrell, NC DENR*)

Minnesota Mapserver, Chameleon : (*Tom Melhuish, WebInsights*)

GRASS GIS : (*Dr. Helena Mitasova, North Carolina State University*)

Le meilleur des autres technologies : GDAL/OGR, CartoWeb, GeoServer, MapGuide Open Source (*Doug Newcomb, US Fish & Wildlife Service*)

Mashups : Les perspectives non lucratives et communautaires (*Wansoo Im, The Center for Community Mapping*)

Southeastern Gap Analysis : Produits et accessibilité (*Todd Earnhardt and Jim White, NC State University*)

Econet : Réseau Environnement et Climat de Caroline du Nord pour la modélisation temps-réel du climat (*Ryan Boyles, North Carolina State Climatologist, NC State University*)

Note : Toutes les présentations de la Conférence SIG NC 2007 seront disponibles au téléchargement depuis le site web de la Conférence SIG NC¹⁵ dans quelques semaines.

Julia Harrell
Coordinatrice SIG, NC DENR
[julia.harrell AT Ncmail.net](mailto:julia.harrell@ncmail.net)

¹⁵Site web de la Conférence SIG NC : <http://www.cgia.state.nc.us/ncgis2007>

Compte-rendu de l'OSGeo Inde au Map World Forum 2007

par K. S. Rajan, traduit par Sidonie Christophe

L'OSGeo Inde a activement participé et était bien représentée au Map World Forum 2007 qui s'est tenu à Hyderabad du 22 au 25 Janvier 2007. Les activités principales proposées par l'organisation de ce premier forum international de la communauté géospatiale ont été les suivantes – (i) "Keynote address on OSGeo" par Gary Lang, Trésorier OSGeo (ii) Atelier de travail OSGeo et (iii) Stand d'exposition OSGeo. La première rencontre du Conseil Exécutif de l'OSGeo Inde s'est aussi tenue à ce moment.

Atelier de travail OSGeo

L'atelier de travail OSGeo a eu lieu le 24 Janvier, pendant la session de l'après-midi. L'objectif principal de cet atelier était de fournir une plateforme de communication et d'éducation de la communauté géospatiale sur les importantes opportunités que FOSS4G propose, les avancées et le niveau de développement technologique afin de fournir des solutions géospatiales efficaces. Pendant l'atelier, différents modèles de développement et d'utilisation des outils FOSS ont été présentés, ainsi que leur potentiel dans les domaines académiques, gouvernementaux (principalement pour les organisations mandatées pour du développement national et de l'e-gouvernance), et d'affaire. D'autres pays d'Asie, comme la Chine, le Japon et le Vietnam étaient aussi présents à cet atelier.

Le professeur Deekshatulu, Ex-Directeur de la National Remote Sensing Agency (Inde) a prononcé le discours inaugural et mis en évidence les avantages fournis par les outils FOSS pour un pays aussi grand que l'Inde, et il a en particulier souligné les avantages des outils FOSS4G pour les différents preneurs de décision du pays qui utilisent ces outils en Télédétection. Il a aussi applaudi la planification de cet effort et espère que celui-ci va maintenant aider la communauté géospatiale à se mobiliser dans le pays pour propager l'utilisation de ces technologies à un public beaucoup plus large.

Mr. Geoff Zeiss d'Autodesk partage les évolutions des idées dans le domaine du business et partage aussi l'idée qu'il y a un bénéfice mutuel à retirer d'une association avec l'Open Source. Il souligne

aussi la nécessité de concerter les efforts autour des Open Standards et Open Content en plus de l'Open Source, ce qui aidera ensuite à créer de nouvelles opportunités de business et comment cela peut aider à construire un écosystème Spatial IT.

Dr. P.S.Roy (Vice Président de l'OSGeo et représentant de l'OSGeo-Inde) a informé le public des efforts de construction de l'OSGeo-Inde et sur la planification à court et moyen-terme de ses projets. Quelques unes des initiatives indiennes prises par les organisations comme la NRSA/DOS, GSI, SACON, NIC, KELTRON, etc. ont aussi été montrées pendant cette présentation. Il a aussi mentionné les défis et les opportunités dans le contexte indien, particulièrement l'établissement de Village Resources Centers and Krishi Vigyan Kendras qui sont mandatés pour fournir une série de services et d'information incluant l'information géospatiale. Il a encouragé tous les participants à l'atelier à participer activement et à contribuer dans le développement d'une vision partagée pour la communauté.

Les représentants de Chine, du Japon et du Vietnam ont donné un aperçu des efforts réalisés dans leurs représentations locales respectives. Prof. Rongguo Chen (Vice Président de l'OSGeo et représentant de l'OSGeo-Chine) a montré l'exemple d'un travail de localisation, GRASS C-API, et la proposition d'installer un GridGIS. Pendant ce temps, Mr. Toru Mori (Vice Président de l'OSGeo et représentant de l'OSGeo-Chine) a donné un bon exemple de comment un modèle open-source peut être une bonne proposition de business. Dans ses conclusions, Prof. Venkatesh Raghavan, Board Member de l'OSGeo, mentionne ce que le futur réserve à la communauté et la longue route pour y parvenir.

L'attrait pour l'atelier de travail a été important, puisque la salle débordait de participants et de public, estimés à plus de 95 personnes.

Stand d'exposition

L'OSGeo-Inde avait installé un stand d'exposition pendant le Map World Forum afin de montrer ses efforts : le lancement de la représentation indienne, la communication d'informations sur l'OSGeo-Inde, la démonstration de ses projets en cours. Ce stand a aussi permis aux membres de la Re-

présentation et la communauté géospatiale d'échanger. La fréquentation a été très bonne et un flux continu de visiteurs de tous les secteurs a été observé. Il vaut la peine de préciser que beaucoup d'entre eux ont loué cet effort et nos objectifs, et ont partagé leur point de vue sur la pertinence de leurs travaux ou de leurs organisations respectives. Quelques uns d'entre eux ont aussi invité la Représentation à organiser des événements comme des démonstrations d'outils et des formations dans tout le pays. Un nombre non négligeable de personnes ont répondu favorablement à cette proposition et sont d'accord pour interagir de façon continue avec l'OSGeo comme utilisateurs, partenaires, développeurs ou promoteurs dans leurs propres organisations respectives.

Rencontre de l'OSGeo-Inde

L'autre événement important, en parallèle du Map World Forum, a été la première rencontre du Conseil exécutif de l'OSGeo-Inde. Le conseil a eu une bonne discussion sur comment poursuivre les efforts et se développer en réponse à ce premier événement majeur. Cette rencontre a réuni des représentants universitaires, gouvernementaux et industriels afin de participer à cet effort et a aussi fourni une représentation pan-indienne à la Représentation. Des représentants de Chine, du Japon, et du Vietnam ont participé en tant qu'observateurs et il y a eu une dis-

cussion sur comment construire des interactions et des synergies entre les différentes représentations.

Toutes les actions de l'OSGeo Inde ont été rendues possibles grâce au généreux support financier de la fondation GNOME et de Osgeo.org. L'OSGeo-Inde voudrait remercier ces organisations pour leur support non seulement en terme logistique, mais aussi pour leur participation aux frais de déplacement de quelques délégués d'Asie. Il est à espérer que ces efforts vont perdurer afin de remplir les objectifs de nos différentes organisations, tout en attendant avec intérêt une interaction continue à l'avenir.

L'OSGEO-Inde cherche maintenant à obtenir un statut légal en tant qu'organisation à but non lucratif en Inde, ce qui aidera à la promotion de notre cause et des missions de l'OSGEO en Inde. La recherche de fonds pour des formations, du développement logiciel et de la localisation a été initiée. La représentation OSGeo-Inde a aussi planifié la construction d'un site Internet pour propager l'adoption et le déploiement des solutions Open Source Géospatiales en Inde. D'autres détails sur les activités de la Représentation OSGeo-Inde peuvent être obtenus en écrivant à l'adresse ci-dessous.

K. S. Rajan

Treasurer, OSGeo-India

International Institute of Information Technology

Gachibowli, Hyderabad 500032, Andhra Pradesh, India

rajan AT iiiiit.ac.in

8^{ème} Rencontre Annuelle des utilisateurs Italiens de GRASS & GFOSS

par Andrea Scianna, traduit par Yves Jacolin

La VIII^{ème} édition des Rencontres des utilisateurs Italiens de GRASS & GFOSS (*Geographic Free Open Source Software*) à pris place les 14, 15 et 16 février à Palerme - Italie, organisée par GISLAB au Département de Représentation de l'Université de Palerme¹⁶

La rencontre a enregistré une forte participation, de toutes les régions d'Italie, de chercheurs, d'étudiants, de professionnels et de collectivités publiques soulignant le fort intérêt envers les *Logiciels Open Source et Libres en Géographie - Geographic Free Open Source Software* en Italie, utilisés dans plusieurs secteurs différents (tels que le suivi environnemental, le calcul, la visualisation et la publication d'informations géospatiales sur des stations de travail ou par le Web, etc.)

Même si, dans le passé, la référence des logiciels *Libres et Open Source* était GRASS, aujourd'hui il y a beaucoup d'autres logiciels SIG disponibles pour le calcul, la visualisation et la publication sur Internet de données géospatiales tels que MapServer¹⁷, QGIS¹⁸, PostGIS¹⁹, ka-Map²⁰, Chameleon²¹, PyWPS²² et d'autres.

La rencontre a été caractérisée par un programme riche en événements.

Le 14 février, avec la collaboration de la Société Italienne de Photogrammétrie et de Topographie, la journée intitulée "Introduction aux SIG" a pris place avec la participation de professionnels et de personnes travaillant dans des sociétés privées ou des administrations publiques qui sont chargées chaque jour de l'étude et la gestion d'informations géospatiales.

Le même jour quelques cours sur Linux et les logiciels SIG (GRASS et QGIS) et sur le logiciel de Web-

mapping MapServer étaient proposés.

Les sujets de la partie principale de la rencontre, tenue les 15 et 16 février, étaient les applications de logiciels Open Source et Libres en Géographie, telles que l'analyse spatiale, les techniques SIG pour les analyses environnementales, les applications en hydrogéologie, les applications avec le patrimoine, l'intéropérabilité et la sécurité dans le Webmapping.

Pendant la rencontre d'autres sujets sur la standardisation des données spatiales ont été discutés ainsi que les problématiques d'échange de données pour les administrations publiques, incluant une intervention du gouvernement Italien.

Au delà de l'intérêt de l'administration publique pour le phénomène "Open Source", il y a en Italie une forte communauté de scientifiques, de développeurs et d'utilisateurs qui soutient le développement de GFOSS. Certains d'entre eux ont fondé l'*Associazione Italiana per l'Informazione Geografica Libera / GFOSS.IT* (L'association italienne pour la géoinformation libre - *GFOSS.it*) pendant la rencontre.

Les principaux objectifs de cette nouvelle association sont de soutenir le développement, la diffusion et les droits des logiciels Libres et Open Source, de promouvoir les standards ouverts pour la géoinformation et l'accès libre aux données spatiales collecté par l'État, de promouvoir les relations entre les personnes qui constituent la Communauté Libre et Open Source en Géomatique. *GFOSS.it* a l'intention de devenir la représentation locale italienne de l'OS-Geo.

Les comptes rendus des rencontres sont publiés sur les pages web des rencontres²³.

Dans un futur proche, les comptes rendus seront également publiés sur les pages web de l'école Polytechnique de Milan - Polo di Como²⁴ où les présentations de toutes les rencontres des utilisateurs italiens de GRASS ont été publiées, à partir de l'année 2000.

¹⁶GISLAB au Département de Représentation de l'Université de Palerme : <http://gislab.dirap.unipa.it/>

¹⁷MapServer : <http://mapserver.gis.umn.edu/>

¹⁸QGIS : <http://qgis.org/>

¹⁹PostGIS : <http://postgis.refractory.net/> et <http://www.postgis.fr/> en France

²⁰ka-Map : <http://ka-map.maptools.org/>

²¹Chameleon : <http://chameleon.maptools.org/>

²²PyWPS : <http://pywps.wald.intevation.org/index.psp>

²³Page web des rencontres des utilisateurs de GRASS/GFOSS : http://gislab.dirap.unipa.it/grass_meeting/index.htm

²⁴Comptes rendus publiés à l'école Polytechnique de Milan - Polo di Como : <http://geomatica.como.polimi.it/workbooks/>

Ph.D. Ing. Andrea Scianna
Management board, <http://GFOSS.it>
CNR D.C.S.P.I. - Dipartimento di Rappresentazione
Università degli Studi di Palermo

Italy
<http://gislab.dirap.unipa.it>
scianna AT dirap.unipa.it

Zoom sur un projet

QLandkarte

par Oliver Eichler, traduit par Sidonie Christophe

Pour gérer et manipuler vos données géographiques personnelles, il existe beaucoup de bons outils open source comme GPSTabel²⁵ et GPSTMan²⁶, mais aucun n'utilise les cartes vecteurs propriétaires de Garmin. QLandkarte remplit cette fonction et vous permet de visualiser ces cartes avec vos données géographiques depuis votre récepteur Garmin GPS. QLandKarte utilise la librairie Qt v4.2 de la société Troll Tech pour la programmation du GUI et la librairie Proj4 pour toutes les projections 2D.

²⁵<http://www.gpsbabel.org>

²⁶<http://www.ncc.up.pt/gpsman/>

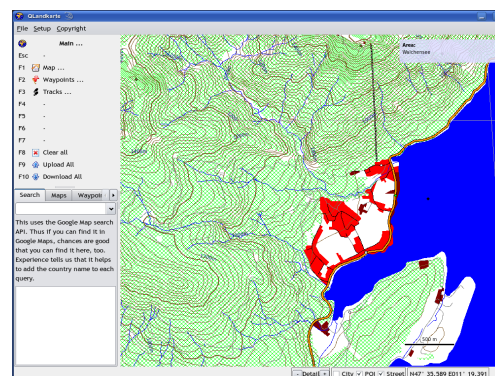


FIG. 1 – QLandkarte après démarrage

Fonctionnalités

- Modèles supportés :
 - GPSTMap 60CS, 60Cx, 60CSx
 - eTrex Legend C
- Import de collections de cartes (nécessite des fichiers *.tdb, carte de base et "basemap and map tiles")
- Chargement de simple "map tiles" (Format *.img)

- Affichage de la carte et information sur l'objet proche du curseur
- Recherche de lieux via Google Maps
- Recherche de géocaches via www.opencaching.de
- Cartes :
 - Charger des cartes dans l'appareil
 - Télécharger une sélection de carte depuis l'appareil
 - Charger / Sauvegarder une sélection de carte depuis / vers des fichiers GPX
- Points de cheminement :
 - Charger / télécharger des points de cheminement
 - Créer / déplacer / éditer des points de cheminement
 - Charger / sauvegarder des points de cheminement depuis / vers des fichiers GPX
- Tracés :
 - Télécharger des tracés
 - Visualiser l'information sur un tracé et le profil d'élévation
 - Fusionner / découper des tracés
 - Purger / supprimer les points de tracé
 - Charger / sauvegarder des tracés depuis / vers des fichiers GPX
- Imprimer des cartes au format PDF ou papier

Aperçu

La Figure 1 présente QLandkarte juste après le démarrage de l'application. Le mode de fonctionnement et le type de curseur peuvent être modifiés au clavier ou à la souris dans le menu en haut à gauche. Les données de l'utilisateur seront listées dans les onglets en-dessous.

Il y a trois modes principaux : cartes, points de cheminement et tracés. Les touches utilisées pour chacun de ces modes sont affichées dans la Figure 2. Les touches F1 à F4 correspondent à des fonctions qui permettent de naviguer dans la carte. Un zoom dynamique peut être réalisé en utilisant la molette de la souris.

Chaque mode principal a son propre mode de dialogue dans l'onglet listant les données disponibles de l'utilisateur. Les données sélectionnées peuvent être supprimées avec la touche DEL en règle générale.

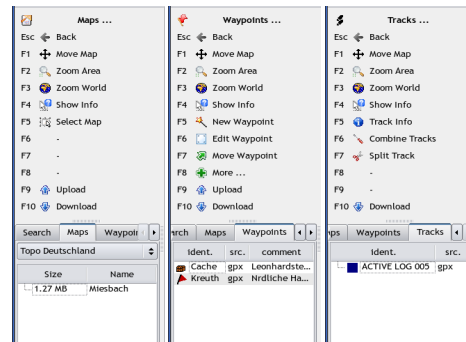


FIG. 2 – Modes de fonctionnement

Mode Carte

Dans le 'Mode Carte' la collection courante de cartes peut être sélectionnée dans la liste de choix dans le haut de l'onglet. Pour charger les cartes dans l'appareil elles doivent être sélectionnées en cliquant dessus dans l'espace de visualisation de la carte. Les tuiles sélectionnées depuis l'ensemble des collections sont listées dans l'arborescence.

Mode Points de cheminement

La Figure 3 présente la boîte de dialogue concernant les propriétés du point de cheminement. À côté des champs habituels comme le symbole, le nom et la position, une distance de proximité et un lien vers une page Web peuvent être renseignés. Le lien est actif et ouvrira un navigateur.

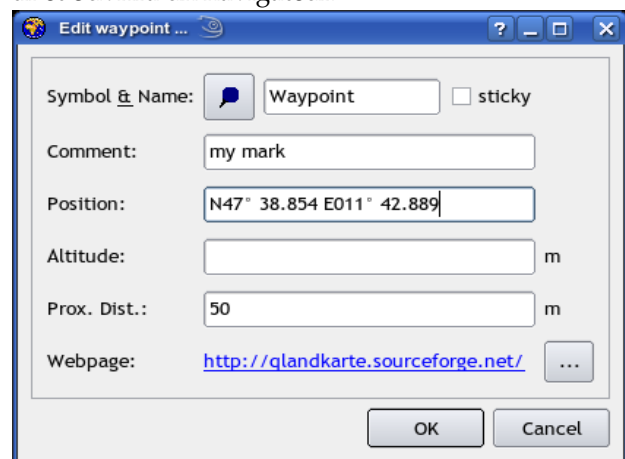


FIG. 3 – Propriétés du points de cheminement

Un double clic dans la liste des points de chemi-

nement centrera la carte autour du point de cheminement sélectionné. Le mode 'Déplacer le point de cheminement' est assez complexe et nécessite une attention particulière. Un clic gauche sur un point de cheminement va le coller au curseur. Une bande entre la position initiale et la position courante mettra en évidence la distance entre les azimuts avant et arrière. Regardez la Figure 4. A ce moment, il est possible de changer de mode vers les fonctions de navigation, permettant à l'utilisateur de déplacer sa carte jusqu'à la zone désirée. Une fois revenu au mode 'Déplacer le point de cheminement', celui-ci peut être supprimé par un second clic gauche ou l'opération se termine par un clic droit.

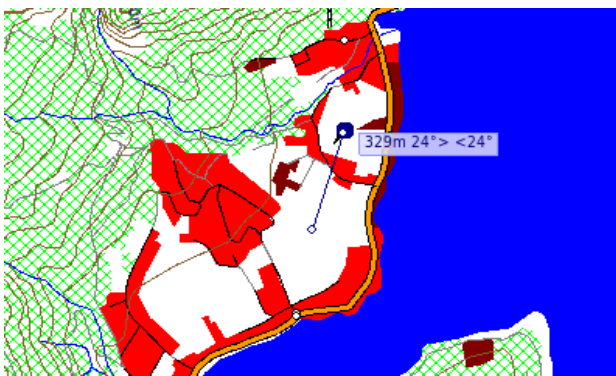


FIG. 4 – Déplacer un Waypoint

Mode Tracé

Dans la Figure 5 QLandkarte fonctionne en 'Mode Tracé'. La visualisation des informations sur le tracé est activée. Un double clic dans la liste des tracés entrainera un zoom dans la vue 'carte' sur le tracé sélectionné.

Le tracé est affiché sous forme d'un chapelet de points dans la carte, sous forme d'un profil d'élévation et sous forme d'une table. Les trois vues sont liées ensemble, donc si un point du tracé est sélectionné dans l'une d'entre elles, il sera mis en surbrillance dans les autres aussi. La visibilité des points du tracé peut être modifiée avec la touche DEL dans la vue 'table'. Cela créera un ombrage sur les points dans la table et les empêchera d'être ajoutés au tracé.

Une sélection multiple de tracés peut être combinée en une nouvelle sélection. Le tracé sélectionné peut être découpé à chaque point. QLandkarte ne détruira pas les données originales. Cela créera deux nouveaux tracés comme copies de l'original.

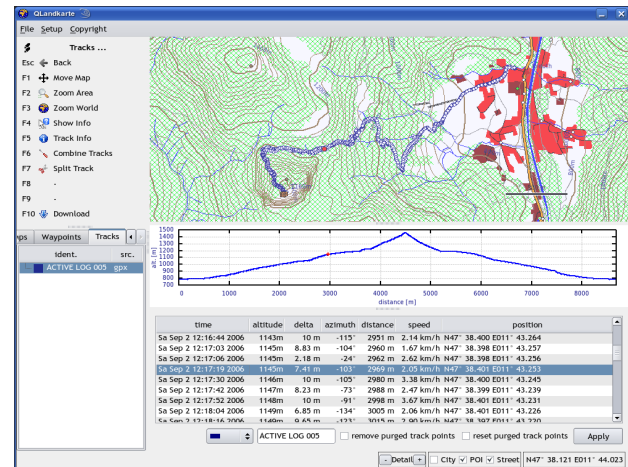


FIG. 5 – Vue Info du Tracé

Développement

Le projet a atteint une étape satisfaisant les envies initiales de l'auteur. Cela ne veut pas nécessairement dire que l'application est mature et complète.

Il manque encore à QLandkarte la gestion d'un certain nombre de modèles Garmin. Cela fournit déjà un plug-in pour les modèles pour permettre un développement décentralisé sur plusieurs développeurs. Développer un pilote est plus simple qu'il n'y paraît. Il y a un petit tutorial sur la page d'accueil de QLandkarte.

Une tâche restante est de porter QLandkarte sur d'autres plateformes comme Windows ou Mac. Cela ne devrait pas être un problème pour les plateformes little-endian. Cependant cela fera trop de travail pour les plateformes big-endian vu que le format IMG est encodé dans un format little-endian et que le décodeur adresse directement les données pour préserver les performances. C'est pourquoi un portage big-endian devrait compenser les pénalités de performance sur la permutation des octets en optimisant le code du décodeur.

Les développeurs intéressés sont les bienvenus pour discuter de tout ça et envoyer leurs patches sur la liste de diffusion utilisateur de QLandkarte.²⁷

Dipl.-Ing. (FH) Oliver Eichler

L'auteur a fait des études en Ingénierie Electronique à Regensburg et travaille depuis 1997

comme freelance principalement "focused on digital subscriber lines".

<http://qlandkarte.sourceforge.net>
[oliver.eichler AT gmxd.de](mailto:oliver.eichler@gmxd.de)

²⁷http://sourceforge.net/mail/?group_id=183501

Une courte introduction à Mapbender

"...plier les cartes à vos exigences"

par Astrid Emde et Marc Jansen, traduit par Marie Silvestre

Introduction

Cet article fournit un aperçu de la portée, des particularités et des fonctionnalités du client web mapping Mapbender.²⁸ Une courte introduction des fonctionnalités de Mapbender sera suivie de l'historique du projet. Ensuite, la fonctionnalité principale sera décrite à la fois du point de vue de sa gestion et de son orchestration ainsi que d'un point de vue de sa visualisation et de sa présentation comme serveur de carte et de Web Feature Service (WFS). La dernière section résume les développements futurs prévus pour l'année à venir. Deux environnements de production seront brièvement présentés et des liens vers plus d'exemples donneront un aperçu de l'utilisation de Mapbender.



Qu'est-ce que Mapbender ?

La suite cliente Mapbender est un framework pour gérer, orchestrer et afficher des services web pour les données géographiques conformes aux standards OGC WMS et WFS. Mapbender n'est pas un serveur cartographique comme MapServer de l'Université du Minnesota²⁹ mais il est utilisé pour afficher, superposer et modifier des cartes OGC produites par MapServer ou n'importe quel autre serveur compatible OGC.

Dans Mapbender, les cartes provenant de différentes sources sont empilées les unes sur les autres et peuvent alors être diffusées à travers le monde via internet. Comme le site de l'OSGeo le précise, *Mapbender est la face visible de l'iceberg, une méta-couche d'application donnant accès à l'Open Source Geospatial SDI (Spatial Data Infrastructure)*.³⁰

L'application fournit une interface pour afficher, naviguer et interroger les services OWS de l'OGC

c'est-à-dire les WMS (Web Map Service), WFS (Web Feature Service), WFS-T (WFS transactionnel) et WMC (Web Map Context). Le framework Mapbender fournit en plus des interfaces pour les utilisateurs et l'administration des groupes. La fonctionnalité permettant la gestion de l'accès aux cartes et aux données fournis par les Open Web Services de l'OGC est également incluse. Le modèle de base de données de Mapbender a été pensé pour utiliser les potentialités multi-client, à savoir, les utilisateurs, les groupes, les interfaces et les services. Les trois composants de base suivant font Mapbender :

1. Les utilisateurs et les groupes
2. Les interfaces utilisateurs (interface homme machine, IHM)
3. Les services cartographiques (OGC Open Web Services)

Actuellement, les développements ont été portés sur la fonctionnalité WFS-T, le catalogue et le service de positionnement.

Court historique de Mapbender

A la fin du dernier millénaire, l'entreprise allemande CCGIS créa la première interface web basée sur PHP pour afficher les services cartographiques de l'OGC (ils n'étaient alors qu'au stade de l'avant-projet WMT, Wep Mapping Testbed). Cette application se mua en produit et devint finalement la base de l'actuel Mapbender. En novembre 2001, la première version de la 'Suite Client CCGIS' fut publiée, mettant en oeuvre le premier standard WMS de l'OGC.

En 2002, la première installation de grande envergure de cette application propriétaire a été menée dans un environnement de production de 400 utilisateurs. L'année 2003 vu un certain nombre de changements importants dans le projet :

- la suite a été complètement redéfinie, tout en gardant deux objectifs à l'esprit :
 1. Réduire la complexité des relations entre les services, les interfaces, les éléments de l'interface et la gestion des utilisateurs.
 2. Fournir des interfaces d'administration et des modèles de données permettant aux utilisateurs de gérer tous les aspects de l'application via des interfaces web.

²⁸Voir la section Annonces des développeurs en page 64 pour plus d'informations sur Mapbender

²⁹MapServer de l'Université du Minnesota : <http://mapserver.gis.umn.edu/>

³⁰Tiré de <http://www.osgeo.org/mapbender>

Les interfaces d'administration sont elles-même gérées de manière réursive, réduisant grandement la complexité de l'application.

- Utilisateurs, développeurs et possesseurs de la licence de la suite ont été sollicités pour donner leur avis sur un changement majeur de philosophie, CCGIS avait en effet prévu de changer pour une licence de logiciel libre.
- Les possesseurs de licence ont été ravis que cela se fasse et l'application a alors été distribuée sous la licence de logiciel libre GNU GPL.
- Le nouveau nom du projet a été annoncé : la 'Suite Client CCGIS' devait désormais s'appeler 'Mapbender'.

En octobre 2003, plus de 500 téléchargements ont été comptabilisés dans les heures qui ont suivi la sortie de la mise à jour. Les utilisateurs ont accepté la nouvelle licence et ont commencé à adopter la nouvelle philosophie de développement. Une année plus tard, en 2004, l'application était utilisée en production par cinq administrations publiques (comtés et villes) membres de la Spatial Data Infrastructure (SDI) de la Rhénanie du Nord Westphalie.³¹

En 2004, une démo officielle de serveur fut mise en place et hébergée sur les serveurs de CCGIS. Elle est actuellement en transfert vers un nouveau serveur³² qui sera alors l'unique serveur officiel. De plus, en 2004, le site internet a été fusionné avec la documentation en un wiki, plateforme technique commune. Plus tard, le wiki fut remplacé par un MediaWiki qui fournissait plus de fonctionnalités et nécessitait peu de gestion des droits des utilisateurs pour résoudre le problème des spams.

Au mois d'avril 2005 s'est tenue la première conférence utilisateur à Bonn, Allemagne et attira 85 participants d'Allemagne, Autriche, Suisse et des Pays-Bas. Ce fut également la dernière conférence exclusivement pour Mapbender car depuis elle a été combinée avec d'autres conférences existantes comme le FOSSGIS³³ ou le FOSS4G 2006.³⁴

Au début de l'année 2006, Mapbender devint l'un des neuf projets initiaux de l'Open Source Geospatial Foundation (OSGeo). Durant les 6 mois qui ont suivi, Mapbender a traversé la phase d'incubation de l'OSGeo avec l'examen rigoureux du Comité d'Incubation et l'aide du mentor du projet Paul Spencer. Pen-

dant ce processus, plusieurs problèmes de licences ont été résolus, le code a été nettoyé, les ambiguïtés au niveau du copyright ont été résolues, le code source ainsi que les listes de diffusion et une partie de l'infrastructure ont été déplacés du CVS de SourceForge vers le SVN de CollabNet. Le 19 juillet 2006, Mapbender a été le premier projet à passer le processus d'incubation.

Un dernier changement a eu lieu fin 2006, lorsque l'infrastructure du projet a été déplacée sur les serveurs de l'OSGeo, une infrastructure plus libre et ouverte. Finalement (presque) toutes les composantes de l'environnement de développement et de l'infrastructure de communication sont gérées sous un même toit, la seule composante qui tourne encore sur un serveur séparé reste le Wiki.

La version stable actuelle de l'application est la 2.4.1, elle est sortie le 12 mars 2007.



FIG. 1 – Une démo de Mapbender. Notez la barre d'outils (en haut), les menus (à gauche, visible ici : option d'export en PDF) et le formulaire email pour envoyer un lien vers la carte affichée (dans le coin supérieur droit). Copyright de l'image : Kober-KiÄçmmerly+Frey Media AG, Cologne, Germany. Utilisée avec leur permission.

Fonctionnalité

Mapbender fournit la possibilité de gérer et d'afficher des données spatiales cartographiées et disponibles au format OGC standard via à une application serveur. Vous trouverez des exemples de différents

³¹Infrastructure de Données Spatiales de la Rhénanie du Nord Westphalie : <http://www.gdi-nrw.org>

³²Nouvelle démo de Mapbender : <http://mapbender.telascience.org>

³³FOSSGIS : <http://www.fossgis.de>

³⁴FOSS4G 2006 : <http://www.foss4g2006.org>

³⁵Galerie Mapbender : <http://www.mapbender.org/index.php/Gallery>

³⁶Démonstration officielle en ligne : <http://www.mapbender.org/index.php/Demo>

scénarios dans la galerie Mapbender³⁵ ou allez sur la démonstration officielle en ligne de l'application.³⁶ L'interface standard de la démonstration et un géoportail complet sont actuellement en déplacement vers les serveurs de l'OSGeo hébergés par Telascience.³⁷

Gérer les Services Web Mapping

Mapbender est utilisé pour orchestrer les services et les formats de l'OGC. L'application fournit des interfaces d'administration pour charger et mettre en cache des documents de Capabilities pour les WMS, WFS et WFS transactionnels. Une fois la description de configuration chargée, elle peut être combinée via l'interface cartographique avec des cartes ou des objets provenant d'autres services.

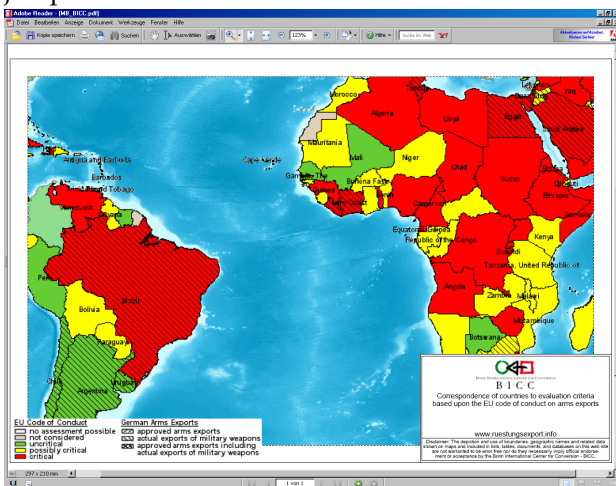


FIG. 2 – Un PDF au format A4 paysage, généré à la volée par Mapbender. Il montre les exportations réelles d'armes d'Allemagne (hachures) et le "Code du Conduite de l'UE pour les Droits de l'Homme" (en couleur) pour une partie de l'Amérique du Sud et de l'Afrique. Voir <http://www.ruestungsexport.info>.

Comme beaucoup de services web manquent d'informations sur les métadonnées, un éditeur qui permet aux utilisateurs d'améliorer les informations existantes et de définir pour le service et pour chaque couche une catégorie telles qu'elles sont spécifiées dans le Dublin Core a été ajouté à Mapbender.³⁸ Ces informations améliorées peuvent être recherchées via le module de recherche de métadonnées et affichées en suivant un lien attaché au TreeGDE (Geo

Data Explorer). le TreeGDE a une structuration en répertoire, il est utilisé pour gérer les couches et les services dans l'interface d'affichage.

Gérer les utilisateurs, les groupes et les services

Une gestion des utilisateurs complète et entièrement côté client permet de sélectionner et de configurer des interfaces avec des collections de WMS et de WFS pour les utilisateurs et les groupes. Le module Proxi OWS de Mapbender peut être activé pour être utilisé en tant que facade par dessus les services. Le ticketing a été implémenté, utilisant la fonctionnalité de redirection d'Apache fournissant à la fois une interface conforme aux standard OGC mais également qui autorise les accès restreints. Ceci peut aussi être utilisé pour implanter le chiffage du protocole qui, dans certains cas, nécessite l'utilisation d'un proxy du côté client tel que celui de l'application Int-Proxy (<http://inteproxy.wald.intevation.org/>). Toutes les requêtes sont alors dirigées à travers le module de gestion des utilisateurs et les permissions peuvent être refusées pour des requêtes sur des services qu'un utilisateur n'a pas le droit d'utiliser.

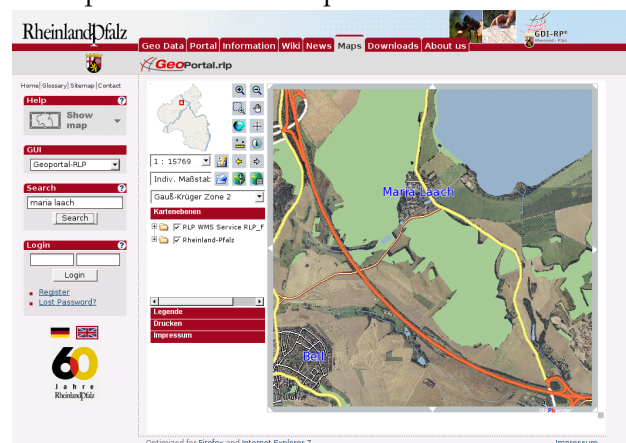


FIG. 3 – Le Géoportail de Rhineland Palatinate (<http://www.geoportal.rlp.de/>) donne accès à plus de 70 services cartographiques WMS avec plusieurs centaines de couches gérées par l'administration publique, les institutions non-gouvernementales et les entreprises privées. Il permet aux gestionnaires des services de charger les images de leurs propres services, éditer les métadonnées correspondantes et offre la possibilité aux utilisateurs et aux groupes d'y accéder.

³⁷Nouveau serveur de démonstration Mapbender : <http://mapbender.telascience.org>

³⁸Dublin Core <http://dublincore.org/>

Visualiser, Naviguer et interroger les services

En plus des fonctionnalités standards de zoom, de déplacement panoramique ou de requête, il est possible de numériser (créer, éditer, supprimer) des entités et de les stocker en utilisant les interfaces WFS-T. Le WFS est également utilisé pour rechercher, trouver et naviguer vers les entités (adresses, parcelles, véhicule se déplaçant avec un GPS transmettant ses coordonnées en ligne) et peut effectuer des opérations.

Des documents OGC Web Map Context (WMC) peuvent être créés, stockés puis utilisés pour retrouver un état défini de l'interface web.

Toute fonctionnalité dans Mapbender est implémentée en tant que module qui peut être activé dans les interfaces utilisateur grâce à l'interface d'administration web.

La liste exhaustive des fonctionnalités SIG implantées se trouve en ligne.³⁹

Développements futurs

Mapbender est étendu et amélioré de manière dynamique grâce aux contributions de la communauté. Les prochaines versions proposeront des fonctionnalités additionnelles et des corrections de bug :

- Amélioration de l'enregistrement de l'activité des utilisateurs,
- Module d'impression avec une configuration plus ergonomique,
- Impression (et export PDF) supportant les objets géographiques temporairement créés,
- Interface de catalogage des services se connectant à GeoNetwork opensource incluse,
- Amélioration de l'édition de l'arborescence,
- Calcul d'itinéraires en ligne,
- Service de positionnement modulable et à différents niveaux,

- Et beaucoup d'autres choses auxquelles nous n'avons même pas encore rêvé.

Plus loin sur la route, un éditeur intégré de SLD (Styled Layer Descriptor) sera nécessaire pour aider à personnaliser les sorties cartographiques.

Toute contribution des utilisateurs est très appréciée, inscrivez-vous aux listes de diffusion⁴⁰ et venez partager vos idées et vos réflexions.

Références and liens internet

- Wiki du projet et page d'accueil (1ère source d'information utilisée dans cet article) : http://www.mapbender.org/index.php/Main_Page
- Documentation, premiers pas et aide générale sur différents sujets : http://www.mapbender.org/index.php/User_Documentation
- Page du projet sur le site internet de l'Open Source Geospatial Foundation : <http://www.osgeo.org/mapbender>
- Copie d'écran, liens & exemples de différentes installations de Mapbender : http://www.mapbender.org/index.php/Mapbender_Gallery
- Serveur de démonstration en ligne : <http://www.mapbender.org/index.php/Demo>
- Listes de diffusion, dépôt SVN du code et IRC : http://www.mapbender.org/index.php/Mapbender_Mailing_Lists
- Le logo est disponible en ligne ici : http://www.mapbender.org/images/2/21/Mapbender_Geoportal_Framework.png

Astrid Emde
WhereGroup GmbH & Co.KG
<http://www.wherogroup.com/astrid.emde> AT wherogroup.com

Marc Jansen
terrestris GbR
<http://www.terrestris.de/jansen> AT terrestris.de

³⁹Liste des fonctionnalités SIG : <http://www.mapbender.org/index.php/Category:Modules>

⁴⁰Listes de diffusion Mapbender : http://www.mapbender.org/index.php/Mapbender_Mailing_Lists

deegree

Construire des briques pour une Infrastructure de Données Spatiales

par Markus Müller, traduit par Yves Jacolin

Introduction

deegree⁴¹ est un framework libre / Open Source basé sur Java pour le développement d'Infrastructures de Données Spatiales (SDI). Il contient aussi bien les services nécessaires pour un SDI (deegree Web Service) que pour un composant d'un portail (deegree iGeoPortal), ou bien un mécanisme pour gérer les problèmes de sécurité et de contrôle d'accès (deegree iGeoSecurity) et le stockage / visualisation de données 3D (deegree iGeo3D). Conceptuellement et vis à vis de l'interface, deegree est basé sur les standards de l'Open Geospatial Consortium (OGC) et ISO / TC 211. Au moment de l'écriture de l'article il s'agit de la mise en oeuvre la plus complète de ces standards dans un framework Open Source. Le framework est basé sur des composants d'un haut niveau permettant la création flexible de solutions pour une large variété de cas d'utilisation. deegree est le développement de référence officiel de l'OGC pour les standards de Web Map Service et de Web Coverage Service. Il est publié sous licence GNU Lesser General Public (LGPL).

Historique

deegree est géré en coopération entre la société privée lat/lon et le groupe de travail SIG de l'Université de Bonn. Le moteur de deegree revient à un projet de l'Université de Bonn nommé EXSE (Serveur EXperimentale GIS) fonctionnant depuis 1997 à 2003. Le but du projet était une analyse basée sur l'expérimentation de fusion de fonctionnalités SIG et de technologie Internet. Plusieurs outils et modules de logiciels ont été développés incluant un premier développement du Simple Feature de l'OGC pour les spécifications CORBA comme une API Java Open Source (appelé sf4j - Simple Feature pour Java).

Au printemps 2001, le projet sf4j, les outils existants et les modules ont été réarrangés en un nouveau projet appelé JaGo (Java Framework for Geospatial Solutions) dont le but est de réaliser un développement Open Source des spécifications des Web

Services de l'OGC. Le premier service développé a été la spécification WMS 1.0.0 de l'OGC à l'été 2001. Jusqu'à la fin de cette année-là WFS 1.0.0 et WCS 0.7 ont suivi. Comme les noms de domaines principaux (.de, .org, .net) n'étaient pas disponibles il a été décidé fin 2001 de renommer le projet en deegree.

La prochaine étape importante a été de publier deegree 1.0 à la fin de l'été 2002. Certains changements dans l'architecture offrait une meilleure prise en charge de l'OWS disponible (Web Service de l'OGC). Le développement d'un moteur et d'interfaces de service multi-threading a été ajouté afin d'activer des OWS distants permettant une grande extensibilité des applications.

Les versions mineures suivantes proposent de nouvelles fonctions comme un WFS transactionnel, un Gazetter (WFS-G) et la gestion de sources de données additionnelles. Depuis ce temps là, le WMS de deegree gère les Descriptions des Styles des Couches (Styled Layer Descriptor - SLD) et un Service de Catalogue. Des mécanismes de sécurité ont été ajoutés au framework. Une étape important pour augmenter la publicité du projet a été de se déplacer vers sourceforge. Plusieurs développeurs ont commencé à regarder le code de deegree et ont ajouté le leur au projet.

Un travail additionnel dans le développement du framework a été démarré en 2003. Il a pour but d'offrir des composants pour permettre les développeurs de créer des clients web basés sur deegree. Ce nouveau framework client est nommé iGeoPortal et gère le standard de l'OGC Web Map Context 1.0.0.

Une des étapes les plus importantes dans le développement de deegree a été la participation dans le projet CITE1 de l'OGC à l'été 2003, qui a transformé deegree en un développement de référence officiel de l'OGC pour la spécification du WMS 1.1.1. La participation de lat/lon et deegree dans CITE a été un tel succès que lat/lon a été sélectionné par l'OGC plus tard pour développer les développements de référence du WCS 1.0.0 et WMS 1.3 avec deegree dans le contexte des initiatives OWS-2 et OWS-4 de l'OGC.

En 2005, deegree2 a été lancé, représentant encore un grand pas en avant dans le développement du framework. Les pierres angulaires de deegree2 sont un mécanisme basé sur les modèles pour le WFS de deegree, permettant un développement flexible de différents modèles de données en utilisant des schémas d'application GML. De plus, le développement du

⁴¹Site Internet de deegree : <http://www.deegree.org>

framework client basé sur des portlets appelé deegree iGeoPortal - édition de portlet, gestion des structures de données 3D et un développement du Web Processing Service (WPS) sont inclus dans deegree2.

Composants

deegree comprend cinq groupes de composants qui sont brièvement décrits dans la section suivante :

deegree Web Services

Ceux-ci sont les composants de base de n'importe quel SDI. Ils peuvent par exemple être utilisés pour afficher les cartes et les données 3D, accéder aux données vecteur et raster, gérer et récupérer les méta-données, et faire du traitement de données spatiales. La liste des Services Web de l'OGC développés inclut WMS, WFS, WCS, CS-W, Gazetteer, SOS, WPS and WTS/WPVS. De plus, deux services de sécurité (WAS/WSS), un service d'impression (WMPS) et un moniteur de service (owsWatch) sont disponibles.

deegree iGeoPortal

C'est le framework de portail du projet deegree. iGeoPortal a une structure modulaire et est capable d'afficher des cartes, gère la recherche en utilisant des identifiants géographiques, la recherche de jeu de données en utilisant les méta-données, permet l'accès contrôlé vers les Services Web de l'OGC et affiche des données spatiales 3D.

Différents types de portails SDI ont été créés en utilisant le standard de deegree iGeoPortal et l'édition de portlet et les services web géographiques correspondants. Bien que l'édition standard de deegree utilise la technologie DHTML, l'édition de portlet est basée sur JSR-168, le standard de portlet. Les deux éditions utilisent la technologie AJAX pour quelques modules spécifiques. Ces portails incluent les standards WebGIS et des applications spécialisées telles que celles utilisées par les municipalités, les agences environnement, d'enquête et d'autres.

deegree iGeoPortal consiste lui-même en différents modules pour séparer, mais aussi combiner, les fonctionnalités. La liste des modules inclus : carte, téléchargement, gazetteer, catalogue, sécurité et 3D.

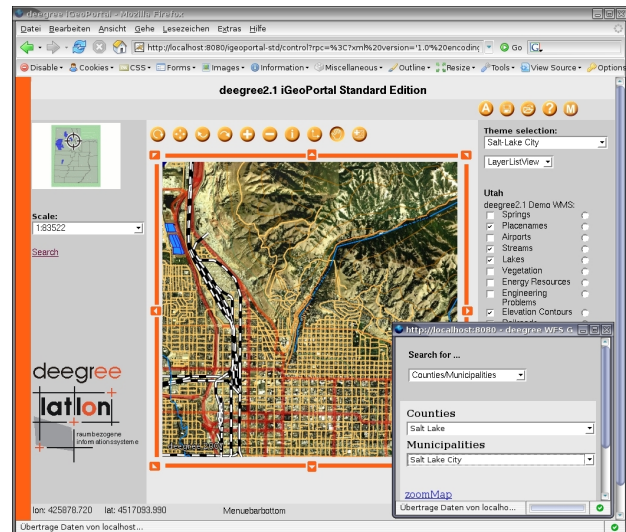


FIG. 1 – Édition standard de deegree iGeoPortal.

deegree iGeoSecurity

Les contrôles d'accès pour les données spatiales et les services sont un problème important pour une large variété d'applications. deegree iGeoSecurity peut être utilisé pour définir des mécanismes d'accès en utilisant des mécanismes d'authentification et de d'autorisation, des connections sécurisées et des filtres pour les données spatiales. Une base de données pour gérer les utilisateurs, les groupes d'utilisateurs, les rôles et des droits appelés deegree U3R est le composant central des composants de sécurité.

deegree iGeo3D

deegree peut être utilisé pour stocker des données spatiales 3D tels que les terrains numériques et les modèles d'immeubles dans un système basé sur des fichiers et des bases de données relationnelles. En utilisant différents services web deegree, cette donnée peut être requêtée et affichée. Ces systèmes sont souvent utilisés par les agences de surveillance et les municipalités pour stocker leur données 3D pour des utilisations commerciales ou pour exécuter des règlements européens tels que instructions européennes sur le bruit. deegree iGeo3D utilise les standards de l'OGC WFS, WCS, WTS et CityGML.

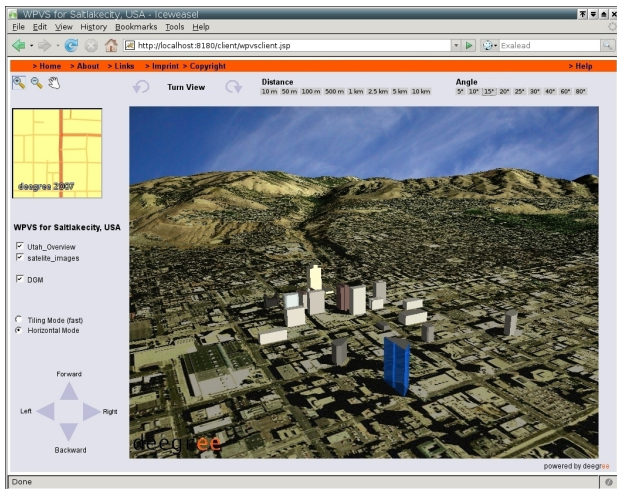


FIG. 2 – deegree iGeo3D.

deeJUMP

Un autre type d'application de deegree est le SIG de bureau classique. En utilisant les composants de deegree, le SIG de bureau Open Source JUMP (Java Unified Mapping Platform) a été amélioré pour gérer le WMS et le WFS. Un certain nombre de modifications et d'extensions additionnelles a été également développé.

Architecture d'ensemble

Les différents composants de deegree peuvent être combinés ensemble ou avec d'autres logiciels conformes aux standards. Une architecture exemplaire d'un système basé sur deegree est affichée à la figure 3.

iGeoPortal avec ses différents modules agit comme un client basé sur Internet pour les différents Web Services. Alternativement, deeJUMP peut être utilisé pour accéder aux services WMS et WFS. Si nécessaire, la gestion de la sécurité peut être développée en utilisant iGeoSecurity. deegree peut accéder à différentes sources de données, par exemple les bases de données PostgreSQL/PostGIS ou Oracle, shapefiles, toutes sortes de bases de données relationnelles

en utilisant une extension spatiale spécifique à deegree et différentes sortes de formats d'images tels que PNG, GIF, JPEG ou (Geo)TIFF.

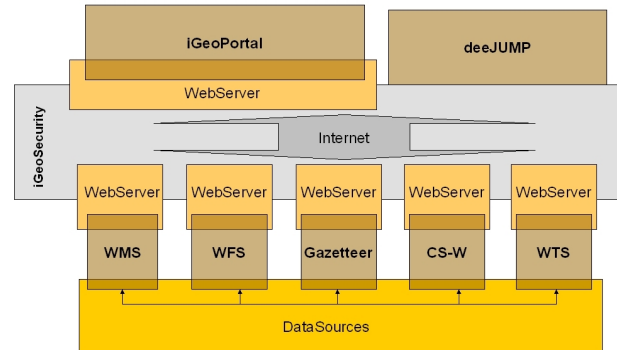


FIG. 3 – Architecture de deegree.

Statut actuel

En juin 2007 la journée de deegree a eu lieu à Bonn en Allemagne. D'ici là, la version 2.1 de deegree sera publiée. La version de démonstration sera extrêmement simple à installer puisqu'elle sera distribuée sous une archive WAR et peut être déployée dans la minute si un moteur Tomcat Servlet est disponible. Les versions de démonstration incluent un Web map Service, des Web Feature Services (avec gazetteer), un Web Coverage Service, un Catalogue Service-Web (pour les méta-données ISO), un Web Terrain Service (Web Perspective View Service), iGeoPortal et deeJUMP. Toutes les versions de démonstration sont fournies avec des sources de données préconfigurées et une documentation détaillée. Allez visiter <http://www.deegree.org> pour connaître l'actualité de la journée de deegree en juin 2007 et les versions à venir.

Dr. Markus Müller
 lat/lon GmbH
 Aennchenstr 19
 53177 Bonn
 Germany
mueller AT lat-lon.de
<http://www.lat-lon.de>

Introduction à OpenModeller

Un ensemble de bibliothèques pour la modélisation des niches fondamentales

par Tim Sutton, Renato de Giovanni et Marinez Ferreira de Siqueira, traduit par Ludovic Granjon

Introduction

En 1957, Hutchinson (4) formalisait le principe de niche fondamentale. En substance, il proposait que si toutes les conditions environnementales qui permettaient à une espèce d'exister indéfiniment, pouvaient être modélisées en un hypercube multidimensionnel, le résultat de cet hypercube pourrait alors être considéré comme la niche écologique fondamentale de l'organisme. Naturellement, connaître toutes les conditions écologiques pour toutes espèces données est une tâche monumentale, limitée d'abord par le manque de richesse de données qui pouvait être conduit lors d'une analyse. Par extrapolation, en associant la localisation de l'occurrence des espèces et un jeu de données géographiques de couches rasters représentant les conditions environnementales comme la pluviométrie, la température, la radiation solaire, etc, une approche corrélative peut permettre de décrire la niche écologique d'un organisme (10). Utilisant cette technique, une couche raster peut être produite, avec les surfaces qui ressemblent le plus aux conditions environnementales connues par les sites d'occurrence pour l'organisme. Sans doute, cette approche recueillerait d'abord un plus grand intérêt avec la publication d'une 'analyse biogéographique des serpents australiens Elapid' (5), où les auteurs ont produit des cartes de prédiction de la distribution des serpents Elapid, basé sur de simples modèles de préférence climatique. À mesure que le monde s'engage de plus en plus dans les discussions sur les impacts potentiels du changement climatique global, la capacité de prévoir l'impact de ces changements sur la distribution des organismes sera encore plus d'actualité. La modélisation des niches fondamentales a été utilisée pour prédire la disparition d'espèces face aux futurs changements du climat (12; 2). Cela a été également utilisé dans les modèles d'invasion d'espèce, de propagation de vecteur

des maladies (6) et dans les modèles phylogénétique pour produire une carte de distribution d'hypothétiques ancêtres (13; 14).

Nombre d'applications existent pour réaliser l'analyse des niches fondamentales. Les choix les plus populaires sont Bioclim (5), DesktopGARP (9), MaxEnt (7) etc. Elith *et al.* (3) fournissent une excellente vue d'ensemble et comparatif de la plus part des applications communément utilisées. Chacune de ces applications sont usuellement créées pour adresser un algorithme spécifique de modélisation à des niches fondamentales. Typiquement ces outils ne sont pas libre, logiciel open source. En plus, chaque application requiert en terme de préparation de données, différents systèmes (Java, MS Windows etc) et une courbe d'apprentissage des utilisateurs pour accéder aux compétences nécessaires à l'utilisation de l'application. Dans d'autres cas, les algorithmes pour la modélisation des niches fondamentales ont été développés à l'intérieur de trousseaux à outils de scripts, tel que le Climate Space Model (8) qui a été développé à partir de MatLab.

Ceci rend difficile la redistribution dans des applications conviviales. Pour s'attaquer à ces (et d'autres) problématiques, le projet *openModeller* a été initié en 2003 par la CRIA⁴². Le projet *openModeller* est gratuit et open source, et tout le code est disponible sur SourceForge. Le développement d'*openModeller* est supporté par la FAPESP⁴³, et par les contributions de la communauté des développeurs open source. Le but principal du projet *openModeller* est de mettre à disposition des outils pour les chercheurs qui s'intéressent à la modélisation des niches fondamentales. C'est une approche nouvelle dans le monde de la modélisation des niches fondamentales puisque l'architecture des plugins devrait permettre l'incorporation simple de nouveaux algorithmes. Fournir tous les algorithmes sous une architecture commune devrait permettre une comparaison plus simple des résultats entre les différents algorithmes puisque tous les modèles pourront être utilisés avec les mêmes données indépendantes et pourront fournir un résultat dans un format commun. L'implémentation de GARP, Bioclim, CSM, Environmental Distance et d'autres ont déjà été développés sous forme de plugins d'*openModeller*.

⁴²Centro de Referência em Informação Ambiental : <http://www.cria.org.br>

⁴³<http://www.fapesp.br>

⁴⁴GDAL - Geospatial Data Abstraction Library : <http://www.gdal.org>

⁴⁵GSL - GNU Scientific Library : <http://www.gnu.org/software/gsl/>

⁴⁶QGIS - Quantum GIS : <http://qgis.org>

openModeller tire profit des logiciels open source existants comme GDAL⁴⁴, GSL⁴⁵, QGIS⁴⁶ etc. *openModeller* est écrit en C++ et est multi-plateforme fonctionnant sur MS Windows, Mac OS X et GNU / Linux.

Les sous projets d'OpenModeller

Il y a de nombreuses thématiques de développement au sein du projet *openModeller*. La première et la plus importante est la bibliothèque *openModeller*. Cette bibliothèque fournit une plateforme commune pour tous les autres outils mentionnés plus bas. Cela inclut la capacité de lire des données environnementales et d'occurrence d'espèces, le paramétrage des paramètres du modèle, le chargement et le lancement des algorithmes et l'écriture des résultats de l'analyse sous forme d'un jeu de données géospatiales. Les liaisons entre C++ et Python (généré en utilisant SWIG) pour la bibliothèque sont disponibles.

Une API de webservice basée sur SOAP (en utilisant le style document / littéral) est disponible pour l'exécution à distance de travaux sur *openModeller*. Une application CGI est disponible, permettant à ce service d'être déployé dans une instance de serveur web Apache.

De nombreux outils en ligne de commande ont été créés par-dessus cette bibliothèque. *om_console* fournit une manière de lancer un modèle en utilisant un simple fichier texte de configuration qui spécifie quels jeux de données d'entrée à utiliser. *om_create* et *om_project* fournissent une manière d'utiliser *openModeller* en utilisant un fichier de configuration XML. D'autres outils d'aide sont également disponibles en ligne de commande pour fournir des extractions simples de données et des requêtes facilitées sur les métadonnées. Par exemple *om_sampledump* extrait les valeurs environnementales de chaque point présent et fournit le résultat dans un format de fichier délimité.

Pour les utilisateurs finaux, il y a une application multi-plateforme bureautique avec une interface graphique (écrite en C++ et Qt4). *openModeller Desktop* fournit un usage et un environnement convivial pour la préparation, le lancement et la visualisation des résultats des modèles de niches fondamentales. *openModeller Desktop* sera décrit plus en détail dans la section qui va suivre.

⁴⁷GBIF - Global Biodiversity Information Facility : <http://gbif.org>

⁴⁸<http://smlink.cria.org.br>

⁴⁹*openModeller Desktop* est capable de modéliser plusieurs espèces en une seule expérimentation, mais dans le cas d'une étude de cas simple seule une espèce sera utilisée

openModeller Desktop

openModeller Desktop est une solution pour les utilisateurs souhaitant utiliser la bibliothèque *openModeller* sans programmation ou être familier avec les différents outils en ligne de commande que le projet *openModeller* fournit. Dans *openModeller Desktop*, les utilisateurs peuvent exécuter les fonctions du noyau suivantes :

- retrouver automatiquement les données présentes depuis les bases de données en ligne comme GBIF⁴⁷ et specieslink⁴⁸ ;
- gérer les couches rasters environnementales qui vont être utilisées pour la modélisation ;
- gérer les paramètres des différents algorithmes des plugins inclus avec *openModeller* ;
- concevoir une expérimentation qui lancera de nombreux modèles pour une ou plusieurs espèces en utilisant un ou plusieurs algorithmes ;
- visualiser les résultats de l'expérimentation en voyant les modèles individuels en utilisant un tableau et une carte.

Modélisation d'*Acacia cyclops* - un cas d'étude simple

Un usage typique de la modélisation des niches fondamentales est la prédiction des surfaces qui pourraient devenir vulnérables à la progression des espèces invasives. Dans cet exemple, *openModeller* sera utilisé pour trouver les surfaces de distribution potentielle d'*Acacia cyclops*⁴⁹. *Acacia cyclops* est une plante native du sud ouest Australien, mais elle est connue pour être une plante invasive nuisible dans d'autres parties du monde. Ainsi le but de cette étude est d'établir les surfaces potentiellement vulnérables à une invasion par cette espèce. Pour commencer, l'outil de recherche de localisation d'*openModeller Desktop* a été utilisé pour chercher les enregistrements de localisation. (voir Figure 1).

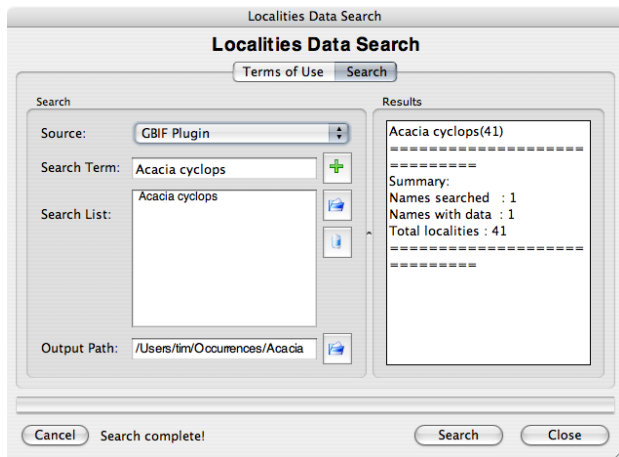


FIG. 1 – Recherche de GBIF pour localiser les données.

Puisque le résultat de la recherche renvoie des résultats en dehors de l'Australie, un éditeur de texte est utilisé pour supprimer toutes les occurrences ne se situant pas en Australie. Cela est réalisé pour modéliser les niches fondamentales basées seulement sur la connaissance des espèces dans leur distribution native. Cela réduit le résultat initial de 41 enregistrements retrouvés à partir de GBIF à 31 enregistrements, tous ont pour origine l'Australian Virtual Herbarium (1) (l'herbier virtuel australien). Après cela, le gestionnaire de jeu de données d'*openModeller Desktop* est utilisé pour définir un jeu de données. 23 couches dérivées du CRU⁵⁰ ont été sélectionnées pour cet usage (voir figure 2.). Ces couches représentent des indicateurs relatifs au climat (basé sur des enregistrements de mesure du climat), relatifs à la température, aux précipitations etc.

⁵⁰CRU - Climate Research Unit : <http://www.cru.uea.ac.uk>

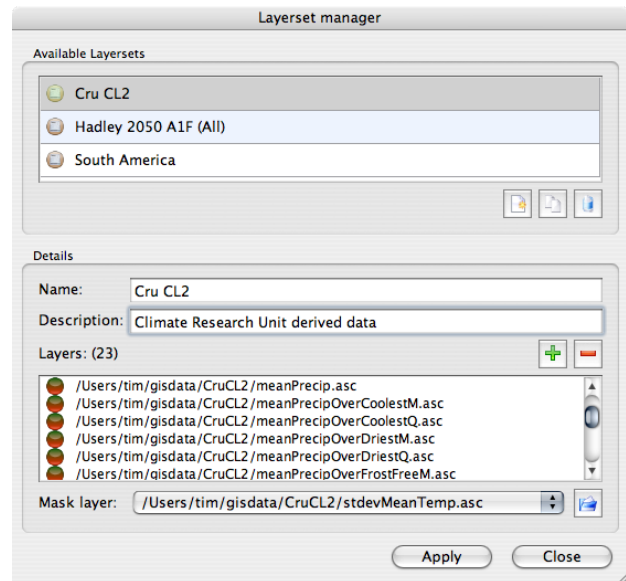


FIG. 2 – le gestionnaire de jeu de données.

La création des noms de jeux de données fournit un grand bénéfice pour les utilisateurs puisqu'ils pourront facilement faire référence à de multiples couches comme une seule entité virtuelle et n'auront pas besoin de perdre beaucoup de temps à sélectionner les fichiers depuis le système de fichier chaque fois qu'ils lancent une expérimentation. Une fois qu'un jeu de données a été défini, il peut être utilisé de manière répétitive dans différentes expérimentations.

openModeller Desktop inclut un gestionnaire d'algorithmes (fig. 3). Cet outil permet de voir tous les plugins de modélisation d'algorithmes disponibles qui ont été installés avec la bibliothèque *openModeller*. Tous les algorithmes sont distribués avec des valeurs par défaut (comme déterminé par les auteurs de l'algorithme) donc il peuvent être utilisés 'out the box'. Dans le gestionnaire d'algorithme, ces paramètres par défaut sont présentés comme des 'profils système' immuables. Les profils systèmes peuvent être clonés et modifiés pour créer des 'profils utilisateurs'. La modification de ces paramètres requiert certaines connaissances sur les principes sous-jacents de l'algorithme, mais permet aux utilisateurs avancés une grande flexibilité et contrôle. Les utilisateurs novices peuvent éviter le gestionnaire d'algorithmes et simplement travailler avec les paramètres par défaut des 'profils système'.

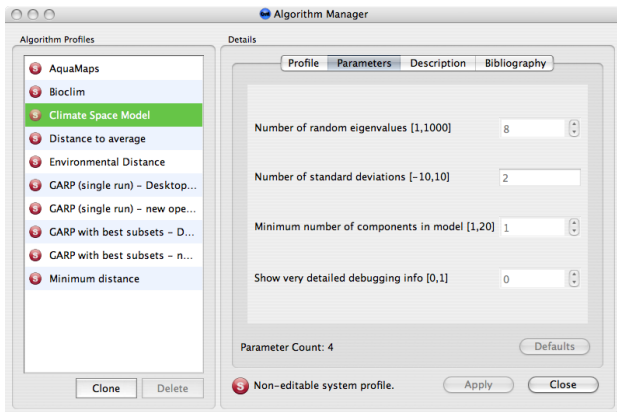


FIG. 3 – Le gestionnaire d'algorithmes.

Les données présentes, les jeux de données et les profils d'algorithmes sont les trois éléments essentiels pour la structuration de l'expérimentation de modélisation. La structure de l'expérimentation (voir Figure 4.) est utilisée pour définir comment ces données vont être combinées pour former une expérimentation. Chaque expérimentation créée possède un nom unique et une description. La désignation de l'expérimentation fournit une interface utilisateur simple avec de puissantes fonctionnalités. Avec elle, d'importantes expérimentations peuvent être créées pour modéliser de multiples espèces utilisant de nombreux algorithmes. Dans le futur, la gestion de la modélisation avec de multiples scénarios de climats sera rajouté. Dans *openModeller Desktop*, l'actuelle invocation du processus de modélisation se fait via un 'plugin de modélisation'. Cela signifie que l'interaction entre l'interface 'bureautique' et la bibliothèque *openModeller* est bien séparée. Cela veut également dire que de nouveaux plugins de modélisation peuvent être écrits pour supporter d'autres outils de modélisation dans le futur. Actuellement, *openModeller Desktop* possède deux tels plugins disponibles : un plugin 'local' qui utilise directement *libopenModeller* et un plugin 'service web' qui pourra appeler des travaux de modélisation sur un serveur distant en utilisant *openModeller service web API*. Ce dernier reste quelque peu expérimental, mais le but est de permettre aux utilisateurs de concevoir de grosses expérimentations nécessitant de fortes ressources en CPU en utilisant l'interface agréable d'*openModeller Desktop*. L'expérimentation sera alors soumise via le service web API pour être lancée sur un groupe de serveurs puissants, avec seul le résultat de la modélisation de sortie renvoyé au client pour une visualisation en local.

Une fois que l'expérimentation est complète, la

principale fenêtre de l'application fournit une vue arborescente de l'expérimentation. Le noeud le plus haut de cet arbre est l'expérimentation elle-même. Sous celui-ci, les noeuds à mi-niveau représentent chaque profil d'algorithme utilisé pour l'expérimentation. Les noeuds terminaux de l'arbre représentent les modèles individuels. Cliquer sur le noeud d'un modèle fournit un rapport détaillé du modèle. Le modèle peut être également visualisé spatialement en utilisant l'onglet carte. L'onglet carte utilise un navigateur SIG (basé sur les bibliothèques de Quantum GIS) pour permettre à l'utilisateur de se déplacer et de zoomer de manière interactive autour de la région géographique du résultat du modèle.

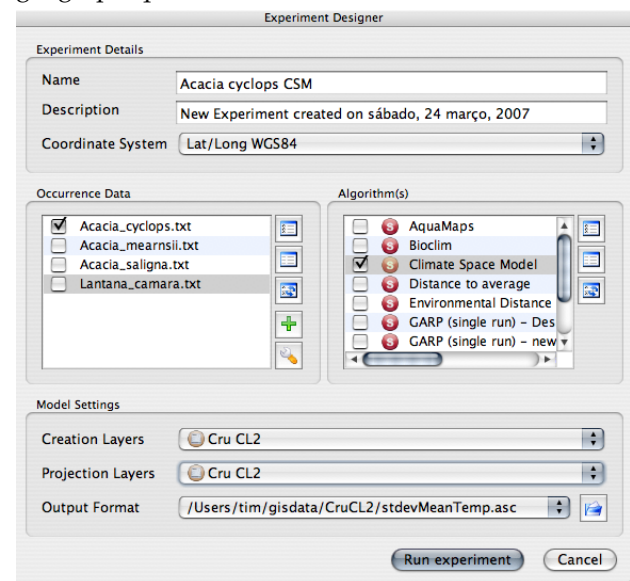


FIG. 4 – Le concepteur d'expérimentation.

Dans le cas d'étude d'*Acacia cyclops*, seulement les données présentes en Australie ont été utilisées pour générer le modèle de niche fondamentale (Figure 4.) qui a été projeté dans un jeu de données global sur le climat. Dans la figure 5, les résultats de ce modèle sont montrés pour le sud ouest de l'Australie. Les points présents originaux utilisés pour créer le modèle ont été représentés par un point noir sur l'image. Les surfaces en bleu foncé représentent une faible correspondance avec le modèle de la niche fondamentale de l'espèce. Lorsque les rangs de couleurs deviennent bleu clair, jaune et enfin rouge, une correspondance croissante avec la modélisation de la niche fondamentale de l'espèce est démontrée. Par déduction, si les espèces sont plantées dans les surfaces délimitées en rouge, alors elles ont probablement une bonne chance de survie. Dans la figure 6, le même modèle de sortie est montré, mais la surface

d'intérêt se situe au sud de l'Afrique. En se basant sur le modèle, nous pouvons émettre le postulat que la région la plus à l'ouest du Cap de l'Afrique du Sud représente un habitat optimal similaire pour *Acacia cyclops*. *Acacia cyclops* est vraiment une espèce invasive problématique dans l'ouest du Cap, qui a été introduit ici pour stabiliser le sable des dunes (11).

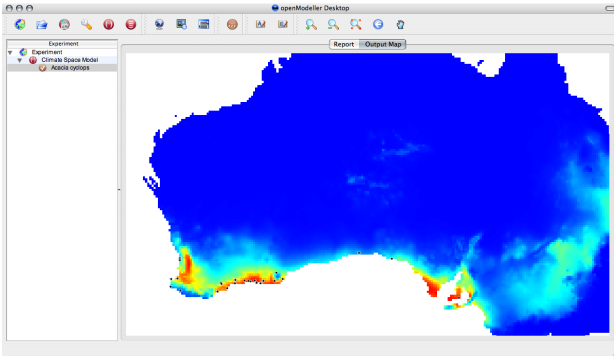


FIG. 5 – Le résultat du modèle pour le sud ouest de l'Australie.

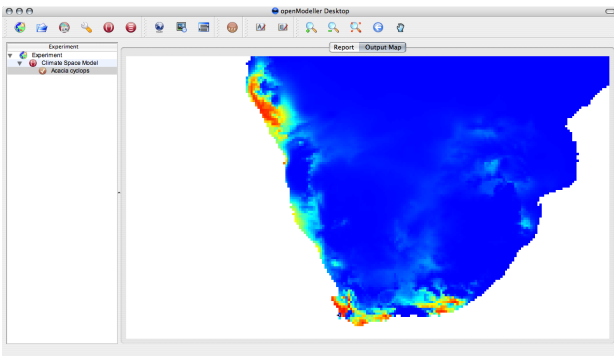


FIG. 6 – Le résultat du modèle pour le sud de l'Afrique.

Conclusion

openModeller fournit des solutions pour les utilisateurs et développeurs qui s'intéressent à la modélisation des niches fondamentales. La bibliothèque *openModeller* permet un accès par programmation à ses fonctionnalités. Les outils de la console permettent le scriptage et l'usage interactif de ligne de commande. L'application bureautique fournit un environnement graphique simple d'usage. L'API de web service d'*openModeller* fournit un moyen de déployer *openModeller* sur de puissants serveurs (et dans un futur proche, sur des clusters). L'approche en plugins pour implémenter différents algorithmes de modélisation permet à l'utilisateur d'expérimenter avec

différentes techniques de modélisation sans avoir à apprendre l'utilisation d'une nouvelle application à chaque fois, ou de devoir préparer des données d'entrée d'une manière différente pour chaque application. Le framework de plugins des algorithmes permet aux développeurs d'algorithme de se focaliser uniquement sur le développement du coeur de l'algorithme sans avoir besoin de 'réinventer la roue' en écrivant des routines d'entrée / sortie de données et une logique d'interface utilisateurs.

L'application *openModeller Desktop* fournit un environnement dans lequel tous les aspects de préparation de données, de structure de l'expérimentation, de lancement de l'expérimentation et de visualisation des résultats d'une expérimentation, peuvent être exécutés. *openModeller Desktop* peut être compilé sur trois des systèmes d'exploitation bureautique majeurs (MS Windows, Mac OS X, GNU/Linux) et est librement disponible sous licence GPL.

L'équipe d'*openModeller* est toujours avide de collaborer avec d'autres projets, utilisateurs et développeurs. Nous vous encourageons à télécharger et utiliser le logiciel et nous accueillons tous les retours. Nous fournissons un support par des listes de diffusion et IRC - visitez notre page principale pour plus de détails.

Bibliographie

- [1] Council of Heads of Australian Herbaria. 2007. **Australia's Virtual Herbarium via Centre for Plant Biodiversity Research** In <http://www.cpbr.gov.au/cgi-bin/avh.cgi> (Accessed 25 March, 2007)
- [2] Brewer, P.W. 2003. **Modelling the global distribution patterns of Leguminosae species in past, present and future climates** PhD Thesis, University of Reading, UK.
- [3] Elith, J., Graham C.H., Anderson R.P., Dudik M., Ferrier S., Guisan A., Hijmans R. J., Huettmann F., Leathwick J.R., Leh-

- mann A., Li J., Lohmann L.G., Loiselle B.A., Manion G., Moritz C., Nakamura M., Nakazawa Y., Overton J.McC., Peterson A.T., Phillips S.J., Richardson K.S., Scachetti-Pereira R., Schapire R.E., Soberon J., Williams S., Wisz M.S., Zimmermann N. E. 2006. **Novel methods improve prediction of species' distributions from occurrence data.** In *ECOGRAPHY* 29(2) :129-151
- [4] Hutchinson, G.E. (1957). **Concluding Remarks.** In *Cold Spring Harbor Symposia on Quantitative Biology.* 22 : 415-42.
- [5] Nix, H.A. (1986). **A biogeographic analysis of Australian Elapid snakes.** In Longmore, R. (ed.) *Atlas of Australian Elapid Snakes.* Australian Flora and Fauna Series 8 : 4-15.
- [6] Peterson, A.T., Bauer, J.T., Mills, J.N. 2004. **Ecologic and Geographic Distribution of Filovirus Disease.** In *Emerging Infectious Diseases* 10(1) :40-47
- [7] Phillips, S. J., Anderson, R. P., and Schapire, R. E., (2006). **Maximum entropy modeling of species geographic distributions.** In *Ecological Modelling*, 190, 231-259.
- [8] Robertson, M. P., Caithness, N., and Villet, M. H. 2001. **A PCAbased modelling technique for predicting environmental suitability for organisms from presence records.** In *Diversity and Distributions* vol.7 : 15-27.
- [9] Scachetti-Pereira, R. (2002). **Desktop GARP.** In <http://lifemapper.org.br/desktopgarp> (Accessed on January 10th 2007).
- [10] Soberon, J., and Peterson, A. T. 2005. **Interpretation of Models of Fundamental Ecological Niches and Species' Distributional Areas.** In *Biodiversity Informatics.*
- [11] Stirton, C.H. (ed). 1983. **Plant invaders - beautiful but dangerous.** Department of Nature and Environmental Conservation, Cape Town.
- [12] Thomas, C.D., Cameron, A., Green, R.E., Bakkenes, M., Beaumont, L.J., Collingham, Y.C., Erasmus, B. F. N., de Siqueira, M.F., Grainger, A., Hannah, L., Hughes, L., Huntley, B., van Jaarsveld, A.S., and Midgley, G.F., and Miles, L., Ortega-Huerta, M.A., Townsend, A., Phillips, O.L., and Williams, S.E. 2004. **Extinction risk from climate change** In *Nature* :427
- [13] Yesson, C., Culham, A. 2006a. **Phyloclimatic Modelling : Combining Phylogenetics and Bioclimatic Modelling.** In *Systematic Biology* : in press.
- [14] Yesson, C., Culham, A. 2006b. **A Phyloclimatic Study of Cyclamen.** In *BMC Evolutionary Biology.* 6 :72

Tim Sutton
 CRIA, Campinas, Brazil
<http://openmodeller.sf.net>
 tim AT linfiniti.com

Étude de cas

MapGuide OS dans le projet de cartographie de la forêt urbaine de San Francisco

La cartographie en open source permet à la ville de voir les arbres de sa forêt.

par Alex Fordyce et Charlie Crocker, traduit par Gwenael Bachelot

Abstract

La ville et le comté de San Francisco, ainsi que l'association à but non lucratif "Friends of the Urban Forest" (FUF) et Autodesk ont lancé le projet de cartographie de la forêt urbaine de San Francisco (San Francisco Urban Forest Mapping Project). Ce projet utilise la technologie MapGuide Open Source comme plate-forme centrale pour la cartographie en ligne de plusieurs dizaines de milliers d'arbres dans les espaces publics de la ville.

C'est l'association FUF qui, la première, a lancé l'idée de cartographier les arbres, dans le but de faire participer le public et de susciter son implication autour de la forêt urbaine. Le projet a ensuite évolué de manière à soutenir la ville dans sa stratégie de plan-

tation d'arbres afin d'optimiser les avantages qu'elle peut apporter en matière d'environnement et de réduction des coûts relatifs à l'entretien et au remplacement des arbres. Ce projet de cartographie de la forêt urbaine assiste le bureau de la forêt urbaine (Bureau of Urban Forestry, BUF) et l'association FUF dans la plantation et l'entretien des arbres et ouvre la voie vers une utilisation des informations géospatiales dans d'autres villes.

La carte dynamique intègre plus que des données spatiales

Le BUF et la FUF géraient les données sur la forêt dans des systèmes séparés, reposant en grande partie sur des cartes papier, ce qui rendait difficile le partage des informations. Les employés du BUF sur le terrain répertoriaient les emplacements des arbres sur papier, ces informations étaient ensuite entrées dans une base de données avec les informations sur les autorisations émises concernant l'entretien privé

des arbres. Pour les mises à jour, les données étaient récupérées lors des opérations d'entretien routinier des arbres, puis insérées dans cette base.

Le BUF estime à environ 100 000 le nombre d'arbres plantés dans les lieux publics de la ville. Pour autant, le nombre d'arbres et d'autorisations répertoriés dans sa base de données s'élève à moins de la moitié de ce total. Pareillement, La FUF évalue à 41 000 le nombre d'arbres plantés par ses soins, pour lesquels des données ont été recueillies.

Il paraissait évident pour ces deux organisations qu'un inventaire plus complet et plus précis de la forêt urbaine de la ville leur permettrait de mieux accomplir leurs objectifs communs. Ainsi, le nouveau système de gestion de la forêt urbaine comprend une base de données intégrée, un outil de cartographie et des applications de rapport en ligne fonctionnant dans un environnement logiciel hybride, libre et propriétaire.

"Grâce à un modèle mixte de technologies libres et propriétaires, nous avons été en mesure de créer un système répondant à tous nos besoins de développement et de fonctionnement", a déclaré Greg Braswell, responsable informatique et SIG au bureau d'ingénierie du département "Travaux publics" de la ville de San Francisco. "Avec MapGuide Open Source, et grâce à la communauté de développement open source, nous bénéficions d'outils de collaboration performants et d'une assistance de meilleur niveau en ce qui concerne les sources de données et le géocodage que ce que peuvent offrir les fournisseurs d'outils commerciaux (propriétaires)."

La nouvelle carte de la forêt urbaine de San Francisco est reliée à un espace de stockage centralisé, contenant les données de l'inventaire de la forêt urbaine, via le Web. Un schéma de base de données partagé permet de prendre en compte les données relatives aux attributs des arbres, intégrées par le BUF et la FUF, ainsi que des mises à jour en temps réel des données spatiales effectuées par le personnel de ces organisations.

Le système combine les technologies libre et propriétaire pour répondre aux besoins de la ville

Toutes les informations concernant les arbres (objet spatial ponctuel et données attributaires) sont stockées dans une base de données Microsoft SQL Server.

Les informations sur les attributs sont accessibles via ASP.NET C#, tandis que les emplacements sont

cartographiés directement à partir de la base de données grâce à la technologie open source d'accès aux données FDO (Feature Data Object) fournie avec MapGuide Open Source.

Le personnel du BUF et de la FUF possédant déjà une certaine expérience de la plate-forme Microsoft ne nécessite pas de formation supplémentaire pour maintenir le système.

D'une même voix, le BUF et la FUF ont reconnu que le développement de la solution dans l'environnement .NET permettrait à l'industrie géospatiale d'en savoir plus sur le développement d'applications en relation avec le monde open source. Le projet tire également parti d'autres travaux de la communauté open source. En raison de l'utilisation par la ville du format MrSID pour la compression des images satellite, l'équipe du projet de cartographie de la forêt urbaine a utilisé l'extension raster de MapGuide Open Source développée par Frank Warmerdam pour prendre en charge ce format raster propriétaire.

Les données spatiales de la forêt urbaine et les détails de l'inventaire sont combinés au référentiel de cartographie de la ville de San Francisco pour les rues, les parcelles, les types de sols et autres couches, de manière à créer une carte interactive sur le Web. Ce projet est également une bonne occasion de tirer parti des performances obtenues grâce au format SDF de MapGuide Open Source, un format de fichier spatial optimisé pour l'environnement Web. Les couches du référentiel, qui nécessitent une mise à jour moins fréquente que les emplacements des arbres, sont stockées en tant que fichiers SDF, tandis que les informations dynamiques sur les arbres sont géocodées et stockées dans une base de données. Le stockage des informations sur l'emplacement des arbres dans une base de données permet à d'autres applications d'accéder à ces données et de les mettre à jour de manière cohérente au sein de l'organisation. Ces mises à jour sont ensuite visibles en temps réel sur l'application Web. Des améliorations ultérieures au système permettront aux utilisateurs autorisés de corriger les erreurs de codage des données géographiques en déplaçant les points pour indiquer plus précisément les emplacements des arbres. Grâce à la superposition d'une image satellite à la carte, il est possible de visualiser le décalage éventuel entre un arbre géocodé et son emplacement réel.

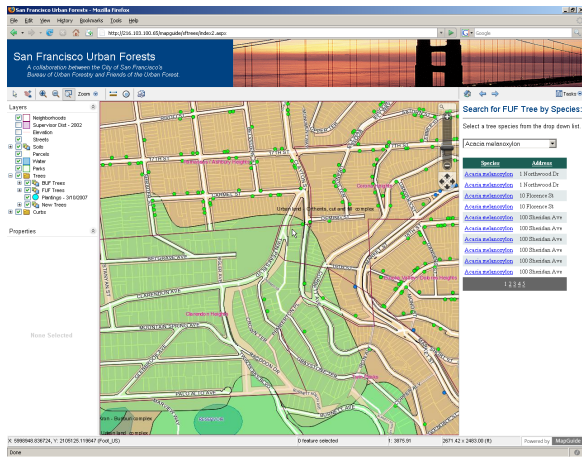


FIG. 1 – Le système de cartographie de la forêt urbaine de San Francisco permet de cartographier rapidement les arbres recensés par le BUF et la FUF à l'aide d'une recherche par adresse.

La logique et les règles métiers ont été programmées à l'aide de ASP.NET C# afin de définir et de contrôler l'activité des applications en fonction de l'utilisateur et de son rôle, et de suivre l'historique des modifications. Si tous les utilisateurs accèdent aux mêmes fonctionnalités lorsqu'ils travaillent avec l'application, seuls les utilisateurs autorisés peuvent accéder à certaines fonctions. Le grand public a la possibilité de parcourir la forêt urbaine pour rechercher et afficher des données sur les arbres selon certains critères (par adresse, quartier, espèce). Il peut également effectuer un zoom ou un déplacement panoramique, activer ou désactiver l'affichage des cartes et ajouter des informations pertinentes sur des arbres telles que l'adresse, des photos, des commentaires et des coordonnées de contact. En revanche, les utilisateurs autorisés ont à leur disposition une large gamme de fonctionnalités et peuvent modifier et mettre à jour les données sur les arbres.

Le projet de cartographie augmente la pertinence et la valeur des données

On estime que le système de cartographie de la forêt urbaine de San Francisco permettra d'écono-

miser l'argent de la ville en donnant la possibilité d'inventorier et de cartographier les lieux arborés existants, futurs et passés, d'évaluer les coûts et les avantages de la forêt urbaine dans son ensemble ou dans des zones spécifiques, de déterminer les stratégies efficaces de plantation et d'entretien des arbres et de rationaliser les procédures telles que les applications des permis. Des améliorations ultérieures du système de cartographie de la forêt urbaine pourraient faciliter l'aménagement urbain en permettant aux responsables de l'administration forestière d'établir un modèle, d'analyser l'ensemble des coûts et des avantages et de cibler une approche stratégique afin de tirer parti des nombreux microclimats de San Francisco.

En outre, la FUF et les responsables municipaux ont pour objectif d'utiliser cet outil pour susciter une implication de la part des membres de la communauté de San Francisco autour de la forêt urbaine et des efforts pour promouvoir les espaces verts. Le grand public a la possibilité d'ajouter des arbres à la carte ; ces informations sont répertoriées dans une base de données différente dans l'attente de leur validation. Lorsque les données sont validées, les arbres sont ajoutés à la base de données de la forêt urbaine.

L'utilisation d'une technologie de cartographie open source peut s'étendre à d'autres forêts urbaines que celle de San Francisco. L'équipe du projet s'attend à ce que ce programme soit également utilisé par d'autres villes, comtés et organisations de travaux publics pour cartographier et gérer des actifs, quels qu'ils soient. La figure 1 montre l'hôtel de ville entouré d'arbres.

Pour plus d'informations sur le projet de cartographie de la forêt urbaine de San Francisco, visitez le site <http://www.urbanforestmap.org>.

Alex Fordyce

Fondateur et développeur SIG senior, Online Mapping Solutions LLC

[afordyce AT ix.netcom.com](mailto:afordyce@ix.netcom.com)

Charlie Crocker

Chef de produit senior, Geospatial Solutions, Autodesk

[charlie.crocker AT autodesk.com](mailto:charlie.crocker@autodesk.com)

Étude d'intégration

Créer des cartes prêtes à être publiées avec GRASS et GMT

par Dylan Beaudette, traduit par Yves Jacolin

Aperçu

La publication de cartes imprimables de haute qualité est un composant important de la plupart des analyses géographiques. Bien que GRASS n'ait pas été écrit pour la production de matériels imprimables, les commandes du type `ps.map` et plus récemment un wrapper bien utile `G-ps.map`, lui permettent de sortir des cartes postscripts de haute qualité. Cependant, la syntaxe et les possibilités plutôt limitées de l'approche `ps.map` produisent souvent une carte qui doit être finalisée dans un logiciel de PAO (Publication Assisté par Ordinateur) tel que Inkscape, Scribus, ou The GIMP. La fonctionnalité composer une carte trouvée dans QGIS peut être une bonne alternative dans un proche futur, mais la version actuelle manque de flexibilité, de stabilité, et de la possibilité d'exporter des données dans des formats tels que EPS ou PDF. Avec de telles options limitées pour la production de cartes à imprimer de grande qualité dans un processus open source, de nombreuses personnes ont bricolé ensemble le nécessaire requis pour

interfacer GRASS avec de nombreux programmes comme le package Generic Mapping Tools. Le projet GMT est très bien décrit par ses auteurs :

GMT est une collection open source d'une soixantaine d'outils pour la manipulation de jeux de données Cartesian et géographiques (comprenant les filtres, l'ajustement de la direction, la création de grilles, la projection etc.) et la production dans des fichiers PostScript Encapsulé (EPS) d'illustrations qui vont du simple graphe x-y en passant par des cartes de contour à la surbrillance artificielle des surfaces et des vues en perspective en 3D. GMT gère environ 30 projections et transformations cartographiques et est fourni avec des données telles que la ligne des côtes, les rivières, les limites politiques. GMT est développé et maintenu par Paul Wessel et Walter H. F. Smith avec l'aide d'une équipe de volontaires, et est géré par la Fondation Nationale des Sciences. Il est diffusé sous la licence GNU General Public.

Cet article décrit les efforts combinés de plusieurs personnes (David Finlayson, Hamish Bowman, Brent Wood, moi-même, et d'autres) pour créer le moyen d'interfacer GRASS à GMT. Plusieurs permutations des commandes du style *.out.gmt ont été créées, en utilisant à la fois des scripts Python et Bash, pour réaliser l'export nécessaire, la configuration, et l'exécution des commandes GMT. Après avoir lu cet article vous devriez avoir une idée générale pour composer de simples cartes dans GMT, avec des scripts modèles pour l'utilisation de données exportées de GRASS. L'article suivant présentera des exemples spécifiques à Spearfish (le jeu de données standard de GRASS) de création de cartes avec GMT.

GMT

Les applications des Outils de Cartographie Générique (GMT) peuvent être installées sur n'importe quelle plate-forme soit par des binaires préparés ou en compilant le code source. Les outils GMT peuvent aussi être installés à partir de votre système de gestion de paquets de votre distribution (par exemple aptitude), cependant cette méthode n'est pas recommandée puisque le paquet a de grandes chances de ne pas être à jour. Si vous êtes sur un système d'opération de type UNIX, compiler à partir des sources est relativement aisé, et assure que vous avez une version à jour. En supposant que vous avez un ensemble d'outils de développement en fonctionnement (gcc, make, etc.) à aller sur la page 'Download' ⁵¹, et suivez les instructions. GMT est distribué avec un manuel compréhensible, des cours pour créer des cartes, et des données pour des cartes à différentes échelles. Une fois que vous avez une copie fonctionnelle de GMT installée sur votre machine, et que vous avez téléchargé les données d'exemple ⁵² vous pouvez suivre les exemples de cet article. Cette archive contient également un ensemble de scripts approfondis dans plusieurs exemples dans cet article. Le script `template.sh` peut être utilisé pour démarrer pour une approche automatisée pour exporter et tracer des données Raster et vecteur de GRASS avec des outils de GMT. Remarquez que ce script est un point de démarrage et est loin d'être complet.

Les commandes GMT sont lancées depuis la console (avec un jeu complexe de paramètres, switches et d'autres arguments) et envoient les fragments Postscript qui en résultent vers la sortie standard. Le comportement par défaut de toutes les ap-

plications GMT peut être ajusté avec la commande `gmtset`. Cette commande est habituellement utilisée avant toute commande qui produit un rendu, ainsi des éléments clé tels que la taille du papier, l'espace des polices etc. sont établis. La première commande de traçage lancée (généralement `psbasemap`) crée le fichier de sortie initial avec l'opérateur standard ">", tandis que chaque appel suivant ajoute à ce fichier avec l'opérateur d'ajout double ">>". Le premier paramètre `-K` est utilisé avec toutes les commandes GMT avant la dernière commande, ainsi le fichier Postscript en sortie n'est pas finalisé prématurément. Nous utiliserons des données échantillons d'une application Mapserver pour illustrer divers aspects d'utilisation de GMT, avec une idée de comment coupler cela avec GRASS et GMT.

Application échantillon

```
# define some global settings:
gmtset ANNOT_FONT_PRIMARY Times-Roman \
HEADER_FONT_SIZE 16 \
ANNOT_FONT_SIZE_PRIMARY 12 \
LABEL_FONT_SIZE 14 \
BASEMAP_TYPE plain \
PLOT_DEGREE_FORMAT DF \
PAPER_MEDIA letter+

# sample map centered on the western USA
pscoast -JB-116/36/30/42/7i \
-R-125/-108/31/44 -B5 \
-Gwhite -W0.5p \
-A250 -Dh -Na -Xc -Yc -P -K > query_centers.eps

# plot mapxy points:
psxy sample_data/mapxy_locations.latlong -J -R \
-Sc0.075c -W1/1/200/1 -G1/200/1 -O -K >> query_centers.eps

# plot ka-map points
psxy sample_data/ka-map_locations.latlong -J -R \
-Sc0.075c -W1/255/1/1 -G255/1/1 -O >> query_centers.eps
```

L'exemple ci-dessus crée un graphique de l'Est des États-Unis, avec des points rouges et verts symbolisant l'activité spécifique à l'endroit d'un site Internet. Le cadre de la carte est créé avec la commande `psbasemap`, les lignes de côtes et limites politiques sont créées par `pscoast`, et les points sont créés avec la commande `psxy`. La localisation des points est stockée comme des paires simples de longitude-latitude au format texte, tandis que la ligne de côte est dessinée à partir du jeu de données inclus dans GMT. Le résultat peut être vu à la Figure 1.

Une explication complète des diverses options des lignes de commandes peut être trouvée dans la page de manuel de chaque commande. De plus, un

⁵¹http://gmt.soest.hawaii.edu/gmt/gmt_download.html

⁵²http://169.237.35.250/~dylan/GRASS/newsletter/sample_data.tar.gz

exemple plus complet de GMT réalisé en dehors de GRASS peut être trouvé sur le site de l'auteur ⁵³. Dans l'exemple ci-dessus, les paramètres de projection et de région sont définis manuellement par l'utilisateur. Des éléments de formatage additionnels (tel que le paramètre `-P` pour un affichage en portrait) doivent être définis manuellement et définis en ligne de commande. La génération automatique d'une carte dépend alors, en partie, d'une approche dynamique pour définir les options passées aux commandes GMT. Un langage de script, capable d'extraire les informations de région de GRASS à partir d'une session de GRASS en cours est une approche idéal pour automatiser cette tâche.

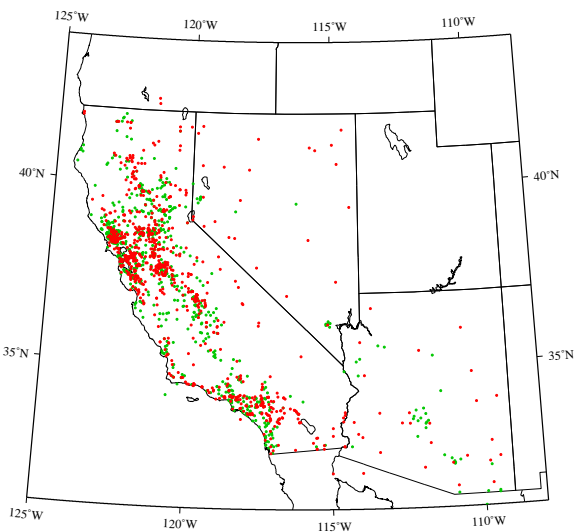


FIG. 1 – Localisations de requête, symbolisé par une application cartographie active. Les points verts représentent une application Mapserver basée sur LandView, les points rouges représentent une application Mapserver basée sur Ka-map.

⁵³<http://casoilresource.lawr.ucdavis.edu/drupal/node/102>

⁵⁴On peut trouver plus d'informations sur l'interpréteur de script GRASS ici http://grass.itc.it/gdp/html_grass63/g.parser.html

Coupler GRASS et GMT

L'approche générale pour interfacier GRASS et GMT peut être résumée comme suit :

- collecter des informations à partir d'utilisateurs sur :
 - les données en entrées
 - les options de formatage
 - les options du rendu
- collecter des paramètres géographiques à partir de la région GRASS
- calculer les paramètres clés de mise en page
 - information sur la projection
 - limites d'échelle
 - intervalles des échellons
- exporter les données raster de GRASS vers le format .gd de GMT
- exporter les données vecteurs de GRASS vers le format ASCII de GMT
- lancer les commandes GMT avec les paramètres ci-dessus

Collecter des informations à partir de l'utilisateur

Le parseur de script GRASS fournit une approche aisée à la création de commandes GRASS personnalisées, sans la nécessité de comprendre le langage C. Un script habile et rigoureux peut être utilisé pour parser des détails sur le jeu de données de GRASS comme des informations sur les paramètres de la région en cours aux commandes GMT, en utilisant la syntaxe familière et la structure de l'interface d'un module GRASS. Avec ces options sauveées dans des variables locales il est possible de modifier les paramètres par défaut tels que la taille du papier, la taille des polices et l'orientation du texte en lançant le programme `gmtset` avec nos variables locales substituées par l'interpréteur de script ⁵⁴.

Collecter des informations géographiques

Extraire des informations clé des paramètres de la région en cours (pour une utilisation des paramètres plus tard dans GMT) peut être réalisé de plusieurs manières. Parmi celles-ci une implique les outils `g.region` et de traitement de texte, grâce à Bruce Raup pour la suggestion élégante de `eval`. Remarquez que tous les exemples suivants nécessitent que vous soyez dans une session GRASS active.


```

# need to be in an active GRASS session for this to work.
#
# region extents: used with -R flag
# resolution: used with the -I flag
#
# elegant approach to extracting extent and
# region resolution as suggested by
# Bruce Raup - National Snow and Ice Data Center
#
region_assignments=`g.region -g`
eval $region_assignments

# setup flags for GMT commands
region="-R$w/$e/$s/$n"
inc="-I$wres/$nsres"

# extract extent values for calculating map aspect ratio
extent_assignments=`g.region -ge`
eval $extent_assignments

# preserve aspect ratio from UTM E,N coordinates:
aspect_ratio=`echo $ns_extent $ew_extent
| awk '{printf("%f", $1 / $2)}'`

# calculate the map length based on the
# original aspect ratio
map_length=`echo $map_width $aspect_ratio
| awk '{printf("%f", $1 * $2)}'`

# calculate the y location (in paper units)
# for the field sheet title
field_sheet_title_y=`echo $map_length
| awk '{printf("%f", $1 + 0.33)}'`

# compile the projection string, with
# width/length variables
#linear projection, width inches / length inches
projection_string="X${map_width}i/${map_length}i"

```

Une approche plus robuste impliquerait la liaison GRASS-Python pour extraire directement les paramètres de la région. Une fois que les données de la région ont été extraites et sauveées dans des variables locales, nous pouvons réaliser les calculs nécessaires pour obtenir les autres paramètres tels que l'échelle de la carte, le rapport de l'aspect de la carte, la taille de la carte sur le papier, etc. Les paramètres de la région sont utilisés pour le paramètre *de traçage de la région* (-R\$w/\$e/\$s/\$n), souvent utilisé seulement dans la première fonction de traçage. Toutes les fonctions de traçages suivantes peuvent hériter de ces valeurs en définissant des paramètres vides -J et -R en fonction du paramètre -O (superposition).

Les projections des cartes dans GMT

L'approche traditionnelle pour utiliser les outils GMT implique le traçage de données, encodé en paire longitude-latitude, dans une projection définie avec le paramètre -J. Les coordonnées sont transformées au fur et à mesure que la sortie Postscript

est générée. Comme noté dans l'exemple de code ci-dessus, une projection Alber Equal Area centrée sur (116°W, 36°N), en ayant des parallèles standards à 30°N et 42°N, et qui a 7 pouces de large est encodé par -JB-116/36/30/42/7i. Lisez le manuel de GMT pour une liste complète de projections gérées et de leurs paramètres.

Lorsqu'on travaille dans GRASS, les données sont habituellement (mais pas toujours) dans une projection géographique et donc le mode de projection *linear* doit être utilisé. Une fois qu'une largeur et une longueur appropriées de la carte (dans ce cas en utilisant des pouces) ont été établies les paramètres de la projection linéaire sont encodées par -JX\${map_width}i/\${map_length}i. Remarquez que les unités métriques ou impériales peuvent être utilisées dans GMT et passer de l'un à l'autre implique seulement l'ajout d'un 'c' pour centimètre et 'i' pour pouces ([NdT] inches en anglais) après le nombre.

Exporter et tracer des données raster GRASS

Exporter une donnée raster GRASS vers un format compatible GMT peut être réalisé avec une combinaison de r.out.bin et xyz2grd (inclu dans la distribution GMT). L'utilisation correcte de xyz2grd nécessite des connaissances dans les données raster en entrée : c'est à dire entier ou virgule flottante, etc. Pour une cellule raster exportée avec r.out.bin, le paramètre -ZTLh pourrait être utilisé pour renseigner xyz2grd que le flux de données en entrées est de type entier court orientée. Pour les cartes FCELL et DCELL les paramètres correspondants pourraient être respectivement -ZTLf ou -ZTLd. Le paramètre -F instruit xyz2grd pour utiliser l'enregistrement des pixels des grilles des cellules, le même enregistrement utilisé pour les rasters GRASS. Une simple déclaration CASE peut être utilisée dans nos exemples de scripts pour créer automatiquement l'appel à xyz2grd. Un paramètre optionnel dans notre exemple de script peut être utilisé pour sauter cette étape pour faciliter la récupération des autres paramètres sans perdre de temps de traitement pour exporter les données raster. L'auteur a trouvé que définir la région par une résolution entière entraîne l'opération la plus fiable de xyz2grd.

```

# obtenir le type de la carte d'un raster donn\ee
MAP_TYPE=`r.info -t "$output_raster" | cut -f2 -d'|`

# exporter le raster bas\ee sur son type,
# ses param\etres de r\egionde l'exemple ci-dessus

```

```
# et les param\`etres de r\`esolution de l'exemple plus haut
case "$MAP_TYPE" in
CELL)
r.out.bin input=$output_raster output=- null=-9999 \
| xyz2grd -G$output_raster.grd \
$region $inc -ZTLh -F -N-9999 ;;
FCELL)
r.out.bin input=$output_raster output=- null=-9999 \
| xyz2grd -G$output_raster.grd \
$region $inc -ZTLf -F -N-9999 ;;
DCELL)
r.out.bin input=$output_raster output=- null=-9999 \
| xyz2grd -G$output_raster.grd \
$region $inc -ZTLd -F -N-9999 ;;
esac
```

Une fois que les données rasters ont été exportées au format .grd nous sommes presque prêts pour leur utilisation avec les commandes de dessins de GMT telles que `psimage`. Cette commande peut soit utiliser un seul fichier grille, avec un fichier de palette de couleurs (.cpt), soit avec des canaux de couleurs séparées rouge, vert, bleu ⁵⁵.

Exporter les données des tables de couleur GRASS

Pour l'instant il n'y a pas de méthode *simple* disponible pour convertir les informations de la table des couleurs de GRASS en un fichier de palette de couleurs fiable pour GMT. Les développements existants sont capables d'approcher le format .cpt, cependant l'auteur n'a pas été capable d'utiliser avec succès ces fichiers générés automatiquement. Un script de conversion fonctionnel (de préférence écrit en un langage de script de haut niveau tel que Python) sera posté prochainement sur le wiki de GRASS dès qu'il sera complet ⁵⁶.

GMT fournit plusieurs outils pour créer manuellement un fichier de palette de couleur dans un domaine de données fourni par l'utilisateur, ou un domaine de données lu à partir d'un fichier grid de GMT. Bien que cela ne soit pas une approche idéale, il peut être utile quand une des palettes de couleur de GMT est appropriée. Dans le cas spécial où l'image exportée est en nuance de gris, la palette de couleur standard "gray.cpt" peut être utilisée pour créer un nouveau fichier cpt avec le programme `makecpt`.

```
# créer une palette de couleur fiable pour une image
# raster CELL en nuance de gris, les valeurs de grille
# sont des entiers dans un domaine compris de 0-255
makecpt -Cgray -T0/255/1 -V > doqq.cpt
```

Lorsque l'on travaille avec des canaux de couleurs qui ont déjà été séparés (par exemple, landsat ou

⁵⁵Les définitions des fichiers de palette de couleur de GMT peuvent être trouvées sur http://www.soest.hawaii.edu/GMT/gmt/doc/html/GMT_Docs/node57.html

⁵⁶De plus amples tests et adaptations du script `r.out.gmt.py` de David Finlayson pourraient être l'approche la plus rapide).

d'autres images multispectrales) il est possible d'utiliser `psimage` sans un fichier de palette de couleurs, au lieu de définir des fichiers de grilles en entrée en rouge, vert et bleu. Cette approche est plus aisée lorsque l'on travaille avec des données qui ont été collectées dans des bandes séparées. Pour des données à une seule bande, la séparation des couleurs peut être créée avec `r.mapcalc`.

obtenir les s\`eparations de couleurs pr\`et pour GMT

```
r.mapcalc "some_raster.red = r#some_raster"
r.mapcalc "some_raster.blue = b#some_raster"
r.mapcalc "some_raster.green = g#some_raster"
```

Exporter et tracer des données vecteurs GRASS

Le format vecteur de GMT est basé sur un format ASCII basé sur des vertex, très simple, désigné au début pour stocker des données géométriques. Cependant, avec des scripts il est possible d'encoder des données attributs simples dans ce format pour la production de cartes thématiques. Une approche alternative implique que les données vecteurs qui ont été pré-filtrées avec un outil GRASS tel que `v.extract`, en plusieurs fichiers où chacun représente une classe spécifique. En utilisant la première approche (encoder les attributs dans un fichier vecteur GMT) la symbologie pour chaque point, ligne et polygone est encodée dans le fichier, tandis que la symbologie est définie avec des paramètres en ligne de commande pour la seconde approche.

Données points

Exporter des données ponctuelles est la conversion la plus simple de GRASS → Vecteur GMT. Dans la plupart des cas, où un ensemble de points doit être tracé dans GMT en utilisant une symbologie fixe, la sortie de `v.out.ascii` ou `v.out.ascii.db` peut être directement tracée avec la commande GMT `texttpsxy`. Notez que filtrer la sortie avec l'une ou l'autre commande avec des outils de traitement de texte UNIX est une méthode idéale pour convertir des attributs dans une symbologie ou pour ajuster les étiquettes.

```
# exporter d'un format ASCII de GRASS
v.out.ascii in=points fs=" " > points.xy
```

```
# tracer les points avec un contour blanc et remplit de
# bleu en utilisant un symbol circulaire de 0.125
```

```
# pouces de diam\`etre en ajoutant le resultat Postscript
# \`a 'outfile.eps'
psxy points.xy -R -J -M -Sc0.125 -G255/1/1 \
-W1/255/255/255 -O -K >> output.eps
```

L'étiquetage des données points peut être réalisé d'abord en filtrant la sortie de `v.out.ascii` avec un simple script `awk`, puis en utilisant la sortie avec la commande `pstext`. Le format des étiquettes utilisé par GMT est documenté dans la page du manuel de `pstext`, cependant une simple explication est donnée dans l'échantillon de code ci-dessous.

```
# export du format de GRASS vers ASCII avec des
# attributs s\`electionn\`es (colonne 'ID'), sortie
# filtr\`e avec awk, classement des colonnes,
# et insertion des \`etiquettes dans le formulaire :
# x_coord, y_coord, font_size, rotation_angle,
# font_number, label_offset, label_text
v.out.ascii.db in=points columns=ID | awk -F"| "
'{print $2, $3, 10, 0, 4, "BL", $4}' \
> points_with_labels.xy
```

```
# trace les textes des \`etiquettes \`a partir des
# g\`eom\`etries et des propri\`et\`es des \`etiquettes stock\`es
# dans 'points_with_labels.xy' les \`etiquettes seront
# bleu et distante de 0.1 cm dans les directions
# horizontale et verticales d\`efinie dans la colonne
# 5 du fichier en entr\`ee
pstext points_with_labels.xy -R -J -Dj0.1c/0.1c
-G0/0/255 -K -O >> outfile.eps
```

Données ligne et polygone

Il n'y a pas pour l'instant de méthode directe pour convertir des données lignes et polygone GRASS en un format compatible avec GMT. Cependant, en utilisant une application externe appelée `shp2gmt`⁵⁷, il est possible de créer des fichiers vecteurs compatibles avec GMT en exportant d'abord les données GRASS en shapefile puis en les convertissant avec `shp2gmt`. Le texte issu de `shp2gmt` contient tous les attributs stockés dans le shapefile en entrée en fonction des coordonnées du vertex, et peut être facilement filtré avec `awk` pour inclure les paramètres de la symbologie. Un exemple de session GRASS illustrant ce type d'opération est listé ci-dessous.

```
# export de GRASS vers le format shapefile
v.out.ogr -e in=lines dsn=. olayer=lines
v.out.ogr -e in=polys dsn=. olayer=polys
```

```
# convertit du format shapefile vers GMT
shp2gmt lines.shp > lines.xy
shp2gmt polys.shp > polys.xy
```

```
# en option filtre le fichier vecteur GMT
# pour inclure la symbologie
```

⁵⁷ce petit programme utile initialement écrit par (Frank Warmerdam), a été modifié par (Mark Fenbers) pour inclure des informations attributaires à partir d'une shapefile. Le code du `shp2gmt` modifié peut être trouvé à <http://www.arcknowledge.com/gmane.comp.gis.gmt.user/2004-01/msg00121.html>

```
awk '
{
# tous les segments lignes avec l'attribut
# 'some_value' sera dessin\`e avec un pinceau
# rouge
if ($0 ~ /some_value/) printf "> -Wred\n"
# ... ajoute plus d'am\`elioration pour les autres classes
# pour les lignes qui ne correspondent pas, pour les
# tracer mot pour mot (c.a.d. les donn\`ees vertex)
else print
}
' lines.xy > lines-thematic.xy

# traces les lignes, avec la symbologie d\`efinie dans
# le fichier vecteur GMT
psxy lines-thematic.xy -R -J -O -K >> outfile.eps
```

L'étiquetage des objets lignes et polygones suit la même approche générale que l'étiquetage des objets ponctuels, par l'utilisation de `pstext`. Pour de simples cartes, une approche presque automatique pour étiqueter les données polygonales implique l'export des centroïdes avec `v.out.ascii.db`, comme listé au-dessus. Étiqueter les données lignes nécessite plus d'actions de la part de l'utilisateur, puisque chaque étiquette nécessite des coordonnées et un angle associés. Avec un peu de travail il est possible d'utiliser le fichier créé par `v.label` pour produire un ensemble compatible avec GMT d'instruction de placement d'étiquettes. Un exemple de script `awk`, utilisé dans une session GRASS, est présenté ci-dessus.

```
# creer le fichier d\`etiquette GRASS :
# tourner les \`etiquettes pour correspondre aux segments
# de ligne
v.label -a map=trails column=trail_name labels=trails.lab

# converti le format d\`etiquette GRASS vers le format GMT
awk '
BEGIN{FS="\n" ; RS="\n\n"}
{
split($1, e, ": ")
split($2, n, ": ")
split($15, r, ": ")
split($16, l, ": ")
# s'il n'y a pas de rotation, nous devons ajouter une
# rotation de 0
if(NF == 16) {rotation = r[2] ; label = l[2]}
else {rotation = 0 ; label = r[2]}
# creation les instructions d\`etiquettage GMT
print e[2], n[2], 10, rotation, 4, label
}
' DATABASE/LOCATION/MAPSET/paint/labels/trails.lab
> labels.xy

# dessin avec pstext
pstext labels.xy -R -J -Dj0.1c/0.1c -G0/0/255 -K -O
>> outfile.eps
```

Bien que l'approche au-dessus peut fonctionner pour de simples cartes, une intervention de l'utilisateur est habituellement requise pour des cartes plus complexes. Une approche plus robuste pour l'étiquetage des objets ponctuels, lignes et polygones avec des possibilités avancées telles que la détection des collisions des étiquettes est nécessaire pour une approche entièrement automatisée.

Divers éléments de cartographie

Des éléments supplémentaires tels que le contexte de la carte (`psbasemap` avec `psxy`), l'échelle graphique (`psbasemap`), ou la légende (`pslegend`) peuvent être ajoutés après que les composants de la carte principale aient été ajoutés au fichier en sortie. Vérifier que la dernière commande GMT n'utilise pas le paramètre `-K` est toujours une bonne idée, afin de s'assurer que le fichier en sortie est finalisé. Par l'utilisation rigoureuse des paramètres `-O`, `-X`, et `-Y` il est possible de redéfinir la région du tracé pour chaque appel d'un programme GMT. Cette approche fonctionne bien pour créer une carte de situation de petite taille, dans une autre carte finalisée. En étendant le premier exemple, vous trouverez ci-dessous un script plus complet pour illustrer l'utilisation de `psbasemap` pour produire des éléments de la carte tels qu'une échelle graphique, une flèche indiquant le nord, en fonction de la création d'une carte de situation dans la carte principale.

```
# definition de param\etres globaux :
gmtset ANNOT_FONT_PRIMARY Times-Roman \
HEADER_FONT_SIZE 16 \
ANNOT_FONT_SIZE_PRIMARY 12 \
LABEL_FONT_SIZE 14 \
BASEMAP_TYPE plain \
PLOT_DEGREE_FORMAT DF \
PAPER_MEDIA letter+

# carte \echantillon centr\ee sur l'est des USA
pscoast -JB-116/36/30/42/7i \
-R-125/-108/31/44 -B5 \
-Gwhite -W0.5p \
-A250 -Dh -Na -Xc -Yc -P -K
> query_centers_2.eps

# trace les points mapxy :
psxy sample_data/mapxy_locations.latlong -J -R \
-Sc0.075c -W1/1/200/1 -G1/200/1 -O -K
>> query_centers_2.eps

# trace les points ka-map
psxy sample_data/ka-map_locations.latlong -J -R \
-Sc0.075c -W1/255/1/1 -G255/1/1 -O -K
>> query_centers_2.eps
```

```
# ajout de la barre d'\echelle
psbasemap -J -R -Lf-119/32.5/32/150k:"Kilometers": \
-O -K -P -V >> query_centers_2.eps

# cr\eer un contexte \`a la carte
pscoast -JB-116/36/30/42/7i \
-R-130/-100/25/50 \
-X0.25i -Y1.75i \
-Gwhite -W0.5p \
-A250 -Di -Na -P -K >> query_centers_2.eps

# ajoute la bo\^ite de la r\egion
psxy context_box.xy -M -JB -R -W3/1/1/0 -P -O
>> query_centers_2.eps
```

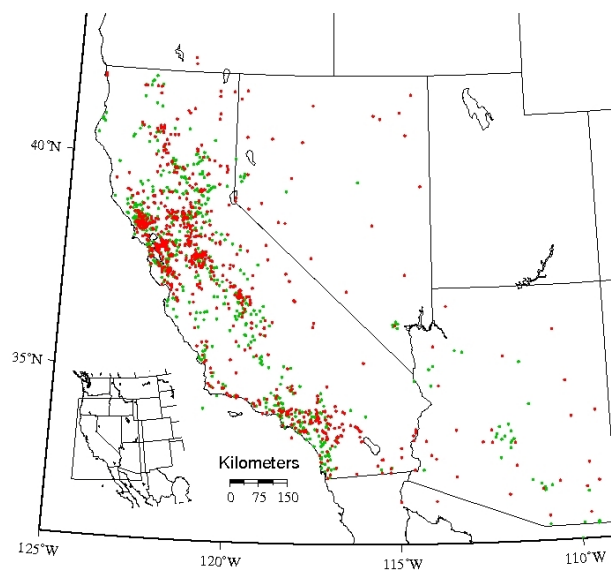


FIG. 2 – Ajout d'un contexte à la carte et d'une échelle graphique.

Limitations actuelles

L'approche actuelle par la console unix pour interfacer GRASS et GMT a plusieurs désavantages : manque de portabilité vers les systèmes non-unix, manque d'opérateurs mathématiques sophistiqués, et une dépendance avec des outils plus utiles tels que `awk`, `sed`, `grep`, etc. Ces désavantages sont particulièrement ennuyeux lors qu'on essaye d'automatiser la construction des éléments d'une carte tel que les intervalles ou les décorations basés sur l'échelle. Les exemples dans cet article utilisent des intervalles définis manuellement, cependant lors de la création de cartes à partir de GRASS (qui est habituellement dans un système de coordonnées projeté) cela prend du temps de rajouter les intervalles annotés ou non pour chaque carte. Regardez le script `template.sh` inclus dans l'archive des données d'échantillon pour avoir une idée. Encore une

fois, une intégration étroite avec GRASS via la liaison Python résoudrait le problème ci-dessus. En attendant qu'un développement entièrement en Python soit terminé, l'estimation des intervalles automatiques peut être simplifiée avec le script perl suivant.

```
#!/usr/bin/perl -w

# Le 1er argument est l'étendue maximale en unit\`e de la carte
# divis\`e par un nombre raisonnable d'annotations par bord
$x = $ARGV[0] / 10;

print round_up($x) , "\n";

# arrondit \`a une \`echelle raisonnable
# trouv\`e \`a :
# http://www.perlmonks.org/?node_id=599865

sub round_up {
    my $n = shift;
    my $scale = 10**int(log($n)/log(10));
    $n = 9 if $scale == 1; #magic for single digits
    if ($n > $scale) {
        $n = int($n/$scale+1)*$scale;
    }
    $n;
}
```

Conclusion

Cet article résume les conditions actuelles de conversion des données GRASS au format GMT pour la production de carte Postscript de grandes

qualités. Bien que plusieurs modèles et exemples existent pour automatiser ce processus, pour la production de cartes complexes de considérables modifications sont nécessaires par l'utilisateur. Après des progrès suffisants sur des projets comme l'interface de GRASS et la liaison Python, il pourrait être possible de créer un système plus complet pour améliorer le processus de GRASS → GMT Il y a actuellement un effort mené par Brent Wood pour réviser le format vecteur de GMT qui pourrait simplifier le processus de cartographie thématique avec GMT. De plus Brent est en train de travailler sur l'inclusion de la gestion en écriture pour le format vecteur GMT dans OGR (GDAL). Une fois terminé, l'export vectoriel de GRASS vers GMT pourrait être aussi simple que `v.out.ogr format=GMT`. Le travail proposé par Wolf Bergenheim, sur un algorithme général de détection et de correction de collision des étiquettes pour `v.label`, pourrait être une excellente solution pour les tâches d'étiquetage à la fois dans GRASS et GMT. Nous verrons la prochaine fois un exemple plus en profondeur de création d'une carte complexe en utilisant des données de GRASS à partir de l'échantillon de données Spearfish.

Dylan Beaudette

Université de Californie à Davis

<http://casoilresource.lawr.ucdavis.edu>
debeaudette AT ucdavis.edu

Utiliser l'interface R— GRASS

Status actuel

par Roger Bivand, traduit par Yves Jacolin

Introduction

Les interfaces entre GRASS et R, l'environnement open source de programmation de statistique et d'analyse de données, existent depuis quelques temps. Les détails de l'interface entre GRASS 6 et R ont été décrit 2 ans plus tôt dans Bivand (1), mais depuis les choses se sont beaucoup simplifiées.

Des fichiers temporaires intermédiaires sont la solution choisie pour l'interface de GRASS 6 : **spgrass6**, en utilisant des shapefiles pour les données vecteurs et des binaires BIL pour les données raster. R est lancé dans une session GRASS à partir de la ligne de commande, et **spgrass6** chargé avec ses dépendances, avec l'interface R utilisé pour accéder et mettre à jour les données GRASS.

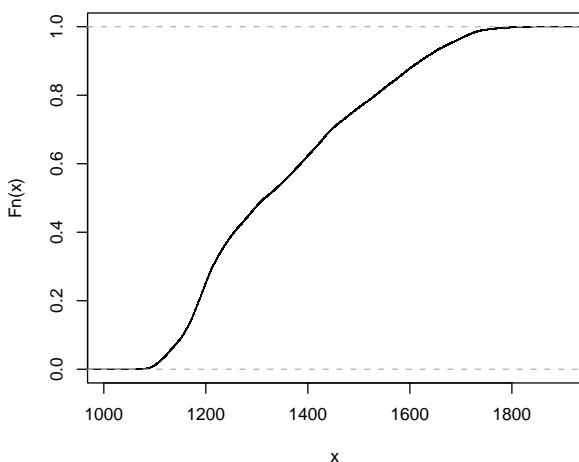


FIG. 1 – Fonction de distribution cumulative empirique de l'élévation de la région de Spearfish.

Installer le paquet de l'interface

L'interface GRASS 6 est disponible à partir de CRAN, Comprehensive R Archive Network. Elle dépend de trois paquets, et s'ils ne sont pas déjà disponibles, ceux-ci (**sp**, **maptools** et **rgdal**) doivent être installés dans R en utilisant l'argument `dependencies=` :

```
> install.packages("spgrass6", dependencies = TRUE)
```

Pour installer sur un serveur qui ne lance pas d'interface graphique, définissez d'abord le miroir CRAN avec :

```
> chooseCRANmirror(graphics = FALSE)
```

Les seules difficultés potentielles pour l'installation de ces paquets à partir des sources sur Linux, Unix ou Mac OS X concernent **rgdal**, à cause de sa dépendance externe avec les bibliothèques GDAL et PROJ.4. Sur Linux/Unix, notez que les fichiers de développement de GDAL sont nécessaires, pas seulement GDAL lui-même, si votre version de GDAL a été installée avec des binaires plutôt qu'à partir des sources. Tous les autres paquets sont disponibles comme binaires pour les utilisateurs de Mac OS X, mais **rgdal** ne l'est pas. Des notes pour les utilisateurs de Mac OS X sur l'installation de **rgdal** peuvent être trouvées sur le site [Rgeo](#) — voyez la partie **rgdal**. Les binaires Windows sont disponibles pour tous les paquets, et fonctionnent avec GRASS 6 sous Cygwin.

Utiliser le paquetage

```
> library(spgrass6)
> gmeta6()
```

Les exemples utilisés ici proviennent de la location des données échantillons "Spearfish" (Dakota du sud, USA, 103.86W, 44.49N), peut-être les plus typiques pour les démonstrations de GRASS. La fonction `gmeta6` est simplement une manière de résumer les paramètres actuels de la localisation et la région de GRASS dans laquelle nous sommes en train de travailler. À ce niveau de l'interface, le transfert des données raster est effectué couche par couche, et utilise des fichiers binaires temporaires. La commande `readRAST6` lit ici des valeurs d'élévation dans un objet `SpatialGridDataFrame`, traitant les valeurs retournées comme des flottants, et la couche spécifique de géologie dans un facteur :

```
> spear <- readRAST6(c("elevation.dem",
+ "geology"), cat = c(FALSE, TRUE))
```

```
> summary(spear)
```

```
Object of class SpatialGridDataFrame
Coordinates:
```

```

      min      max
coords.x1 589980 609000
coords.x2 4913700 4928010
Is projected: TRUE
proj4string :
[+proj=utm +zone=13 +a=6378206.4
+rf=294.9786982 +no_defs
+nadgrids=/home/rsb/topics/grass63/grass-6.3.cvs
```

```

/etc/nad/conus
+to_meter=1.0]
Number of points: 2
Grid attributes:
  cellcentre.offset cellsize cells.dim
1          589995      30      634
2          4913715     30      477
Data attributes:
  elevation.dem      geology
Min.   : 1066      sandstone:74959
1st Qu.: 1200      limestone:61355
Median : 1316      shale   :46423
Mean   : 1354      sand    :36561
3rd Qu.: 1488      igneous :36534
Max.   : 1840      (Other) :37636
NA's   :10101     NA's    : 8950

```

Quand l'argument `cat=` est défini à `TRUE`, les étiquettes de catégorie de GRASS sont importées et utilisées comme niveaux de facteur ; En vérifiant a posteriori, nous pouvons voir qu'ils concordent :

```

> table(spear$geology)
metamorphic transition igneous
 11693          142    36534
 sandstone limestone  shale
 74959      61355    46423
sandy shale claysand sand
 11266      14535    36561

> system("r.stats --q -cl geology",
+ intern = TRUE)

[1] "1 metamorphic 11693"
[2] "2 transition 142"
[3] "3 igneous 36534"
[4] "4 sandstone 74959"
[5] "5 limestone 61355"
[6] "6 shale 46423"
[7] "7 sandy shale 11266"
[8] "8 claysand 14535"
[9] "9 sand 36561"
[10] "* no data 8950"

```

La figure 1 montre un graphique de distribution cumulative empirique des valeurs d'élévation, permettant la lecture de la proportion de l'aire d'étude pour des élévations choisies. À son tour la figure 2 montre un simple graphique d'élévation des élévations par catégorie dans la surface totale. Nous avons utilisé la fonction `readRAST6` pour lire à partir de GRASS vers R ; la fonction `writeRAST6` permet à une seule colonne nommée d'un objet `SpatialGridDataFrame` d'être exportée vers GRASS.

Le paquet `spgrass6` fournit également des fonctions pour déplacer des objets vecteurs et les données attributaires associées vers R et les récupérer. La fonction `readVECT6` est habituellement utilisée pour importer des données vecteurs dans R, et `writeVECT6` pour exporter vers GRASS.

```

> bugsDF <- readVECT6("bugsites")
> vInfo("streams")
  points    lines boundaries centroids
    0      104         12          4
  areas    islands    faces    kernels
    4         4          0          0

> streams <- readVECT6("streams", type = "line,boundary",
+   remove.duplicates = FALSE)

```

L'argument `remove.duplicates=` est défini à `TRUE` quand il y a seulement des lignes ou des surfaces par exemple, et que le nombre présent est plus grand que le nombre de données (le nombre de lignes dans la table de données attributaires).

L'argument `type=` est habituellement utilisé pour annuler le type de détection lorsque de multiples types sont non nuls, comme ici, où nous choisissons des lignes et des contours, mais la fonction suppose des surfaces, retournant seulement les masses d'eau remplies.

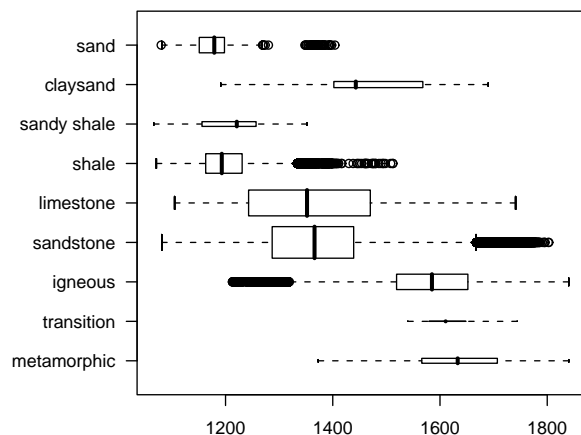


FIG. 2 – Graphiques d'élévation en fonction de la catégorie de la géologie, région de Spearfish.

À cause du mécanisme utilisé pour passer des informations concernant les coordonnées de la région de GRASS le système de référence diffère sensiblement entre les données raster et vecteurs, la chaîne PROJ4 diffère souvent sensiblement, alors même que le CRS réel est le même. Nous pouvons voir que la représentation pour les localisations des points des sites des scarabées diffèrent ici ; la représentation du vecteur est plus en accord avec la notation des standards de PROJ4 que celle pour les couches rasters, alors même qu'ils sont les mêmes. Dans le résumé de l'objet `spear` ci-dessus, l'ellipsoïde a été représenté par les balises `+a=` et `+rf=` au lieu de la balise `+ellps=` en utilisant la valeur `clrk66` :


```
> summary(bugsDF)
Object of class SpatialPointsDataFrame
Coordinates:
      min      max
coords.x1 590232 608471
coords.x2 4914096 4920512
Is projected: TRUE
proj4string :
[+proj=utm +zone=13 +ellps=clrk66
+datum=NAD27 +units=m +no_defs
+nadgrids=@conus,@alaska,@ntv2_0.gsb,@ntv1_can.dat]
Number of points: 90
Data attributes:
      cat      str1
Min.   : 1.00  Beetle site:90
1st Qu.:23.25
Median :45.50
Mean   :45.50
3rd Qu.:67.75
Max.   :90.00
```

Cela nécessite des modifications manuelles d'une représentation à l'autre, et cela est dû au fait que GRASS utilise des extensions non standards mais équivalentes à PROJ4. Il y a de nombreuses fonctions utiles dans le paquet **spgrass6**, dont `gmeta2grd` qui permet de générer un objet `GridTopology` à partir des paramètres de région en cours dans GRASS. Cela est typiquement utilisé pour l'interpolation à partir de données ponctuelles vers une grille raster, et peut être masqué par coercion d'un objet `SpatialGrid` vers un objet `SpatialPixels` ayant un jeu de cellules en dehors de la zone d'étude vers NA.

Une seconde fonction utile pour les données vectorielles utilise le fait que GRASS 6 utilise un modèle de données vectoriel topologique.

La fonction `vect2neigh` retourne un cadre de données avec les voisins de gauche et de droite de chaque arc des contours des polygones, avec la longueur de ces arcs.

Cela peut être utilisé pour modifier le poids des contiguités des polygones basé sur la longueur des limites communes. Comme GRASS, GDAL/OGR, PROJ4 et d'autres projets de l'OSGeo, les fonctions offertes par **spgrass6** changent, et les pages d'aide doivent être consultées pour vérifier les usages correctes.

Bibliographie

- [1] Bivand, R. S., (2005) Interfacing GRASS 6 and R. *GRASS Newsletter*, 3, 11–16, <http://grass.itc.it/newsletter/>.

Roger Bivand
 Section Géographie Économique, Département d'Économies,
 Norwegian School of Economics and Business Administration,
 Bergen, Norvège
<http://www.r-project.org/Rgeo>
 Roger.Bivand AT nhh.no

Traitement géospatial sur des serveurs distants via Internet – PyWPS

PyWPS et Embrio

Jáchym Čepický et Lorenzo Becchi, traduit par Vincent Picavet

Le document 05-007r4⁵⁸ de l'OGC décrit un procédé pour mettre à disposition des opérations géospatiales à travers des réseaux en utilisant des services Web. Cet article présente une implémentation : PyWPS. Même si la cible initiale de PyWPS était de rendre accessible depuis des applications Web les modules du SIG GRASS, en pratique il est possible d'utiliser n'importe quel outil en ligne de commande ou qui possède une interface avec le langage de programmation Python. Avec l'aide de WPS il est possible de réaliser des opérations coûteuses en temps de calcul côté serveur, ou de construire votre propre application de Web-SIG tournant dans un navigateur Web. Décrivons maintenant la façon dont PyWPS fonctionne et déterminons si il peut répondre à vos besoins.

OGC Web Processing Service

Le standard OGC WPS (Web Processing Service) est relativement jeune et pas encore aussi connu que son cousin le WMS (Web Map Service). Nous commencerons donc par en donner un bref aperçu. L'unité de base du WPS est le *process*, une opération géospatiale, avec des entrées et des sorties de type défini. Le client communique avec le serveur à l'aide de trois types de requêtes. La requête peut être envoyée au serveur par HTTP GET avec des paramètres donnés sous la forme de paires clé-valeur (Key-Value Pairs, KVP) ou par HTTP POST, avec des paramètres données dans un fichier XML. Les trois types de requêtes pouvant être envoyées au serveur sont :

GetCapabilities – Le serveur répond en XML, décrivant le serveur, les coûts, une description générale et donnant une liste de process, prêts à être exécutés.

DescribeProcess – Le serveur répond en XML, décrivant en détail les types des entrées et des sor-

ties, pour que le client soit capable d'écrire la requête *Execute*.

Execute – Le client demande l'exécution d'une opération géospatiale, avec toutes les données nécessaires. Le serveur lance le process et informe le client (l'utilisateur) de l'avancement.

La suite donne quelques exemples pour illustrer l'utilisation de ces requêtes. Disons que nous voulons faire un calcul de ligne de visée à partir de coordonnées *x* et *y* sur un fichier raster, disponible à partir d'un serveur distant. Le nom du process sera *visibility*.

Utilisation basique de l'OGC WPS

Tout d'abord nous avons besoin de déterminer quelles opérations le serveur propose (voir ce lien⁵⁹). À partir du XML renvoyé, il apparaît clairement que le process *visibility* est disponible sur le serveur, et le résumé nous dit qu'il correspond au service que nous recherchons. Ensuite, nous avons besoin de savoir quel type d'entrée et de sortie le process nécessite et renvoie. (voir ce lien⁶⁰).

- *x* de type *LiteralValue*
- *y* de type *LiteralValue*
- *maxdist* – distance maximale à l'observateur. De type *LiteralValue*, valeur minimale 0, maximale 5000 mètres.
- *observer* – hauteur de l'observateur. De type *LiteralValue*, valeur minimale 0, maximale 50 mètres.
- *dem* – de type *ComplexValue* – carte raster de modèle numérique de terrain, sur laquelle la ligne de visée doit être calculée.

Maintenant nous pouvons écrire la requête d'exécution, l'envoyer au serveur et attendre les résultats de calcul⁶¹. Le serveur va télécharger le modèle numérique de terrain, faire le calcul de ligne de visée et renvoyer le raster de résultat au client.

⁵⁸OGC Web Processing Service (WPS) : <http://www.opengeospatial.org/standards/requests/28>

⁵⁹<http://pywps.ominiverdi.org/cgi-bin/wps.py?service=WPS&request=GetCapabilities>

⁶⁰<http://pywps.ominiverdi.org/cgi-bin/wps.py?service=WPS&request=DescribeProcess&version=0.4.0&identifiant=visibility>

⁶¹<http://pywps.ominiverdi.org/cgi-bin/wps.py?service=WPS&version=0.4.0&request=Execute&identifiant=visibility&datainputs=x,602829.1875,y,4925326.875,maxdist=2000,observer=1.2,dem,http://somewhere/?some&service>

Introduction à PyWPS

PyWPS est un projet relativement jeune, commencé en avril 2006. Le but premier du projet était de faire la connexion entre UMN-MapServer et le SIG GRASS de la manière la plus simple possible, afin de pouvoir créer une véritable application WebSIG, capable d'effectuer par exemple de l'interpolation de données raster, ou diverses analyses de modèles numériques de terrain. Avec le temps, il s'est avéré que même si GRASS est un outil puissant, il n'est pas forcément le meilleur ou le seul choix possible pour toutes les tâches. La conception de PyWPS a changé de telle sorte qu'il puisse être utilisé sans GRASS en arrière plan, mais avec n'importe quel autre outil, ou même juste avec Python. PyWPS est une implémentation du standard WPS de l'OGC comme défini dans le document OGC 05-007r4. Actuellement, le support de tout le standard n'est pas complet, mais presque 95% de la norme est implémentée et utilisable.

Le projet est construit sur un simple script CGI, afin de rendre la vie des développeurs de WebSIG la plus simple possible. Il met à disposition les fonctionnalités pour :

- Analyser toutes les entrées et créer toutes les sorties,
- Réaliser des validations basiques sur les entrées, comme vérifier le type d'entrées Literal-Value ou la taille maximale de fichier pour une entrée de type ComplexValue, etc.
- Créer et supprimer à la volée des fichiers et répertoires temporaires, comme des emplacements et mapsets GRASS, ou d'autres fichiers créés pour pouvoir effectuer le calcul.
- Lancer d'autres opérations utiles.

Le développeur n'a qu'une seule chose à faire : définir les entrées et sorties, ce qui est simplement un script en langage Python. Le process est une classe *Process*, avec une méthode requise nommée *execute*, dans laquelle le calcul est effectué. Les données d'entrée et de sortie sont définies d'une manière similaire, dans une structure de dictionnaire complexe⁶² :

```
{
  'Identifiant': 'maxdist',
  'Title': 'Maximal distance',
  'Abstract': 'Maximal distance of visibility',
  'LiteralData': {
    'values': [ [0., 5000] ],
  },
  'dataType': type(0.0),
}
```

⁶²Ceci a été remplacé par des appels de méthodes *Add*Input()* dans la version *svn* actuelle

```
},
{
  'Identifiant': 'dem',
  'Title': 'Digital elevation mode'
  'Abstract': 'Raster map with elevation model',
  'ComplexValueReference': {
    'Formats': ["image/tiff"],
  }
},
},
```

La méthode *execute()* peut utiliser par exemple directement des modules de GRASS :

```
def execute(self):
  # importation de dem
  os.system("r.in.gdal in=%s out=dem" %\
            (self.datainputs['dem']))
  # r\'eclage de la r\'egion \`a partir du fichier dem
  file os.system("g.region rast=dem")
  # module de ligne de vis\ee
  os.system("r.los input=dem output=output \
            coordinate=%s,%s max_dist=%f \
            obs_elev=%d" % \
            (self.datainputs['x'],
             self.datainputs['y'],
             self.datainputs['maxdist'],
             self.datainputs['observer']))
  # export du raster de sortie
  os.system("r.out.gdal in=output out=out.tif")
  # r\'eclage de la valeur de retour
  self.dataoutputs['output'] = "out.tif"
  return
```

Comme le code source est la meilleure documentation, environ dix exemples de process sont distribués avec le code source de PyWPS, pour que l'utilisateur puisse avoir une idée générale de la définition d'un process. Il y a aussi une documentation disponible en ligne comme hors ligne, qui tente de décrire le mieux possible le processus d'installation et la mise en place de vos propres process.

Récemment, une nouvelle classe a été définie, qui permet une définition facile des entrées et sorties de process, ainsi qu'un meilleur support pour les modules de GRASS. Toute interface Web sera désormais capable de suivre l'avancement des imports dérivés directement du module *r.in.gdal*. Cette amélioration sera disponible dans la prochaine version de PyWPS.

Avancées du développement

La première version "stable" de PyWPS avec le numéro de version 1.0.0 est sortie en Novembre 2006. Actuellement, l'équipe de développement de PyWPS

projette de sortir la version 2.0.0 prochainement, avec de nouvelles fonctionnalités et quelques corrections de bogues. L'effort général de développement est effectué dans trois directions :

- Implémentation du standard OGC WPS complet,
- Sécuriser l'application au maximum afin d'éviter la compromission de serveur,
- Rendre la vie du développeur de process la plus simple possible.

L'équipe de développement de PyWPS voudrait également engager des discussions avec la communauté géospatiale pour définir des métadonnées de process. Le standard OGC WPS définit les types d'entrée d'un point de vue du process, mais n'aborde pas le sujet des types d'entrée (et de sortie) d'un point de vue utilisateur (c'est à dire de l'interface). Par exemple, la coordonnée x est de type *LiteralValue*, mais cela ne définit pas x comme une coordonnée. Il serait cependant utile de définir cette entrée avec un clic de souris sur une carte, plutôt qu'en tapant au clavier dans un champ de texte. Si vous voulez jeter un oeil de plus près à PyWPS, vous pouvez visiter la page du projet⁶³, où le code source est disponible, ainsi que des liens vers des projets utilisant déjà PyWPS.

Utiliser ka-Map & PyWPS pour créer un WebSIG GRASS

Ominiverdi⁶⁴ a joint le développement de PyWPS après la première version. Notre but initial était de créer un client Web libre (FOSS : Free Open Source Software) pour accéder aux fonctions de GRASS. PyWPS n'était pas la première tentative de créer une connexion entre GRASS et le Web, mais aucune autre n'était basée sur des standards ouverts, rendant impossible la création d'une plateforme partagée. Dès le départ nous voulions travailler à deux différents projets : **Embrio** et **Wuiw**.

Embrio

Embrio est la première implémentation que nous avons à l'esprit : utiliser PyWPS pour faire interagir UMN MAPSERVER et l'API MapScript avec GRASS.

Un autre but important était un client web riche, et c'est pourquoi nous avons décidé de baser nos efforts sur ka-Map. Ka-Map utilise Mapscript pour ac-

céder à l'ensemble de puissantes fonctionnalités de UMN Mapserver et offrir une expérience de navigation à la Google Maps. La sortie de PyWPS, une fois que le process retourne une carte, peut être en GeoTiff, pour une sortie raster, ou en GML pour une sortie vecteur. Les deux formats sont accessibles nativement par UMN MapServer et peuvent être facilement couplés avec d'autres environnements cartographiques. Ainsi, une requête au module *Visibility* peut retourner un GeoTiff, qui est gardé par ka-Map et inséré dans le vrai système de coordonnées de la carte. Ensuite un style, qui peut être un SLD, est appliqué aux valeurs du raster, et un cache temporaire est créé à la volée pendant que les tuiles sont renvoyées au client. Le cache temporaire est lié au *sessionId*.

Ce système permet à différents modules de s'exécuter en parallèle et de comparer leurs sorties en utilisant l'interface de ka-Map. Le contrôle de la transparence et de la position du layer peuvent être utiles pour comprendre le résultat final. La fonctionnalité de requête est également synchronisée si la sortie est requêteable. Le module *Visibility* peut produire l'angle incident avec le point de vue, et la fonction de requête peut retourner la valeur de chaque pixel cliqué.

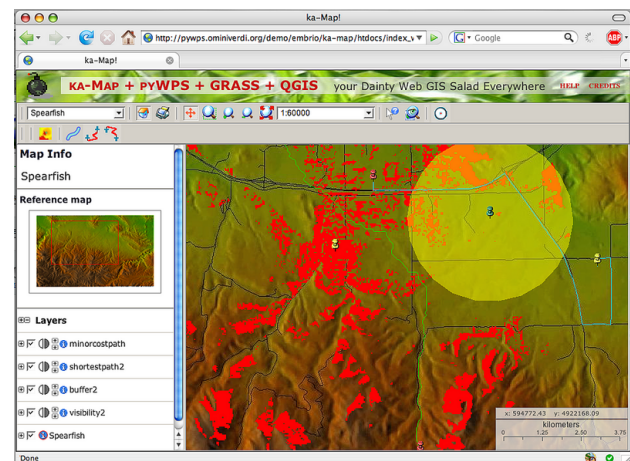


FIG. 1 – Copie d'écran de l'interface Embrio la plus récente.

Wuiw

Wuiw est encore aujourd'hui plus un concept qu'une réelle application. Il a pour objectif de fournir une API Javascript qui connecte le service WPS à des sources de données en OWS, et de faire un rendu

⁶³Page du projet PyWPS : <http://pywps.wald.intevation.org>

⁶⁴Site d'Ominiverdi : <http://www.ominiverdi.org>

de la sortie indépendamment d'une quelconque application de serveur cartographique.

Nous sommes encore en train d'évaluer les limites actuelles du brouillon WPS, qui ne relie pas encore les définitions des entrées avec la définition d'un type complexe. La première idée était d'utiliser des informations en métadonnées pour créer ce type de relation, mais le risque existe de créer un dialecte parallèle qui viendrait amoindrir les avantages de l'interopérabilité par les standards. Nous espérons que le chemin de WPS vers une version 1.0 apportera une solution adaptée à cette limitation.

Développements futurs

Il est clair que WPS, PyWPS, Embrio et Wuiw ont tous une histoire courte. Même si la plupart des choses sont encore à faire, le début est prometteur. Concernant Embrio, nous prévoyons d'améliorer l'interaction avec les retours des scripts de process WPS afin d'afficher une barre de progression. Beaucoup d'autres modules de GRASS peuvent être développés, et nous espérons recevoir de l'aide de la communauté GRASS pour les réaliser.

Un autre but objectif important est d'ajouter plus d'interaction avec les applications en ligne de commande (CLI : Command Line Interface), comme R, GDAL/OGR, etc. pour réaliser des géostatistiques, des conversions de format, et beaucoup, beaucoup

d'autres fonctionnalités.

Nous espérons aussi ajouter une ligne de commande en AJAX pour interagir avec un système protégé utilisant WPS.

Licenses

Il est important de noter que ce projet utilise beaucoup de logiciels avec au moins deux licences différentes :

- GNU/GPL : GRASS, PyWPS, R
- MIT/BSD : UMN MapServer, MapScript, ka-Map, Embrio

Références

- OGC Web Processing Service⁶⁵
- Site web de PyWPS :⁶⁶
- Dernière démonstration d'embrio :⁶⁷
- Site web d'Embrio⁶⁸

Jáchym Čepický

<http://les-ejk.cz>

jachym AT les-ejk cz

Lorenzo Becchi

<http://ominiverdi.org>

lorenzo AT ominiverdi com

⁶⁵<http://www.opengeospatial.org/standards/requests/28>

⁶⁶PyWPS Home page : <http://pywps.wald.intevation.org>

⁶⁷Démonstration :

http://pywps.ominiverdi.org/demo/embrio/ka-map/htdocs/index_wps_qgis.html

⁶⁸<http://pywps.ominiverdi.org/>

Tikiwiki un GeoCMS

par Franck Martin, traduit par Marie Silvestre

Vue d'ensemble

Vous avez peut-être entendu parlé de Content Management Systems (CMS) ou Systèmes de Gestion de Contenus (SGC), il s'agit d'applications web qui permettent de partager du contenu via internet. Le concept a évolué en mélangeant les wikis avec les sites internet basés sur un forum. Un wiki ⁶⁹ est un site internet permettant la création rapide de pages utilisant un langage simple de formatage. Les liens vers les nouvelles pages sont créés en premier puis les pages elles-mêmes. A partir de là, des forums, des galeries d'images et des blogs ont été ajoutés. Tikiwiki ⁷⁰ a été créé comme un CMS intégré dès le début, l'approche est plus robuste, et permet à toutes les fonctionnalités d'interagir entre elles. Dans Tikiwiki, vous avez un wiki, une galerie d'images, une galerie de fichiers, des blogs, des forums, des articles, des trackers et bien d'autres choses encore, le tout lié via un système de gestion des utilisateurs et des groupes pour définir les permissions sur les objets. Tikiwiki ne dépend pas d'un format de base de données spécifique, utilisant *PHP ADODB*⁷¹ et *Smarty*⁷², un moteur de template pour PHP. Il est alors facile de personnaliser l'apparence et le ressenti de Tikiwiki de façon à ce que votre site internet ne ressemble pas au site Tikiwiki par défaut.

Qu'est-ce que Tikiwiki exactement ?

Tikiwiki est un des premiers CMS développés, il est hautement configurable mais souffre d'un code de base long et d'un manque de module. Cependant ce désavantage est aussi un avantage si vous voulez déployer toutes les fonctionnalités dans un CMS car elles sont directement intégrées plutôt que dans des modules indépendants. Tikiwiki peut également être étendu via *mods*, un système de plug-in qui permet d'étendre les fonctions. J'ai choisi Tikiwiki comme CMS parce qu'il a été mis au rang de projet du mois sur Sourceforge et également parce que la communauté de développeurs est très ouverte. Il a été très facile d'ajouter du code au code de base. C'est important lorsque vous commencez à contribuer à un

⁶⁹Définition d'un wiki : <http://en.wikipedia.org/wiki/Wiki>

⁷⁰Site internet de Tikiwiki : <http://tikiwiki.org>

⁷¹Le projet PHP ADODB : <http://adodb.sourceforge.net/>

⁷²moteur de template Smarty : <http://smarty.php.net/>

projet.

Tikiwiki a un système de sécurité étendu et les développements actuels se focalisent sur l'implémentation d'*AJAX* dans le code pour le rendre plus efficace. L'autre point important de Tikiwiki est l'internationalisation. Il y a plusieurs langues pour l'interface et il est aisé d'ajouter votre propre langue avec l'équivalent php de *gettext*. Aussi, Tikiwiki supporte l'encodage en UTF8 des caractères et les écritures de droite à gauche. Vous pouvez aussi bien mélanger plusieurs langues dans la même page que faire une traduction intégrale des pages du wiki. Enfin, Tikiwiki est hautement personnalisable via le CSS et les templates *Smarty*.

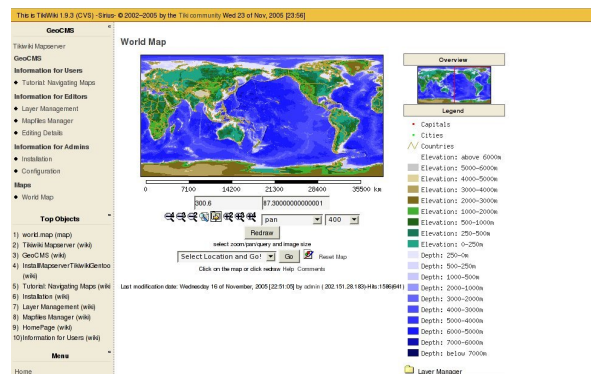


FIG. 1 – Une carte indépendante dans Tikiwiki.

Un GeoCMS, qu'est-ce que c'est ?

Depuis le succès de Google Maps, les cartes sont devenues une partie importante de tout site internet. Le succès de Google Maps est dû à la publication de son API, permettant à n'importe quel utilisateur d'ajouter des données ponctuelles à une carte Google, comme la localisation d'un voyage. Les cartes sont disponibles depuis longtemps sur internet, comme avec *Mapquest*, mais elles n'étaient pas personnalisables par le grand public. Un *Geospatial Content Management System* (GeoCMS) est un CMS avec des objets géographiques et des capacités de rendu cartographique. En plus d'avoir sur votre site internet une carte interactive en ligne sur laquelle vous pouvez zoomer, recadrer, interroger les objets, vous avez des objets géographiques que vous pou-

vez représenter. Ces cartes peuvent être indépendantes (voir fig. 1) ou incluses dans une page du wiki (voir fig. 2). Par exemple, vous pouvez demander à vos utilisateurs authentifiés d'ajouter leur localisation et les voir représentés par un point sur la carte. Vous pouvez créer une galerie d'images où chaque image à une localisation et les voir représentées par un point sur la carte, pour montrer votre dernier voyage, des monuments importants, des entités géologiques, etc.

Avec toute couche géographique, se pose le problème d'en savoir suffisamment sur la couche (métadonnées) : qui a créé la couche, quand, quelle est la qualité de la donnée, dans quel système de projection les objets sont-ils représentés, etc. Il est important d'avoir ces informations pour établir un niveau de qualité des données. Si vous avez une couche de population et qu'une catastrophe frappe une région spécifique, vous devez avoir une certaine qualité de données pour préparer la réponse. Lier des couches sur la carte avec une page du wiki est possible dans Tikiwiki. Les métadonnées sont de fait faciles à entrer et le spécialiste peut décrire les caractéristiques intéressantes représentées sur la carte.

Tikiwiki fournit également un module complet de chargement et traite les couches géographiques pour les rassembler sur les cartes via les mapfiles de MapServer. Un historique est maintenu pour chaque mapfile afin de garder une trace des changements et, optionnellement, vous prévient de tels changements.

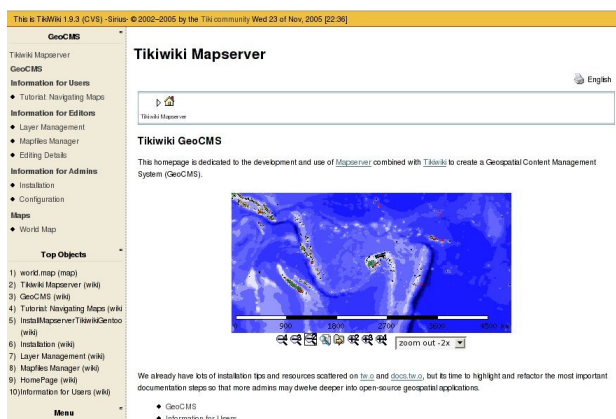


FIG. 2 – Une carte incluse dans une page du wiki.

Installation et configuration

La fonctionnalité de cartographie de Tikiwiki a pu être créée grâce à la publication de l'API MapSer-

ver⁷³ pour PHP. Le moteur de MapServer est appelé dans Tikiwiki pour créer des rendus image pour l'interface de Tikiwiki.

Je ne vais pas passer trop de temps sur l'installation et la configuration mais résumer rapidement car le sujet de l'article est de vous donner envie de l'installer, donc je me concentrerai sur la description de ses fonctionnalités.

L'installation de Tikiwiki est assez linéaire et la documentation complète sur l'installation est disponible en ligne⁷⁴ pour différents systèmes d'exploitation incluant MS-Windows. Il y a également quelques paquets disponibles pour différentes versions de Linux. L'installation de Tikiwiki peut être résumée ainsi : décompressage et remplissage d'un formulaire qui dirige Tikiwiki vers une base de données vide et déjà créée. L'installation de l'extension *php-mapscript* du code de MapServer est un peu plus compliquée car vous devez compiler *MapServer* avec toutes ses dépendances et enregistrer l'extension avec PHP. Vous allez également devoir créer un répertoire accessible pour Apache et protégé de manière adéquate pour stocker les mapfiles, les couches géographiques et les images générées. Il y aura également quelques paquets disponibles avec quelques exemples de données, notamment pour Mandriva. Pour cette installation spécifique, la documentation peut-être trouvée ici : <http://doc.tikiwiki.org/tiki-index.php?page=Maps+Install>

Créer des cartes

Tikiwiki utilise le moteur sous-jacent MapServer, ce qui signifie qu'il utilise le format mapfile. Ce mapfile, avec l'extension *.map*, crée un espace de travail web à utiliser dans l'application. Il existe un éditeur de texte dans Tikiwiki pour éditer le mapfile. Il utilise également des balises pré-définies pour compléter automatiquement certaines parties du mapfile (par exemple une couche template). Il y a plusieurs sections ou objets dans un mapfile. L'objet général est l'objet carte qui commence par MAP et termine par END. A l'intérieur de cet objet on trouve les objets REFERENCE, LEGEND, WEB, SCALEBAR et QUERYMAP. Ils définissent la présentation générale de la carte générée, la carte de situation (REFERENCE), l'échelle (SCALEBAR), la légende (LEGEND) et les objets surlignés par une requête (QUERYMAP). Dans l'objet MAP, vous pouvez insérer des objets LAYER (couche). Il y a principalement deux

⁷³Site internet de MapServer : <http://mapserver.gis.umn.edu/>

⁷⁴Documents d'installation de Tikiwiki : <http://doc.tikiwiki.org/>

types de LAYERS : raster et vecteur. L'utilisation de la librairie GDAL (prononcer *goodle* en Anglais) pour les raster et OGR (prononcer *ogre* en anglais) pour les vecteurs permet d'écrire et de lire beaucoup de formats de fichiers de Système d'Information Géographiques (SIG). Le format le plus utilisé pour les rasters est le GeoTIFF et les formats shapefiles ou les fichiers MapInfo TAB pour les vecteurs. A l'intérieur d'un objet LAYER, vous définissez un ou plusieurs objets CLASS pour spécifier la couleur et le type d'entités dessinées sur la carte. Vous pouvez également ajouter un objet LABEL si vous souhaitez ajouter des étiquettes aux entités. Un mapfile typique peut être trouvé sur <http://doc.tikiwiki.org/Maps+Mapfile+Tutorial>.

Chaque couche (objet LAYER) pointe sur un fichier SIG qui est géré par une interface de gestion des couches vous permettant de charger des fichiers SIG à utiliser dans le mapfile dans Tikiwiki.

Tikiwiki tire profit des commentaires et de l'objet METADATA du mapfile. Tikiwiki place un entête dans le mapfile qui permet de garder une trace des révisions entre chaque édition du mapfile, un historique est également créé.

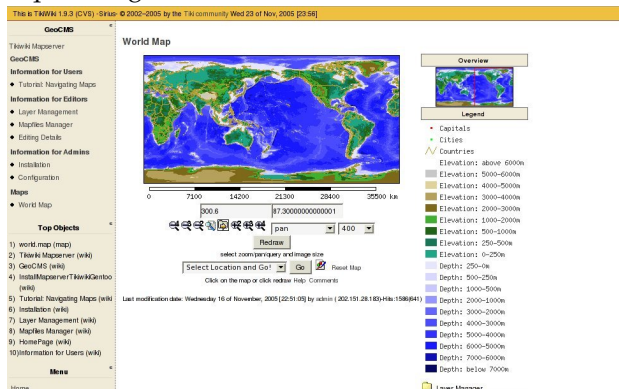


FIG. 3 – Une table montrant les informations relatives aux objets requêtés.

Le mot clé WIKI de l'objet METADATA est utilisé pour faire le lien entre une couche et une page du wiki, vous permettant de fournir des explications sur une couche spécifique. Ce lien est créé dans le gestionnaire de couches sur l'interface de la carte. Le mot clé DOWNLOAD de l'objet METADATA est également utilisé pour télécharger les fichiers SIG composant la couche. Tikiwiki est donc un outil de publication de données géographiques.

Enfin, le mot clé VIEW de l'objet METADATA est utilisé pour créer des vues prédéfinies de la carte. Les

utilisateurs peuvent donc zoomer sur le Pacifique ou sur l'Europe ou sur n'importe quel autre endroit intéressant.

Le système permet l'inclusion de beaucoup de couches mais Tikiwiki peut lui-même créer quelques couches spécifiques. Par exemple une couche contenant la localisation de tous les utilisateurs enregistrés qui ont rentré une localisation valide peut être créée. De même, une galerie d'images peut être déclarée de type géographique et chaque image aura une position. Une couche issue de toutes ces images peut être créée et ajoutée à n'importe quel mapfile en tant que couche de points sur la carte. Lorsque les utilisateurs interrogent un de ces points, une table contenant les informations à propos de l'image et l'image elle-même peuvent être affichées juste en dessous de la carte (voir fig. 3).

Un plugin wiki a été développé, il inclut n'importe quelle carte dans une page wiki et met en oeuvre quelques fonctionnalités basiques telles que le zoom et le déplacement panoramique. A son niveau le plus simple, une image générée peut-être insérée dans une page web, et donc une page wiki, dès lors que les paramètres pour générer l'image de la carte sont présents dans l'URL. La méthode GET est utilisée plutôt que POST pour générer les cartes.

La nouvelle interface de visualisation

J'ai travaillé sur l'implémentation d'AJAX dans l'interface de visualisation standard via la librairie *xajax*⁷⁵. Je propose également une nouvelle interface basée sur le code de Ka-Map⁷⁶. J'ai intégré ce code dans Tikiwiki et il est fourni en tant que *mods* (module) pour les versions 1.9.x et 1.10.x de Tikiwiki. Alors que la version 1.10.x est encore en développement, il permet au système de gestion du site d'intégrer la nouvelle interface immédiatement en téléchargeant le *mod*.

Le code de Ka-Map a été modifié pour le rendre portable et une petite librairie identifie les mapfiles compatibles Ka-Map en regardant la présence du mot clé KAMAP dans la balise METADATA de l'objet WEB. Vous n'avez qu'à spécifier le nom de la carte et les différentes échelles disponibles, par exemple :

```
KAMAP ?Pacific,1000000,500000,100000?
```

Vous pouvez également utiliser les mots-clé standard de Ka-Map tels qu'ils sont décrits dans la documentation de Ka-Map. Au final le code va identifier

⁷⁵XAJAX library site : <http://www.xajax.org/>

⁷⁶ka-Map site : <http://ka-map.maptools.org/>

le mot-clé WIKI dans LAYER METADATA et l'afficher à côté de la légende de la couche. La page wiki apparaît avec un fond semi-transparent par dessus la carte. Cette nouvelle interface a l'apparence et le ressenti de Google Map pour la navigation tout en offrant la possibilité de personnaliser les jeux de données (voir fig. 4).

L'avenir est brillant

Tikiwiki, en tant que GeoCMS, a été mis en place dans plusieurs pays⁷⁷ et est en train de devenir un succès, MapServer PHP MapScript étant assez stable avec Apache. MapServer est conforme à la norme Web Map Server (WMS) de l'Open Geospatial Consortium (OGC), ce qui signifie que beaucoup de couches peuvent être diffusées via internet à un grand panel de clients cartographiques et permettant une intégration totale.

Du côté du CMS, plusieurs développements sont envisagés. Le premier est d'inclure plus de javascript dans l'interface cartographique pour afficher les coordonnées cartographiques de la position du pointeur en temps réel, pour mesurer des distances et pour effectuer des zooms à l'aide d'un rectangle de sélection. D'une manière générale, AJAX pourrait être utilisé pour rendre l'interface encore plus ergonomique.

Quelques fonctionnalités supplémentaires seront axaminées et développées. Par exemple, la possibilité de localiser géographiquement des objets grâce à leurs attributs, par exemple : où est situé la ville appelée Suva ? Pour les couches raster et les autres, l'utilisateur pourra afficher dans une fenêtre tous les attributs d'un endroit au passage du pointeur dessus.

Plus d'objets dans Tikiwiki auront une composante géographique, comme les galeries de fichiers, les blogs, les articles, etc. Mais la fonctionnalité la plus intéressante sera de lier les trackers aux cartes. Cette fonctionnalité de tracker est en fait un système qui crée un formulaire afin d'entrer des données dans la base de données. Pour le moment, les données de tracker sont uniquement sauvegardées dans la base de données Tikiwiki mais avec l'utilisation de DSN, les trackers pourront être enregistrés dans n'importe quelle base de données, dont Postgresql. Postgresql, par l'intermédiaire de son extension PostGIS⁷⁸ peut stocker des objets géographiques et MapServer peut lire de telles bases Postgresql. Les

trackers seront un moyen d'ajouter n'importe quel type d'information géographique. Les perspectives sont donc infinies : suivi de véhicules, informations sur les pêches, système d'échantillonnage de sol, etc. L'avenir est donc brillant. Codeurs, rédacteurs et utilisateurs sont toujours les bienvenus.

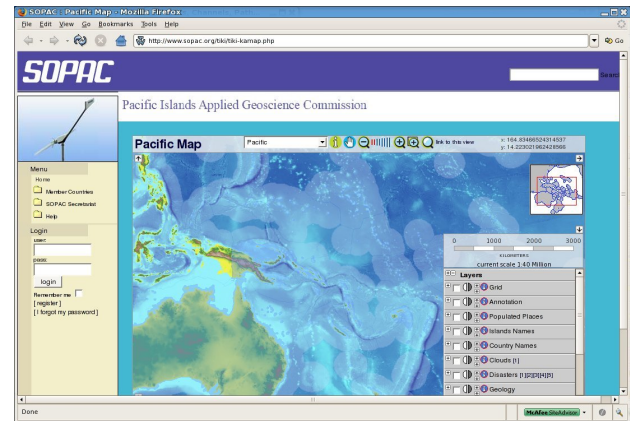


FIG. 4 – L'interface Ka-Map.



FIG. 5 – Le système de Serveur cartographique à Nauru.

Franck Martin

franck AT sopac.org

<http://www.peachymango.org/tiki-index.php?page=Franck+Martin>

Franck Martin est le principal développeur des aspects cartographiques dans Tikiwiki. Il travaille et vit à Fidji dans le Pacifique sud et a été un utilisateur précoce de GNU/Linux. Il travaille actuellement à l'installation de GeoCMS dans 14 pays des îles du Pacifique (voir fig. 5). Ce projet est financé par le Fond Européen de Développement.

⁷⁷Pays du Pacifique Sud utilisant GeoCMS : <http://www.sopac.org/maps>

⁷⁸Site de PostGIS : <http://postgis.refractory.net/>

Étude thématique

Relations spatiales dans les SIG - Une introduction

Landon Blake (alias The Sunburned Surveyor), traduit par Marie Silvestre

Introduction

Cet article fournit au lecteur une introduction aux relations spatiales dans les Systèmes d'Information Géographique (SIG). Il en présente une définition simple et explique pourquoi elles sont importantes dans les SIG. Cet article est destiné aux lecteurs ayant des connaissances basiques en SIG et n'aborde pas de concept avancé. Pour en tirer le meilleur parti, les lecteurs doivent (1) comprendre ce qu'est une "entité" dans le contexte des SIG, (2) comprendre que les entités sont habituellement représentées par des géométries de type vecteur telles que des points, des lignes et des polygones et (3) avoir des connaissances de base en géométrie et systèmes de coordonnées.

Qu'est-ce qu'une relation spatiale ?

Dans les abstractions ou les simplifications du monde réel dans les SIG, les objets sont appelés "entités". Par exemple, on pourrait représenter un réseau

routier urbain par un réseau de segments linéaires. Les entités dans un SIG ont typiquement plusieurs façons d'être représentées spatialement. On peut utiliser différentes manières de modéliser la forme et la localisation des entités dans un SIG. Les relations spatiales décrivent comment ces entités sont localisées les unes par rapport aux autres. En étudiant les relations entre ces entités, on peut en apprendre plus sur les objets réels qu'elles représentent.

Sujet de l'article

Dans cet article, nous traiterons uniquement les relations spatiales entre des entités modélisées par des "géométries vectorielles". Pour les besoins de cette discussion, nous définissons les géométries "vectorielles" comme des formes pouvant être décrites par des distances, des angles ou des coordonnées. Cet article ne traitera pas des relations spatiales dans les raster ou les grilles.

La forme et la localisation d'une entité dans un SIG sont habituellement décrites en deux dimensions mais peuvent l'être en une dimension voire en trois ou plus. Dans cet article, nous nous focaliserons sur

les relations spatiales entre des objets réels en deux dimensions uniquement. Des articles futurs pourront considérer les relations spatiales entre des entités représentées en trois ou plus de dimensions.

Définition d'une relation spatiale

La définition de *relation spatiale* que nous utilisons dans cet article est la suivante : "Une description d'une ou plusieurs manières dans laquelle la localisation ou la forme d'une entité est liée à la localisation ou la forme d'une autre entité".

Qu'est-ce qu'une géométrie vectorielle ?

La géométrie vectorielle utilise des notions mathématiques de distances, d'angles ou de valeurs de coordonnées pour décrire la "forme" d'une entité. Par exemple, vous pouvez représenter un arbre isolé par un point, une rivière ou un ruisseau par un segment linéaire ou une série de segments linéaires connectés et un groupe d'arbres ou un bosquet par un polygone. Dans le paragraphe suivant, nous considérerons la façon dont on peut décrire les relations spatiales entre deux ou plusieurs géométries vectorielles utilisées pour représenter des entités dans un SIG.

Comment peut-on identifier et décrire des relations spatiales ?

On peut décrire la relation spatiale entre deux entités représentées par des géométries vectorielles de trois manières. Vous pouvez apprendre à identifier des relations spatiales dans vos données SIG en considérant différentes entités et en considérant comment elles sont liées selon ces trois manières.

Trois manières d'identifier des relations spatiales entre des géométries vectorielles.

[1] Mesurer la géométrie de chaque entité ou forme et les comparer. [2] Considérer les mesures "entre" les localisations de deux ou plusieurs géométries d'entité. [3] Considérer comment une entité touche, recouvre, contient ou est connectée à une autre entité. (Il s'agit d'un type particulier de relation appelée topologie).

Dans cet article, nous allons nous intéresser plus précisément aux deux premières manières de décrire une relation spatiale. La topologie fera l'objet d'un prochain article.

Un exemple

La meilleure façon de comprendre les deux premières manières de décrire une relation spatiale est de prendre un exemple. Dans cet exemple, nous allons regarder quelque unes des relations spatiales entre deux entités représentées par des segments linéaires. Un segment linéaire est une portion de ligne. Dans notre exemple nous allons supposer que tous les segments linéaires sont droits et non courbes. (Si l'on s'en tient à la stricte définition géométrique, une ligne n'a pas de point terminal et s'étend à l'infini dans deux directions. C'est pourquoi nous utilisons le terme de "segment linéaire" qui a des points terminaux définis. La stricte définition géométrique d'une "ligne" ne correspond pas à ce que la plupart des gens non-initiés pensent quand ils lisent ou entendent le mot "ligne".)

La figure 001 montre deux entités. L'entité bleue sur la gauche de la figure est un ruisseau ou une rivière représentée par une série de segments linéaires connectés. L'entité sur la droite de l'image est un chemin qui court le long du ruisseau. Le chemin est représenté par un unique segment linéaire droit. J'ai numéroté chaque point ou noeud au niveau des angles et des points terminaux des deux géométries des entités avec un identifiant à trois chiffres.

Quelles sont quelques unes des relations spatiales entre les deux entités de la figure 001 qui rentrent dans les deux premières catégories listées précédemment ?

Pensez à comment peut-on comparer les mesures de chaque géométrie des entités aux mesures de la géométrie de l'autre entité. Dans cet exemple, la comparaison évidente que l'on peut faire concerne le total ou la longueur de la géométrie de chaque entité. (Notez que la "géométrie d'une entité" n'est pas la même chose qu'une "entité". Une entité représente un objet du monde réel alors que la géométrie d'une entité représente la forme d'une entité.) Notre ruisseau est plus long que l'entité adjacente du chemin dû fait de son cours sinueux. C'est une relation que l'on peut quantifier en utilisant les données de mesure des deux géométries d'entité. La longueur totale du ruisseau entre le point #101 et le point #108 est de 370 pieds. La longueur du chemin est de 345 pieds. Nous savons grâce à cette information que le

ruisseau est plus grand que le chemin de 25 pieds et que chaque pied le long du chemin correspond à une moyenne de 1,072 pieds de ruisseau.

Maintenant considérez comment peut-on décrire la relation spatiale entre les deux entités en utilisant les mesures "entre" les localisations des géométries des entités.

- On peut mesurer la distance entre chaque noeud sur le ruisseau et le chemin le long d'une direction perpendiculaire au chemin.
- On peut également comparer l'angle entre chaque segment du ruisseau et la ligne représentant le chemin.
- On peut calculer l'aire entre chaque segment de ligne du ruisseau et la ligne représentant le chemin.

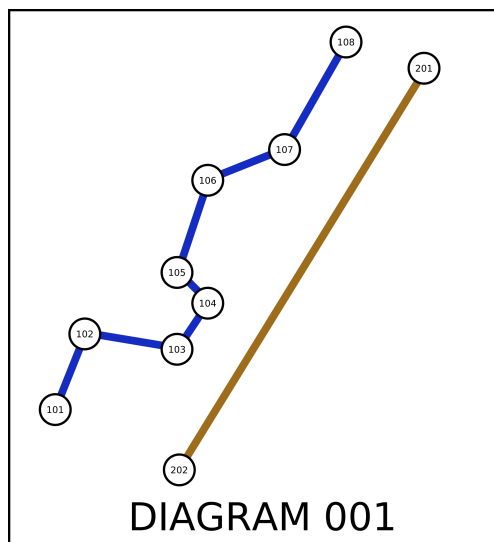


FIG. 1 – Figure 001 - Un Exemple

Pourquoi les relations spatiales sont importantes dans un SIG ?

Lorsque l'on considère les relations spatiales dans un SIG, elles nous permettent de répondre à des questions sur les entités auxquelles nous ne pourrions répondre autrement. Ainsi les relations spatiales sont une aide pour l'analyse spatiale. L'analyse spatiale nous permet de répondre à des questions utilisant des données auxquelles nous ne pouvons répondre avec des systèmes d'information traditionnels tels que les bases de données relationnelles. Le SIG permet à cette analyse spatiale de se faire parce qu'il enregistre des informations de "localisation".

Vous devez vous dire que les relations spatiales prennent une part importante dans l'analyse spatiale. La capacité d'identifier et de quantifier les relations spatiales est effectivement très importante.

Revenons à l'exemple de la figure 001 pour montrer à quel point les relations spatiales sont une part importante de l'analyse spatiale. Pensez aux types de questions auxquelles on peut répondre en utilisant les informations collectées grâce aux quelques relations spatiales qui existent entre le ruisseau et le chemin :

- Quelle portion du chemin est la plus sujette à l'érosion à cause du ruisseau ? (Cela pourrait être les portions du chemin les plus proches des segments du ruisseau et également les portions du chemin qui sont les plus perpendiculaires au ruisseau.)
- Quelle longueur quelqu'un devra-t-il parcourir le long du chemin pour parcourir la longueur du ruisseau ?
- Si nous prévoyons de réhabiliter le territoire compris entre le ruisseau et le chemin en plantant de nouveaux végétaux, combien d'ares devront être couverts ?

Conclusion

Dans cet article nous avons défini une relation spatiale comme "une description d'une ou plusieurs manières pour laquelle la localisation ou la forme d'une entité est liée à la localisation ou à la forme d'une autre entité". Nous avons discuté des trois manières d'identifier des relations spatiales entre deux entités représentées par des géométries vectorielles. On peut le faire (1) en comparant les mesures de la géométrie d'une entité ou forme aux mesures de la géométrie d'une entité ou forme d'une autre entité, (2) en considérant les mesures "entre" les localisations de la géométrie d'une entité ou forme et celle d'une autre entité et (3) en considérant comment la géométrie ou forme d'une entité touche, recouvre, contient ou est connectée à la géométrie ou forme d'une autre entité. Nous avons terminé l'article en expliquant rapidement pourquoi les relations spatiales prennent une part importante en analyse spatiale et dans les SIG.

Landon Blake

The SurveyOS Project & The JUMP Pilot Project

<http://openjump.blogspot.com/sunburned.surveyor@gmail.com>

Évaluation du Web Processing Service de l'OGC ...

... pour une utilisation dans un SIG côté client

par Christopher Michael et Daniel P. Ames, traduit par Yves Jacolin

Résumé

La spécification de Web Processing Service de l'Open Geospatial Consortium a été proposée pour répondre aux besoins de développements de modules de géotraitement par Internet, et de facilité de partage des algorithmes et des fonctionnalités de géotraitement. Cet article cherche à évaluer la proposition de WPS en respect de la faisabilité et de l'utilité potentielle, et d'identifier les zones à améliorer.

Les questions que posent la proposition du WPS sont discutés ensemble avec des solutions possibles. Plusieurs améliorations potentielles à cette proposition sont présentées et considérées, incluant un mécanisme pour guider les applications clientes dans la demande correctes des données et dans le moyen de lister les données disponibles sur un serveur.

Introduction

L'Open Geospatial Consortium (OGC, ou OpenGIS) est une organisation de normalisation par consensus d'abord concerné par la diffusion de spécifications libres (c.a.d. non propriétaire) pour unifier les logiciels d'information géographique, en apportant à la fois une variété de formats disjoints et des mécanismes de communication pour permettre une interopérabilité (Open Geospatial Consortium, 2006). Plutôt que d'éviter cette normalisation et la laisser entièrement propriétaire, plusieurs développeurs de système SIG " ont montrés une extraordinaire coopération en équipe pour soumettre des Spécifications de l'OpenGIS" (Information Today, 1997) et ont activement fait leur ces normes, certains en participant eux-même à leur développement. Comme l'OGC est composé de professionnels dans de nombreux champs d'application, plutôt qu'un simple comité dans un seul environnement corporate, les normes sont typiquement de grande qualité et sont disponibles pour n'importe quel nombre de tâche scientifiques différentes.

Le 17 novembre 2005, l'OGC a publié la quatrième révision de la proposition pour la spécification appelé Web Processing Service (WPS) (Open Geospatial Consortium, 2005). Cette spécification proposée décrit un mécanisme par lequel un géotraitement peut être réalisé sur des serveurs distant, en utilisant principalement du XML (Extensible Markup Language) pour la communication par Internet. La spécification est faite de telle sorte qu'elle doit être complètement indépendant du langage et de la plateforme. L'OGC a demandé au lecteur de commenter pendant une certaine durée, avec une date butoir au 4 février 2006. Bien que le forum pour les commentaires des lecteurs soit déjà fermé, il y en a eut, jusqu'ici, peu si toutes les études réelles sur la faisabilité et l'utilité de la proposition du point de vue du SIG côté client.

Avant le WPS, des systèmes de géotraitement basé sur le web et des approches similaires au WPS ont été développés par diverses entités. Le plus remarquable, le produit ArcInfo 8.3 (ESRI 2003) de l'Institut de Recherche de Système Environnementale (ESRI) contient une fonctionnalité appelé le Serveur de Géotraitement [NdT : Geoprocessing Server], lequel fonctionne sur des serveurs UNIX pour réaliser des géotraitement derrière des logiciels client ESRI qui envoit un travail à réaliser. Le protocole du Seveur de Géotraitement d'ESRI est propriétaire et fermé ce qui implique que seul les logiciels d'ESRI sont capable d'utiliser les possibilités de traitement distant. D'une manière intéressante, cette fonctionnalité a été retirée de la vesion suivante (ArcGIS 9.0). Une fonctionnalité similaire mais subtilement différente a été introduite dans ArcGIS Server 9.2, où un outil appelé "Model Builder" construit à partir de composants d'ESRI plus simple peut servir des clients ArcGIS Desktop et ArcExplorer (Environmental Systems Research Institute, 2006). Contrairement au WPS, le développement d'ESRI n'est pas compatible avec les produits qui ne proviennent pas d'ESRI et est un protocole de communication fermé, propriétaire l'empêchant d'être adopté largement ou étudié dans un environnement non-ESRI.

Qu'est ce qu'est le Web Processing Service (WPS) ?

Le Web Processing Service définit un mécanisme par lequel un client peut soumettre un traitement à un serveur. Le service définit une "instance serveur", ou un serveur, en tant qu'entité qui peut fournir un ou plusieurs traitements, ou des tâches de traitement individuel (par exemple, ajouter deux jeux de rasters ensemble peut être un traitement). De cette manière, n'importe quel serveur peut être capable de réaliser des traitements multiples et différents, et pas nécessairement liés.

La spécification indique que le XML (eXtensible Markup Language) doit être utilisé pour toutes les communications. Les documents XMLS sont faits d'éléments individuels, qui sont des conteneurs logiques pour des données liées. Un élément peut contenir d'autres éléments, et n'importe quel élément peut contenir des attributs qui décrivent cet élément. Un exemple simple d'un document XML pourrait être :

```
<landscape name="Smithsonian Park">
<tree type="Elm" height="8" />
</landscape>
```

Ce document XMS décrit un paysage, lequel est indiqué par l'élément nommé "landscape". Un attribut nommé "name" indique qu'il s'agit du Parc Smithsonian. Un sous-élément nommé "Tree" implique que le paysage contient ou possède cet arbre, et l'arbre a deux attributs supplémentaires qui décrivent le type de l'arbre et sa hauteur. L'élément est fermé, ou terminé, en répétant le nom de l'élément avec un slash devant.

XML a été conçue pour être "directement utilisable sur Internet", "lisible par un être humain et raisonnablement clair", "formel et concis", et "facile à écrire" (W3C, 2000). Utiliser du XML est bénéfique principalement par sa lisibilité par un être humain ce qui aide grandement dans la conception et le débogage d'application l'utilisant. Les documents XML peuvent être validés pour s'assurer qu'ils contiennent tous les éléments et attributs nécessaires. La validation est faite par un schéma XML, qui est une forme spéciale de document XML qui décrivent la structure qu'un document XML doit suivre.

Le but principal du Web Processing Service est de définir comment communiquer pour réaliser un traitement distant. À cette fin, il y a trois requêtes clés qui peuvent être faites à un serveur WPS : GetCapabilities, DescribeProcess, et Execute. La première de ces requêtes demande au serveur de lister les traitements individuels qui sont disponibles sur le serveur,

accompagné d'un court résumé et de mots-clés. La requête ne nécessite pas de paramètres. Une fois qu'un traitement a été identifié à partir de la réponse, une requête "DescribeProcess" peut être envoyée, en précisant le traitement en question comme paramètre. La réponse à cette requête inclut les mêmes informations que la réponse GetCapabilities, avec des informations plus détaillées sur les paramètres en entrée nécessaire pour le traitement et si l'entrée est simple (par exemple un nombre simple comme 23) ou complexe (par exemple un fichier de données). Les sorties complexes sont typiquement encodées en XML, par exemple en utilisant GML (Geographic Markup Language, un parent du XML) pour les données vecteurs.

Si la réponse DescribeProcess indique que c'est le traitement que l'utilisateur ou le client souhaite exécuter, la troisième requête (Execute) peut être invoquée. C'est avec cette requête que le serveur réalisera en réalité le traitement. Les paramètres obligatoires incluent le nom du traitement ainsi que toute entrée nécessaire pour ce traitement particulier. La réponse à la requête Execute est un document ExecuteResponse, un autre document XML qui indique un statut du traitement, qui retourne les entrées qui ont été utilisées, et fournit soit une sortie littérale simple ou des liens vers une sortie complexe. Le statut du traitement peut être "ProcessAccepted", informant que le traitement a été reçu et est mis dans la pile des traitements ; "ProcessStarted", informant que le traitement est en cours ; "ProcessSucceeded", signifiant que le traitement est terminé ; ou "ProcessFailed", indiquant que un problème est apparu. Si le statut est ProcessAccepted ou ProcessStarted, le statut est accompagné par un attribut qui indique où le document ExecuteResponse peut être trouvé. De cette manière le client peut vérifier le statut du traitement par une requête sur le prochain document ExecuteResponse. Dans le cas d'un retour ProcessStarted, un message du statut et un pourcentage de progression peut être fourni.

Si le statut du traitement est ProcessFailed, le document ExecuteResponse contient aussi un code d'erreur dans l'élément XML ExceptionReport, qui peut être un des 5 codes d'erreurs (MissingParameterValue, InvalidParameterValue, NoApplicableCode, ServerBusy ou FileSizeExceeded). Si le traitement réussit, le document en réponse inclura soit les sorties (dans le cas d'une simple valeur littérale) ou des liens URL vers les sorties complexes (tels qu'un fichier avec des données raster). Si une seule sortie complexe est produite, cette sortie peut être retournée directement à la place du document ExecuteRes-

ponse. Ensemble, ces trois requêtes d'opération et leurs réponses constituent la majorité de la proposition du Web Processing Service.

Pourquoi et quand un Web Processing Service doit être utilisé ?

Parce que les fonctionnalités de géotraitement proposées sont illimitées de capacités ou de nature (Open Geospatial Consortium, 2005), la proposition contient de grandes promesses dans l'utilisation d'outils de calcul sans inquiétude traditionnelle telle que la distribution de correction de bug ou la vérification de la version la plus à jour. Bien que l'étendue de ce qui peut être accompli est illimité, plusieurs opérations peuvent être complétées plus rapidement localement (c'est à dire, sur un PC de bureau d'utilisateur) qu'à distance (c'est à dire sur un serveur central), spécialement après avoir pris en compte le temps de téléchargement des données en entrée et renvoyé les données resultantes en sorties par la suite. Lors du choix entre le traitement des données en local ou distante, un certain nombre de facteur doit être considéré en plus de la taille brute des jeux de données impliquées. La complexité du calcul a une grande importance. Si le calcul prend plusieurs heures pour se terminer même pour un petit jeu de données, il peut être préférable de réaliser les calculs à distance. Si la tâche n'est pas complexe et la plus grosse partie du travail repose sur l'urgence de gros volumes de données stockées localement, il peut être plus efficace de réaliser le calcul localement. D'où, avant de choisir d'utiliser seulement le calcul de données distant, un choix intelligent doit être fait entre mettre en place un traitement local ou distant. Comme illustré dans le graphique de la Figure 1, le traitement distant a une place utile dans le traitement moderne. De plus grandes capacités de puissance de calcul dans une ferme de serveurs peut facilement réduire le coût des tâches qui demandent du temps et qui sont hautement complexe, spécialement lorsque elles sont combinées à un réseau à haut-débit tel que les fibres optiques ou l'éthernet Gigabit. Par conséquent, les données doivent être envoyées aux serveurs ayant typiquement de grande capacité de calcul, au lieu d'utiliser l'ordinateur local plus lent, lorsque le temps de calcul des données localement serait plus important que le temps combiné de transmission des données, réaliser le calcul à distance, et les récupérer.

Le calcul distant est également un choix idéal si l'algorithme est relativement nouveau, et est encore

en développement actif. Dans ce scénario, les nouvelles publications des versions ne nécessite pas des mis à jour logiciels par l'utilisateur final. Plutôt, seul le serveur nécessite une mise à jour pour refléter le nouveau code ou algorithme. Par conséquent tous les utilisateurs automatiquement gagne un accès à la version la plus récente et la plus précise du calcul simplement en utilisant le service de calcul basé sur le serveur. Ce seul point de contrôle sur le calcul crée également la possibilité de charger le serveur par son utilisation du calcul, si cela est voulu.

La vue traditionnelle des mandats de traitement distant dont la donnée est téléchargée, tel que dans un Appel de Procédure Distant [NdT : Remote Procedure Calls] (RPC) où la donnée en entrée et les paramètres sont envoyé ensemble avec un appel de fonction (Bloomer, 1992). La donnée en entrée peut être également stockée sur le serveur, nécessitant que le client définisse seulement l'entrée particulière qui est nécessaire. Cela crée l'opportunité pour un serveur de traitement de fournir également la donnée : la donnée brute ne sera pas nécessairement fournie, mais la donnée du résultat calculé peut être renvoyé sans nécessairement télécharger la donnée en entrée. Cela prend tout son sens particulièrement lors de traitement nécessitant des données en temps réel tel que les observations des stations météorologiques, les observations du trafic en temps réel, etc. Ces calculs peuvent être fourni par la même agence qui collecte la donnée, autorisant le traitement à avoir accès aux dernières données disponibles. Les motivations pour utiliser un serveur de traitement distant sont multiples, mais la décision ultime doit reposer sur l'utilisateur si le traitement distant est approprié pour la tâche ou non.

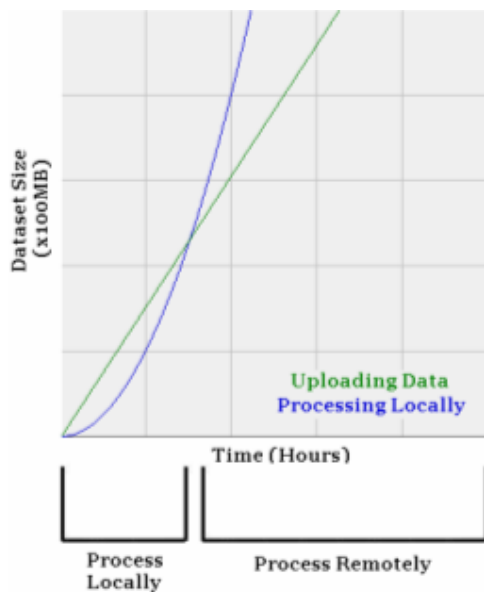


FIG. 1 – Traitement local et traitement distant ont chacun leur temps approprié.

Considérations sur le développement d'un WPS

La proposition du WPS décrit un mécanisme par lequel un ordinateur client peut soumettre un travail à réaliser sur un ordinateur distant. C'est une architecture client/serveur classique, signifiant qu'à la fois un composant client et un composant serveur sont nécessaires. Dans un but de développement et de test il est utile de compiler le composant côté client sur un système d'information géographique pré-existant pour profiter des fonctionnalités de visualisation existantes des données géospatiales à traiter et retourner, cependant, cela n'est pas toujours nécessaire et les tests initiaux peuvent être facilement réalisés en ligne de commande ou avec des outils de test ignorant l'aspect géographique. Ce composant côté client est la portion qui prend en charge la communication XML à travers Internet avec le serveur, idéalement sans que l'utilisateur n'ait besoin de voir directement ou d'écrire le WML dans le but de voir les traitements disponibles ou pour lancer leurs requêtes et récupérer les résultats. Il est préférable de compiler ce composant côté client de telle manière qu'il ne repose pas sur un logiciel particulier, propriétaire ou autre.

Le composant côté serveur lequel fournit à l'implémentation cliente (et à d'autres implémentations clientes) de véritables services doit être également indépendant (c'est à dire non relié à un algorithme

de géotraitement ou un calcul qui doit être fourni). Une approche à ce problème est de placer une couche d'interface autour de routines de géotraitement existantes ou non écrites comme des utilitaires en ligne de commande basé sous linux, des services Windows ou des applications windows. Aussi longtemps que le développement de l'application ou de l'algorithme en question peut être exécuté sans interaction avec l'utilisateur (par exemple, à travers des arguments en ligne de commande, communication TCP/IP, ou par un transfert de données OLE), cette fine couche de communications est une bonne option. Il peut être placé autour de fonctionnalité de géotraitement existant et d'outils existant (ou de nouveaux outils développés) pour leur permettre d'être servi par du WPS, en fournissant un XML qui répond aux nécessités minimales du schéma de communication WPS de l'interface. Plutôt que de nécessiter une ré-écriture complète d'un logiciel existant, celui-ci nécessitera certainement de faibles modifications pour le faire fonctionner d'une manière inattendue, idéalement via des arguments en ligne de commande et des fichiers de log générés. Les commandes de GRASS (Neteler, 2006) sont un bon exemple d'outils en ligne de commande existants lesquels peuvent être utilisés (wrapper) d'une manière telle que celle-ci.

Ce wrapper peut alors initialiser l'outil en fonction de son besoin, et contrôler un fichier de log ou renvoyer un message sur le status pour indiquer si le traitement a réussi, est encore en train de tourner, est en échec ou est dans la liste des traitements à faire. Les pourcentages du status et des messages d'erreur pertinent peuvent également être récupérés des fichiers de log, pour être retourné à l'utilisateur final à travers des communications appropriées d'une manière automatique, avec cette technique de wrapping. En plançant ce petit wrapper autour de cet outil indépendant leur permet d'être utilisés dans plusieurs endroits et pour de nombreux objectifs avec peu d'effort. Au fond ce wrapper devient le "Serveur WPS", parce qu'il prend en charge toutes les communications nécessaires par le WPS. Ce wrapper, constituant le serveur WPS, peut être développé comme une page PHP, comme une page en ASP.NET, comme une application indépendante ou en utilisant une autre technologie de serveur.

La plupart des opérations nécessaires à un serveur WPS sont simplement des opérations de métadonnées : fournir des informations sur les traitements individuels (c'est à dire, les entrées requises) et lister les traitements disponibles sur un serveur. Le serveur WPS chargera idéalement les informations sur les traitements disponibles à partir d'un fi-

chier de configuration (peut-être un par traitement) ou d'une base de données, permettant alors au codé écrit pour le serveur WPS d'être réutilisé simplement en ajoutant des traitements additionnels aux fichiers de configuration ou à la base de données. Ces fichiers de configuration ou cette base de données peut également indiquer au serveur WPS comment lancer le traitement et comment lui envoyer les fichiers en sorties.

Un aperçu du flux de communication suggérée entre une application cliente et une application serveur est montré Figure 2. Au début, l'application cliente envoie une requête au serveur, fournissant une requête GetCapabilities. Le serveur WPS renvoie alors un document XML correspondant au schéma WPS (flèche 1 dans la Fig. 2). Le client présente alors à l'utilisateur une liste des traitements disponible sur le serveur (flèche 2 dans la Fig. 2). L'utilisateur sélectionne un des ces traitements, entraînant l'envoi par le client d'une requête sur les informations additionnelles sur ce traitement (flèche 3) en envoyant une requête DescribeProcess. L'application cliente aide l'utilisateur à collecter et à entrer les paramètres en entrés nécessaire pour que le traitement soit exécuté; il initialise alors le traitement sur le serveur (flèche 3) en envoyant une requête Execute. Le serveur à son tour déclenchera la commande de lancement du traitement spécialisé (flèche 4).

Remarquez que le serveur WPS ne doit pas exécuter le traitement demandé directement, parce que cela pourrait impliquer un délai : le serveur aurait besoin de terminer le traitement avant qu'une réponse soit envoyé au client. À la place, il est préférable d'avoir le serveur WPS initialise un traitement dans différent thread, ou dans différents processus, ainsi une ExecuteResponse initial peut être immédiatement retourné vers le client. De cette manière, les opérations de géotraitement qui nécessitent beaucoup de temps permettra à l'utilisateur de travailler sur son ordinateur. Pour une meilleure propriété et sécurité, il est plus intéressant de démarrer un processus complètement nouveau, ainsi si un processus se termine anormalement il n'arrêtera pas le serveur WPS entièrement également : une commande de lancement de processus spécialisé est alors une bonne solution.

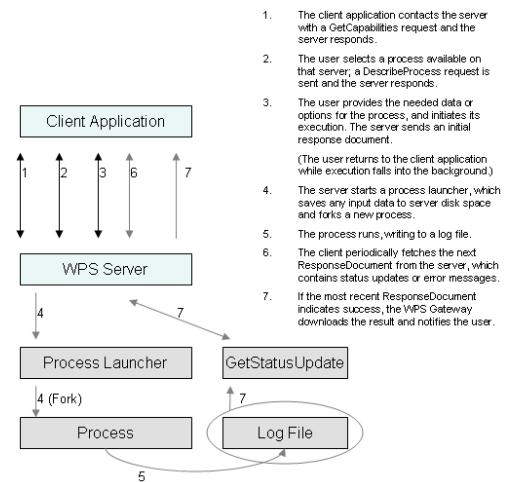


FIG. 2 – Suggested communications flow between the client and the server.

Comme expliqué à la section 2, ce document ExecuteResponse initial informe où l'application cliente peut trouver les futures mises à jours du status (c'est à dire, la prochaine ExecuteResponse) à propos de l'opération demandée, ainsi que la notification du succès, de l'échec, des travaux acceptés ou refusés, et fournit des mises à jours du status. L'application cliente doit questionner le document ExecuteResponse suivant autant de fois que l'utilisateur le désire (flèche 6 à la Fig. 2). Idéalement le serveur WPS doit être conçu de telle manière qu'un document ExecuteResponse peut être trouvé à un endroit logique, peut être en utilisant toujours la même URL pour les réponses les plus récentes. Une bonne approche pour adresser des générations de documents de réponse serait la création de page web spécialisé ou de composant serveur qui lirait simplement un fichier de log qui serait écrit par un processus exécuté pour déterminé le status en cours ou les erreurs.

Avoir la possibilité de fournir la mise à jour des status nécessite que le composant du serveur soit capable d'obtenir de tel mise à jour du processus qui est exécuté. Cela implique qu'un fichier de log est une excellent solution : le processus doit être capable d'écrire un fichier de log ou de rediriger sa sortie standard vers un fichier texte. De cette manière, le fichier peut être automatiquement lu par le serveur WPS, permettant une bonne connectivité entre les composants du système.

Une fois que le document ExecuteResponse indique que le traitement s'est terminé, le serveur doit retourner un message d'erreur applicable ou retourner l'endroit où la donnée générée peut être téléchargée ou récupérée. Le logiciel client doit alors autori-

ser l'utilisateur pour sauver ou voir la données, puis compléter le rôle du WPS.

Améliorations proposées, problèmes et solutions potentiels

Bien que la proposition du WPS soit capable d'accomplir ses buts initiaux sous sa forme courante, la proposition peut être améliorée par 6 changements clés. Ces changements incluent deux éléments additionnels dans la réponse DescribeProcess fournit par le serveur, lequel décrit l'entrée et la sortie d'un traitement données, ainsi qu'un mécanisme par lequel un client peut annuler une requête qui est en attente ou en cours. Les changements potentiels incluent également des corrections dans des comportements inconsistants, la fourniture de types d'erreur additionnels pour la prise en charge d'erreur, et la possibilité d'avoir un seul point d'entrée pour chaque traitement et peut être un seul point d'entrée pour chaque serveur.

Le premier changement suggéré est d'ajouter un élément au document XML (eXtensible Markup Language) lequel est retourné après une requête DescribeProcess. À ce jour, les entrées nécessaires sont listées par ce document, mais aucune description de la manière dont le client doit questionner l'utilisateur pour cet entrée n'est fournis. La données nécessaires peut arriver sous la forme d'une sélection d'une forme sur une carte, d'une valeur littérale tel que "23", rechercher un fichier d'un type données, ou une douzaine d'autres méthodes pour récupérer des données. Dans nos tests de développement, nous avons introduit un nouvel élément XML appelé "PromptMethod" pour résoudre ce problème. Cet élément peut contenir les valeurs "browseforvector", "browseforraster", "getboundingbox", ou "getmatchingregex". Cela impliquera que l'application cliente devra demander respectivement un fichier vecteur, un fichier raster, récupérer une boîte englobante (par exemple, en demandant à l'utilisateur de la dessiner) ou récupérer un certain nombre d'information correspondant à un motif particulier. Les trois premiers ne nécessite aucune explication; la dernière getmatchingregex, acceptera seulement une valeur entrée par l'utilisateur laquelle correspondra à l'expression régulière fournit.

Une expression régulière est une règle définie par des caractères spéciaux, tels que "[a-zA-Z][a-zA-Z]\$. Cette expression régulière cherchera au début de l'entrée (symbolisé par ^), suivit de deux caractères, de A à Z, indépendamment de la casse, suivie

par la fin de l'entrée (symbolisé par le symbol dollar). Cela fournit un filtre en entrée flexible et rapidement configurable afin de s'assurer que l'utilisateur entre bien l'entrée qui est nécessaire. Il existe certaines variations de la syntaxe dans les expressions régulières à cause de la diversité des moteurs de traitement des expressions régulières (Goyvaerts, 2006), singifiant qu'un soin particulier doit être pris afin que les expressions sont conçu de telle manière qu'elles seront interprétées correctement par tous les clients et le serveur. Un exemple de l'addition suggérée au WPS peut être vue à la Figure 3, où l'expression régulière a été définie par "`^\d{8}$`". Cette expression indique que le début de la chaîne (^) doit être suivit par 8 chiffres (d{8}) puis par la fin de la chaîne (\$). Cette expression assume que le traitement des expressions régulières de Microsoft sera utilisé.

Le second changement suggéré est une autre addition à la réponse DescribeProcess. À ce jour il n'existe pas de mécanisme par lequel un serveur peut lister les données disponibles sur le serveur pour une utilisation pour un traitement données. Un très bon exemple où cela peut être utile est dans le calcul des limites des bassins versant - définissant un réseau de flux et de bassin basé sur un jeu de données de modèle numérique de terrain (Savant et al., 2002). Les raster d'élévation peuvent typiquement être très importants, il est donc indésirable et souvent impossible de télécharger complètement le fichier en entrée sur le serveur. Les jeux de données tels que les données d'élévation ne change pas souvent, et peuvent être stockées sur le serveur pour sauvegarder du temps. Dans nos tests de développement, nous résolvons cela en introduisant un élément XML nommé "AvailableData", avec un élément enfant pour chaque données contenant un nom et une brève descriptions comme montré Figure 4. Non seulement cela augmente la vitesse du traitement en enlevant le besoin de télécharger les données en entrée, mais cela crée également un moyen pour lequel un serveur WPS peut agir comme dépôt de données ou les traiter et renvoyer les données calculées.

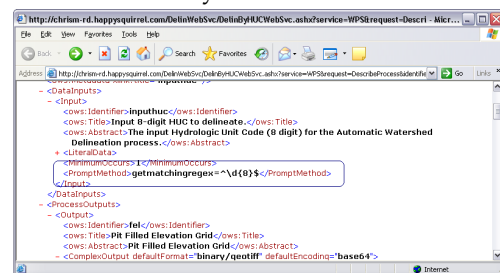


FIG. 3 – Élément PromptMethod suggéré dans la réponse DescribeProcess.

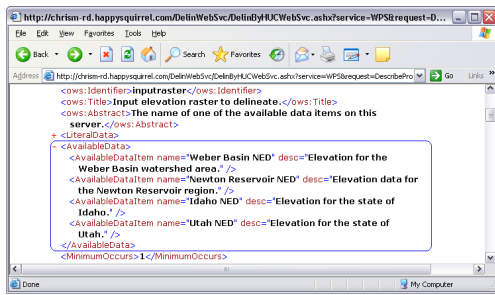


FIG. 4 – Élément AvailableData suggéré dans la réponse DescribeProcess.

Dans la proposition actuel du WPS il y a une incohérence après avoir soumis un travail au serveur en regards de ce qu'il devrait arriver par la suite. Si le traitement résultera en une seule valeur retourné et que la requête Execute a été réalisé avec le paramètre "store" à false, le WPS autorisera le traitement à retourner immédiatement la sortie, plutôt qu'un document ExecuteResponse XML. S'il y a plus d'un résultat, ou si on a demandé au traitement de stocker les résultats, alors une document ExecuteResponse est généré. Ce comportement est incohérent, puisque n'importe quel traitement pourrait retourner soit de multiples résultat en sortie ou un seul, en fonction des paramètres fournis au traitement. Au lieu de cela, il est plus simple de toujours retourner un document ExecuteResponse, en stockant n'importe quelle donnée en sortie (par exemple des fichiers vecteurs ou rasters) sur le serveur jusqu'au téléchargement par le client. De simple valeur en sortie (par exemple un nombre unique) peut être retourné directement inclus dans le document ExecuteResponse, avec des liens pointant vers les sorties complexes. Notre troisième suggestion est de toujours demander de retourner un document ExecuteResponse pour fournir une meilleur cohérence et une simplification du développement du client.

Après avoir soumis un travail au serveur WPS, il peut être utile d'annuler l'opération demandée; cette éventualité n'est pas prévue dans la proposition actuelle. Au cours de nos développements tests (et comme pour nos quatre changements proposés) nous avons introduit l'utilisation d'une "URL d'annulation de requête" dans le document ExecuteResponse, en compagnie de l'URL existante indiquant où trouver le prochain document ExecuteResponse. De cette manière un client souhaitant annuler un traitement a seulement besoin d'accéder à l'URL pour lancer l'annulation du traitement demandé, empêchant de gacher du temps de traitement du serveur pour des traitements indésirables.

Notre cinquième changement suggéré à la proposition du WPS est d'avoir un unique point d'entrée pour chaque requête possible (GetCapabilities, DescribeProcess, et Execute) pour un traitement donné. Idéalement, un unique point d'entrée (par exemple une unique page PHP) pourrait être utilisé non seulement pour chaque requête sur un traitement, mais aussi pour chaque traitement disponible sur ce serveur. Pour l'instant, chacune des requêtes qu'un traitement gère peut avoir une URL différente pour réaliser cette requête, comme illustré à la figure 5. Dans cette figure, l'URL GetCapabilities est entouré en bleu, l'URL DescribeProcess en vert et l'URL Execute en rouge. Ici chaque URL est la même, rendant la maintenance et le développement plus facile, bien qu'il soit acceptable d'avoir différent URL d'accès pour chaque opération. Des confusions ou des erreurs peuvent facilement arriver avec les différentes URL pour ces opérations. Cela peut être corrigé facilement en le rendant nécessaire pour utiliser un seul URL pour les trois requêtes qu'un traitement doit gérer.

Enfin, notre sixième suggestion est de développer un système de gestion d'erreur plus structuré. Pour l'instant, il ya très peu de types d'erreur (

Presently there are only a few exceptionMissingParameterValue, InvalidParameterValue, NoApplicableCode, ServerBusy et FileSizeExceeded), qui sont limité - particulièrement lors de la tentative de lecture automatique d'une erreur. Avec un faible nombre de types d'erreur (codes d'erreur), il n'y a pas de manière générique pour prendre en charge les erreurs de la part de l'utilisateur. Une hiérarchie plus complexe et structurée permettrait aux applications clientes de comprendre la nature de l'erreur qui est apparue, peut être en récupérant ou réessayant automatiquement si nécessaire. La possibilité de insonoriser l'utilisateur des erreurs et de récupérer élégamment est une partie importante de n'importe quelle structure de standard, et cette capacité pourrait être rendu possible avec une hiérarchie des erreurs plus détaillée.

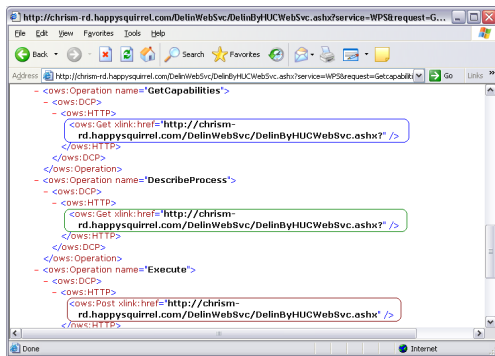


FIG. 5 – Possibilité d’avoir différente URL pour chaque requête pour le même traitement peut entraîner une confusion et une complexité à éviter.

Conclusions

Dans l’ensemble, la proposition du WPS est un moyen performant de normalisation d’un mécanisme de communications pour le géotraitement client/serveur. La proposition fonctionne comme elle est actuellement structurée et est en effet appropriée pour plusieurs tâches SIG. Le développement de composant client et les traitements WPS du serveur a montré que la proposition du WPS est solide et efficace. Néanmoins, il y a plusieurs opportunités d’amélioration à la proposition du WPS en incluant les six recommandations spécifiques fournies ici. Nous espérons que ces observations seront bénéfiques à chacun pour développer les spécifications du WPS proposé sur diverses plateformes SIG.

N’importe quelle activité de géotraitement peut être présentée au format WPS. Cela permet aux développeurs d’algorithme pour continuellement améliorer l’algorithme tout en permettant aux utilisateurs n’importe où d’utiliser le code, sans devoir mettre à jour vers la dernière version. Cela assure également que le code et les algorithmes utilisés soient de grande qualité : si un bug est détecté dans une version publiée d’un algorithme qui est distribué sur un CD, aucun moyen n’existe pour s’assurer que tous les utilisateurs mettent à jours avec succès à la version corrigée. Avec une architecture de traitement basé sur Internet, ce problème est simplifié simplement en publiant le nouvel algorithme sur le serveur de traitement. Les géotraitement consommant du temps et intensément le processeur peut être réalisé sur un serveur distant, libérant le bureau de l’utilisateur pour travailler sur d’autres tâches. Les gros jeux de données peuvent être stockés sur le serveur et traités en fonction des paramètres particuliers de l’utilisateur, en combinant les aspects de dépôt de données

et de traitement de données dans un seul système.

La proposition du WPS introduit un standard qui permettra à différents développeurs dans des endroits différents de produire des offres de géotraitement et de les leur fournir facilement à différentes plateformes clientes. Dans ces tests, il a été démontré que la proposition du WPS est pratique et utilisable dans sa définition actuelle. Les améliorations additionnelles proposées peuvent améliorer le WPS significativement, le rendant meilleur pour la tâche pour laquelle il a été créé.

Bibliographie

- [1] Bloomer, John. 1992. **Power Programming with RPC**. Cambridge, MA : O’Reilly Media.
- [2] Environmental Systems Research Institute (ESRI). 2003. **ArcGIS Desktop Products Data Sheet**. WWW document, <http://www.esrichina-bj.cn/produce/esri/arcgisdesktopsheet.pdf>
- [3] Environmental Systems Research Institute (ESRI). (2006). **ArcGIS 9.2 Webinar – ArcGIS Server : Publishing a Geoprocessing Model**. WWW document, <http://events.esri.com/info/index.cfm?fuseaction=seminarRegForm\&shownumber=9919>
- [4] **GIS Competitors Cooperate on OpenGIS Specs**. 1997. Information Today, 14(2), 15-15.
- [5] Goyvaerts, Jan. 2006. **Regular Expression Tutorial**. WWW document, <http://www.regular-expressions.info/tutorial.html>
- [6] Open Geospatial Consortium, Inc. 2006. **Vision and Mission**. WWW document, <http://www.opengeospatial.org/about/?page=vision>
- [7] Open Geospatial Consortium, Inc. 2005. **OpenGIS® Web Processing Service (WPS) Discussion Paper**. WWW document, <http://www.opengeospatial.org/>

Christopher Michael
Research Assistant
Idaho State University Geospatial Software Lab

Assistant Professor
Department of Geosciences, Idaho State University

Daniel P. Ames, PhD

Annnonce des développeurs

Mise à jour des développeurs

Pour le premier trimestre 2007

Livecd Omniverdi

par Luca Casagrande, traduit par Yves Jacolin

Qu'est ce qu'un LiveCd

Comme le définit Wikipédia, un Livecd est un "terme générique pour une distribution de système d'opération qui est exécutée au démarrage, sans installation sur un disque dur. Typiquement, il est stocké sur un média bootable tel qu'un CD-ROM (Live CD), un DVD (Live DVD), une carte flash USB (Live USB) ou d'autres." En utilisant cette technologie, les utilisateurs peuvent avoir un avant-goût d'une box GNU/Linux sans se battre avec une installation et une configuration. Une fois que le liveCd a démarré, tout est chargé en mémoire, gardant le disque dur sain. Bien sûr, une fois la machine éteinte, tout changement est perdu.

Applications

Il y a plusieurs applications à l'utilisation d'un livecd :

1. Lancer votre système d'exploitation favori partout où vous pouvez trouver un PC. Avec une clé usb pour sauver vos données, vous aurez le kit parfait de survie.
2. À des buts d'enseignement : pendant des travaux pratiques, votre audience peut avoir tout ce que vous pensez être utile pour améliorer la compréhension.
3. En utilisant une vieille machine : le livecd peut être lancé sur une vieille machine et même sans disque dur !
4. Diffuser vos données : c'est une manière très sympathique de distribuer votre travail. Vous diffusez votre travail et toutes les applications nécessaires configurées pour l'apprécier pleinement ! Par exemple, diffuser une solution complète incluant toutes les données les applications.

Limites d'un Livecd

Dans un livecd nous avons un kernel, des pilotes, WM, et toutes les applications nécessaires pour un environnement de travail. Bien sur, il est quasiment impossible d'y placer tous les pilotes qui correspondent à toutes les machines qui existent. Il peut arriver que durant un démarrage, le CD s'arrête ou que quelque chose ne tourne pas rond (par exemple, un problème de détection de LAN). Grâce au travail d'un grand nombre de personnes (et en premier lieu, Knoppix) nous avons de très bons logiciels qui détectent le matériel pendant le démarrage, mais comme tout logiciel, il peut encore y avoir des bugs. Comme vous pouvez le comprendre, l'idée d'avoir une machine universelle se rapproche du rêve et le chemin pour y parvenir est encore long.

D'un autre côté, aucune information n'est sauvegardée lorsque l'on travaille dans l'environnement du liveCd. Si cela est un avantage (c'est à dire qu'il garde votre disque dur sain), il peut être gênant d'un autre point de vue. Une solution est d'utiliser un périphérique de stockage (comme un disque dur USB, par exemple) pour stocker toute votre production. Cette approche peut aussi être utilisée pour sauver les fichiers de configuration, afin que vous puissiez les garder après un redémarrage.

LiveCd et logiciels en géomatique open source

Le LiveCd peut être un outil intéressant s'il est appliqué aux logiciels de géomatique open source.

Un exemple d'application du liveCd et de logiciels SIG FOSS est de lancer une station de travail qui permettra aux utilisateurs d'explorer les données stockées dessus ou, par exemple, dans une base de données Postgis distante. Les vieilles machines peuvent être utilisées comme un point d'accès à la base de données et peuvent être configurées avec un visualisateur qui est toujours disponible (comme les Points Internet le font avec un navigateur), permettant la sauvegarde de données sur un périphérique de stockage.

Pendant le FOSS4G 2007 à Lausanne, nous avons utilisé 2 LiveCd pendant les séminaires (ka-map et GRASS 3d). Nous avons également donné la possibilité aux participants de tester immédiatement ce qu'il avait vu dans la présentation. Une fois les travaux pratiques terminés, ils ont pu garder le liveCd et l'utiliser chez eux ou à leur travail. C'est un bon exemple du grand potentiel d'un liveCd utilisé à des fins éducatifs.

Objectifs

Notre liveCd essaiera toujours d'offrir aux utilisateurs :

1. des logiciels à jour avec chaque version principale officielle
2. des guides, documents et cours
3. une bonne compatibilité avec le matériel

Développement technique

Nous avons décidé d'utiliser Catalis : le même outil que celui qu'utilise l'équipe de création des versions Gentoo.

Ce choix se base sur deux raisons principales :

1. garder un système qui gardera une mise à jour saine et rapide
2. permettre la personnalisation de tout le contenu du CD

Desktop 2007.0

Notre première version pour 2007 contient :

- QGIS 0.8
- GRASS 6.2.1
- PostgreSQL 8.1.5
- PostGIS 1.1.4
- GDAL 1.3.2

Le système tourne sur un noyau Linux 2.6.18 et utilise GNOME comme gestionnaire de bureau.

Pendant l'écriture de cet article, nous attendons la sortie de la version 2007.1 avec Qgis 0.8.1.

Développement et tâches futurs

Notre objectif est de garder le CD à jour en créant des nouvelles versions régulièrement, ou au moins dès qu'une mise à jour majeure ou incluant des logiciels sera publiée.

En ce moment, nous travaillons sur deux tâches :

1. Un installateur pour le livecd
2. Une édition serveur avec MapServer et quelques applications de WebSIG

Installeur pour le LiveCD

L'installateur sera accessible par une icône sur le bureau. La procédure d'installation guidera l'utilisateur dans la configuration de la distribution GNU/Linux sur son ordinateur. Les installations sur disque

dur permettront de meilleures performances et offriront la possibilité de mettre à jour tous les paquets, grâce au coeur du système Gentoo Linux (<http://www.gentoo.org/>).

Édition serveur

L'édition serveur visera la création d'une machine serveur avec une gestion graphique minimale mais avec un ensemble d'applications pleinement fonctionnelles pour offrir des Web Services et des interfaces graphiques à UMN Mapserver(ka-Map, Open Layers, p.Mapper, etc.)

Plus d'informations

- Projet LiveCd d'Ominiverdi⁷⁹
- Projet Catalyst Gentoo⁸⁰

Mise à jour du status de GDAL

par Frank Warmerdam, traduit par Yves Jacolin

Publications 1.4.0 et 1.4.1

L'année 2007 a démarré avec la publication de GDAL/OGR 1.4.0. Cette version inclut un nouveau pilote raster pour le WCS (Web Coverage Server protocol), PD Set ISIS (mars). Elle inclut de nouveaux pilotes de vecteurs pour les formats Informix, KML et E00 et plusieurs améliorations de fonctionnalités des pilotes existants.

À l'heure actuelle, la version 1.4.1 devrait être disponible et elle comprend environ 40 bug corrigés. La version 1.4.1 est la première version stable de GDAL. C'est à dire que c'est la première fois que le projet GDAL a été séparé en une branche stable et se focalise sur la correction de bug dans cette branche stable, tandis que les développements continuent séparément. Cela doit permettre de sortir des versions de correction de bugs sans risquer d'introduire de nouveaux bugs dus à de nouveaux développements.

Infrastructure

Ces derniers mois ont vu des changements dans plusieurs aspects de l'infrastructure de GDAL/OGR.

⁷⁹LiveCD : <http://livecd.ominiverdi.org>

⁸⁰<http://www.gentoo.org/proj/en/releeng/catalyst/>

- Migration de CVS vers le système de contrôle de source Subversion, hébergé maintenant par l'OSGeo.
- Migration de Bugzilla au système de gestion de bug Trac, hébergé maintenant par l'OSGeo.
- Création d'un wiki pour le projet via Trac.
- Migration du site Internet et des facilités de téléchargement de mon serveur personnel vers les serveurs de l'OSGeo / Telascience.
- Utilisation d'une authentification centrale LDAP pour Subversion et Trac, partagée avec les autres projets OSGeo.

Je voudrais spécialement remercier Howard Butler qui a été l'homme de la migration sans peine de Subversion et Trac en préservant les 10 années d'historiques du vieux système.

Sponsorisation

Ces derniers mois ont également vu le lancement du programme de sponsorisation de GDAL/OGR via l'OSGeo. Nos premiers sponsors sont Analytical Graphics, Applied Coherent Technology, SRC, Safe Software, Cadcorp, Waypoint et MicroImages.

Grâce au généreux support financier de nos sponsors, le projet a pu employer Mateusz Loskot pour réaliser une maintenance générale et des tâches de gestion pour le projet GDAL/OGR de mars à août. Avec l'aide de Mateusz et d'autres membres de l'équipe de développement de GDAL/OGR, la version 1.4.1 a vu un nombre record de vieux rapports de bug nettoyés. L'embauche de Mateusz a mis le projet dans une position de force pour effectuer des tâches nécessaires sans chercher à savoir si cela correspondait à une préoccupation actuelle d'un développeur ou pas.

Un grand merci à Mateusz et nos sponsors ! Je pense que cela sera un modèle pour le travail durable nécessaire sur plusieurs projets OSGeo. Pour plus d'informations sur nos sponsors et le programme de sponsorisation de GDAL/OGR visitez : <http://www.gdal.org/credits.html>

GeoNetwork Open Source

par Jeroen Ticheler, traduit par Marie Silvestre

Introduction

Le travail s'est récemment concentré sur quelques aspects du projet :

- Développement de la version 2.1 de [GeoNetwork](#) opensource amenant à la sortie de versions bêta
- Amélioration du [site communautaire](#)
- Travail lié au processus d'incubation de l'OSGeo

Développement de la version 2.1

Les développements effectués au cours des neuf derniers mois ont permis d'ajouter beaucoup de nouvelles fonctions et d'améliorer celles existantes. L'objectif est de sortir la version finale 2.2 de l'appliquatif de catalogage vers la fin avril 2007. Voici un résumé des outils développés pour cette version :

- Support de la norme de métadonnées ISO-19115, formaté et validé par le schéma d'implémentation la norme ISO-19139. [En savoir plus](#) et se référer à la page sur la [confusion ISO19115 / 19139](#) pour plus de détails sur ces normes.
- Support de la spécification de base OGC Catalog Service for the Web (OGC-CSW 2.0.1), basé sur le profil de norme ISO pour les applications. [En savoir plus](#)
- Configuration, planification et suivi de récoltes via internet entre des noeuds GeoNetwork et des dossiers accessibles depuis internet [En savoir plus](#)
- Support de thésaurus intégré dans l'éditeur de métadonnées et dans la fonction de recherche
- Installation, mises à jour et migrations simplifiées : fonctionnalités accessibles depuis la configuration web et l'application autonome [GeoNetwork Administration Survival Tool \(GAST\)](#)
- Support multi-langues amélioré avec une fonctionnalité de traduction en ligne pour les catégories, les régions, etc.

Amélioration du [site communautaire](#)

Le site internet de la communauté a été déplacé sur un serveur telascience avec l'aide de John Graham. L'accent sera mis sur les utilisateurs de [GeoNetwork](#) opensource. Il reste encore du travail à faire pour nettoyer le contenu et enlever celui liés aux développeurs.

Le site internet basé sur Plone a été mis à jour et le contenu déplacé. Le site permet maintenant d'avoir

un contenu multi-langues, de faire des exports de pages en PDF, et d'obtenir une structure plus propre. Un Centre de Documentation et un Centre de Téléchargements forment la base du site internet.

Pour les développeurs, un site basé sur Trac <http://trac.osgeo.org/geonetwork> sera le site central, fournissant un mécanisme de suivi des bugs et un wiki. L'objectif est également de déplacer le dépôt SVN vers ce serveur pour bénéficier de l'intégration de SVN avec Trac. Le contenu du site communautaire relatif aux développeurs sera déplacé ici.

Le processus d'incubation de l'OSGeo

Le projet travaille sur le processus d'incubation de l'OSGeo. La révision du code est terminée. Le site web a été mis à jour pour intégrer le lien avec l'OSGeo. La révision des dépendances du logiciel est sur le point d'être terminée sans aucune difficulté en vue.

La direction de GeoTools

Cet article a été rédigé pendant le premier trimestre 2007 et donne un aperçu du développement de la communauté et des objectifs de recherche
par Jody Garnett, traduit par Sidonie Christophe

Communauté d'utilisateurs

GeoTools a de nouveau relancé la production mensuelle de versions majeures. La publication de ces versions régulières aide à la visibilité des progrès et à l'accessibilité à la communauté d'utilisateurs. Nous n'avons pas fait état de ceci, hormis dans la liste d'utilisateurs, car ces versions sont strictement réservées à un "accès anticipé".

- [2.4-M0](#)
- [2.4-M1](#)

Nous avons aussi mis à jour notre page Web [Module Matrix](#) (ainsi que celle de nos politiques de développement) afin de mettre en évidence les différences de gestion des modules.

Développement

En ce moment nous travaillons à la fois sur plusieurs projets payés (avec des dates limite) et sur la présentation de Martin au groupe de travail sur la GeoAPI (rencontre OGC de cet été ?).

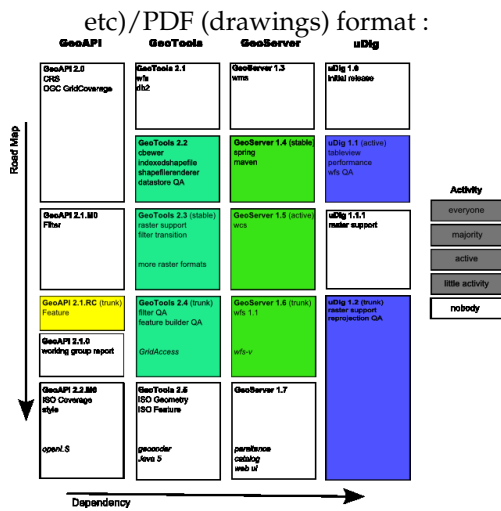


FIG. 1 – Le code couleur du diagramme ci-dessus est basé à la fois sur le projet concerné et l'activité de développement (l'intensité de la couleur indique le taux d'activité).

Axes de développement

Depuis le dernier trimestre 2006, les axes de développement des projets GeoTools et GeoServer ont été modifiés :

- **GEOT : Proposition de modification GeoTools** - Modifier l'API existante
- **GEOT : Gérer votre module** - Ajouter un nouveau module à GeoTools
- **GEOS : GSIP 6** - Suivez le tronc commun GeoTools

L'objectif des axes ci-dessus est d'obtenir une ramification uDig et GeoServer disponible, basée sur la version tronc commun GeoTools, sans laquelle nous ne sommes qu'une communauté de développement dispersée sur différentes versions (tendant certes toutes à être actives).

La modification d'un axe de développement a été planifiée :

- Quand l'accord d'un contributeur est disponible (après négociations avec l'Osgeo), nous demanderons à chaque développeur de signer quelque chose.

Objectifs de recherche

Il y a plusieurs pistes pour la recherche en cours, et quelques problèmes de conception non résolus.

etc)/PDF (drawings) format :



Etapes de recherche

Les projets GeoTools et GeoServer apportent quelques modifications de principe, en accord avec la recherche en cours.

- **GEOT : Créer votre propre module**

L'introduction d'un espace pour modules "non supportés" dans GeoTools est parvenu à rendre visible des recherches qui étaient jusque là "hors de vue, hors de pensée". Les projets uDig et GeoServer ont déjà des sections communautaires remplissant ces besoins.

Dates limite connues

Dans le diagramme ci-dessus, les dates limite connues sont soulignées.

- Voici les dates qui ont été rendues publiques :
- Article GeoAPI pour le 11 Juin (code bloqué le 11 Mai)
 - Relecture de l'article pendant les **Rencontres du 9 Juillet** du groupe de travail
 - L'article doit être soumis le 11 Juin, dernier délai (cad un mois avant).
 - La dernière fois cet article a mis un mois à être finalisé, donc le code est bloqué le 11 Mai pour les besoins de l'article.
 - Implémentation de GeoTools ISO Geometry pour mi-Mars.
 - jstWrapper ou Geometry peut être terminé suivant le soutien apporté
 - Le module ne sera peut-être pas inclus dans le téléchargement tant que Java 5 n'est pas disponible.

Le paragraphe suivant présente les objectifs de recherche uniquement (pas de version officielle requise) :

- GeoTools Feature / schéma de la communauté GeoServer pour mi-Mars

- l'implémentation sera disponible en tant que module non supporté, l'intégration se fera plus tard.

Problèmes restants

Les problèmes suivants sont listés comme des problèmes de conception auxquels il faut s'attaquer. Ces problèmes arrêtent souvent les travaux payés, mais par l'importance de la collaboration nécessaire se retrouvent sans financement spécifiques.

- **GeoAPI TypeName** - Nous sommes coincés entre la définition ISO de GenericName (entre autres le Schéma dont les noms doivent être connus avant construction) et la nécessité d'avoir un "identifiant" rapide quand on recherche le contenu d'une entité. C'est un exemple de combinaisons de choix pour la facilité d'utilisation .. difficile à résoudre. La classe Java QName est un exemple d'un bon compromis (basé sur la même spécification ISO).
 - L'interface GeoAPI TypeName a été définie en suivant le format QName, entrant en conflit avec l'interface pour GenericName
- **GeoTools ConnectionPool** - GeoTools commence à être utilisé dans des applications J2EE plus importantes, et ainsi, nous devons utiliser une recherche type JNDI pour nos DataSource (l'utilisation d'une ConnectionPool Oracle est la plus souvent demandée sur la liste utilisateur).
 - Nous avons besoin de définir une approche, de la documenter et de l'étendre aux datastores et aux plugins d'autorité des bases de données epsg.
- **GeoTools FilterCapabilities** - Nous utilisons maintenant les interfaces filtres GeoAPI. L'interface "Function" ne fournit pas le nombre de paramètres requis; l'information est récupérée comme une partie de l'information FilterCapabilities (que nous n'utilisons pas couramment).
 - Nous devons séparer la description des fonctions disponibles de la structure des données utilisée pour définir une expression.
- **GeoServer Persistence** - La capacité de persistance de l'état de GeoServer (dans un fichier XML) est difficile à maintenir et non extensible.

- Nous avons plusieurs sources d'inspiration (utilisation de XML pour les technologies bean), quelques restrictions (permettant de la configuration programmatique) et quelques tests d'acceptation (aptitude à ajouter des éléments de configuration additionnels sans changer le modèle objet).
- **GeoServer Catalog** - Un catalogue est utilisé pour gérer les connexions aux ressources, et permet des ajouts d'autres types de ressources dans le temps. L'approche existante (couper & coller) n'est pas bien adaptée à l'échelle de travail de GeoServer.
 - Nous devons ramener les interfaces GeoTools au même niveau que celles utilisées dans GeoServer et uDig. Il existe beaucoup d'exemples négatifs (sur ce que nous ne voulons pas faire) et quelques bonnes idées. GeoTools doit s'évertuer à construire un terrain minimum commun.
- **GeoServer J2EE** - GeoServer doit devenir une meilleure application J2EE.
 - Faites attention aux DataSources renvoyées via une recherche JNDI
 - Configuration du support externe et clustering
- **uDig Rendering** - Le système de rendu de uDig ayant été suffisamment utilisé en conditions réelles, nous pouvons commencer à le simplifier de façon expérimentale. Selon les limites que l'on se pose, nous sommes capables d'ajouter de l'OpenGL à notre liste de cibles (SWT et AWT sont supportés à présent).
 - Même si le travail pourrait être rétribué dans GeoTools, nous devons nous assurer qu'il y a une bonne raison (mesurée en développeurs) de rendre cet effort valable.
- **uDig Catalog** - Le système de catalogue de uDig ayant été suffisamment utilisé en conditions réelles, nous pouvons commencer à le simplifier de façon expérimentale.
 - Même si le travail pourrait être rétribué dans GeoTools, nous devons nous assurer qu'il y a une bonne raison (mesurée en développeurs) de rendre cet effort valable.

GRASS GIS

par Markus Neteler, traduit par Yves Jacolin

À propos de GRASS

GRASS (Le Système de Gestion d'Analyse de Ressources Géographique - Geographic Resources Analysis Support System) est un SIG vecteur et raster, un système de traitement d'image, un système de production graphique, et un système de modélisation spatiale. Il contient de nombreux modules pour la manipulation des données raster et vecteur le rendu des images à l'écran ou sur papier, traitement et géocodage des images multispectrales, et la gestion des attributs. Il est publié sous la licence GNU General Public. GRASS est écrit en langage C avec une interface prototype GRASS-SWIG et plusieurs applications WebSIG basées sur Python. Les fonctionnalités de gestion de données incluent aussi la modélisation de raster 3D (voxel) ainsi que l'analyse de réseau vectoriel. À travers l'interface R, des analyses de géostatistiques peuvent être réalisées. GRASS est un logiciel multi-plateforme fonctionnant sous Linux, MacOS X, les Unix compatibles Posix et MS-Windows (via Cygwin, et une majeure partie fonctionne aussi nativement).

Événements récent

- Printemps 2006 : Expertise Qualité de GRASS (QA) navigateur du code source en ligne
- 22 fév. 2006 : Publication de GRASS GIS 6.0.2 - version de correction de bug
- 12 avril 2006 : Version expérimentale native winGRASS 6 dans QGIS
- 11 Août 2006 : Version GRASS 6.1.0 publiée - c'est une version de test
- 31 Oct 2006 : Version GRASS 6.2.0 publiée - La version stable est publiée
- 12 Déc 2006 : Version GRASS 6.2.1 publiée - Cette version corrige plusieurs bugs découverts dans le code source de la 6.2.0
- 20 Déc 2006 : Newsletter GRASS GIS / OS-Geo publiée - Le premier volume combiné de GRASS-News et OSGeo-News est disponible
- 10 Feb 2007 : les binaires de GRASS GIS 6.2.1 winGRASS/Cygwin sont disponibles
- 12 Feb 2007 : Nouveau tracker de bug et de fonctionnalité pour GRASS - basé sur Gforge
- Rencontre des Utilisateurs italiens de GRASS et GFOSS - Le salon des utilisateurs de GRASS et GFOSS, Palerme (Italie), 14-16 Fév. 2007

En développement

En plus de nombreuses corrections de bug et de tests poussés, plusieurs projets majeurs sont en cours :

- finalisation du port natif de winGRASS
- Publication de GRASS GIS 6.3.0 ([Brouillon de l'annonce](#))
- nouvelle interface graphique utilisateur basée sur python en progression
- travail lié au processus d'incubation de l'OS-Geo (vérification du code et ajout des énoncés de copyright manquant)

Statistiques

- Durant les trois premiers mois de 2007, les [statistiques du serveur principal](#) indiquent (il y a plus que 20 sites miroirs, non inclus ici) :
 - plus de 5000 téléchargements du code source de GRASS 6.2.1
 - plus de 2000 téléchargements de MS-Windows/Cygwin
 - plus de 1000 téléchargements pour Linux
 - 11100 téléchargements du jeu de données échantillon Spearfish
- La liste de diffusion [grassuser](#)
 - plus de 800 membres
 - moyenne de 15 messages par jour
- La liste de diffusion [grass-dev](#)
 - plus de 450 membres
 - moyenne de 20 messages par jour
- Il existe 12 listes additionnelles et plusieurs listes d'utilisateurs nationales sur GRASS
- Durant les 8 derniers mois, toutes les 1.16 heures, une contribution est apportée au dépôt du code source de GRASS ([source](#))
- Plus de statistiques utiles peuvent être trouvées sur le site Ohloh : <http://www.ohloh.net/projects/3666>

Mise à jour du projet Mapbender

par Christoph Baudson, traduit par Cyril de Runz

À propos de MapBender

[MapBender](#) est le logiciel et portail web dédié au traitement des données géographiques ayant une architecture OGC Web Service (OWS). Le logiciel fournit des technologies internet pour le traitement des

services de données spatiales programmés en PHP, Javascript et XML. Il procure, de même, un modèle de données et des interfaces pour la visualisation, la navigation et l'interrogation de services de cartographie conforme aux standards OGC. La structure de Mapbender offre, de plus, des services d'authentification et d'autorisation, des fonctionnalités de proxy OWS, des interfaces de gestion pour les utilisateurs, les groupes ainsi que pour l'administration des services dans des projets WebSIG.

Événements récents

Janvier 2007

- sortie de la version candidate Mapbender 2.4.1
- adoption des conventions de codage
- création des modules maintainers, personnes chargées du suivi des bugs, de la documentation, etc. sur un module

Février 2007

- nouveau gestionnaire de suivi des problèmes
- établissement d'une version propriétaire

Mars 2007

- conférence hebdomadaire des développeurs sur IRC
- Mapbender 2.4.1
- rencontre des développeurs en face-à-face à Berlin

Nouvelles caractéristiques de Mapbender

- **Nouveaux outils d'administration** : manipulation des métadonnées, nouvelles interfaces d'administration.
- **Nouveau module d'arborescence** : amélioration de l'utilisabilité par la centralisation des contrôles au sein d'un unique module. Les contrôles dans des fenêtres pop-up sont désormais obsolètes. Possède aussi un indicateur de statuts des WMS.
- **Recherche dans les métadonnées** : une recherche textuelle dans les métadonnées et les mots clés de la couche WMS. Récupérer une couche simple ou un WMS grâce à une requête AJAX.
- **Amélioration des possibilités de configuration de WFS-T** : une meilleure interface utilisateur pour faciliter et enrichir les manipulations de données WFS-T.
- **Amélioration du module de numérisation** : un design complètement nouveau suivant une ap-

proche AJAX. Amélioration de l'utilisation et des performances.

Développements futurs

- **Intégration d'OpenLayers** : une intégration d'OpenLayers comme module optionnel de Mapbender est prévue pour 2007.
- **Amélioration de la numérisation et des WFS-T** : manipulation des géométries complexes.
- **Utilisation large des technologies AJAX, JSON et OO** : amélioration de modules d'interfaces pour faire justice au large évolution du public.
- **Evaluation des garnitures des algorithmes** : A cause de la minimisation du trafic, il existe peu commentaires dans le code source. Dans le futurs, JSDoc et PHPDoc seront utilisés pour commenter. Le problème du trafic sera résolu par l'extraction des commentaires antérieurs au déploiement et en complétant le JS-code.

Statistiques

- **Croissance de la participation à la liste de diffusion des développeurs** : 2006 S3 : 40 posts, 2006 S4 : 121 posts, 2007 S1 : 209 posts (jusqu'au 19 mars)
Jetez un oeil aux statistiques de Mapbender sur Ohloh. En voici quelques extraits :
- **Valeur du projet** : \$2,436,378. "Ce calculateur estime le coût salarial d'une équipe pour écrire ce projet en partant à zéro."
- **17 développeurs** : "Durant les douze derniers mois, 10 développeurs ont contribué à Mapbender en produisant du nouveau code. C'est une équipe relativement importante, qui propulse le projet dans le top 5% de tous les projets pour Ohloh. Pour cette statistique, Ohloh ne considère que les changements récents du code. Depuis le début du projet, 17 développeurs y ont participé."
- **Croissance de l'activité** : "Durant les douze derniers mois, Mapbender a connu une substantielle augmentation de son activité. Ceci est probablement significatif de la croissance de l'intérêt général pour ce projet et que la communauté de l'OpenSource a adopté ce projet."

Mise à jour du projet MapBuilder

Direction stratégique de mars 2007 par Cameron Shorter, traduit par Yves Jacolin

Status

MapBuilder est un client de cartographie puissant, conforme aux standards et libre qui fonctionne dans un navigateur Internet. Il affiche des cartes raster à partir de serveur WMS, Google maps et autres, et des couches vectorielles à partir de serveur WFS, GeoRSS et GML. Il offre même la fonctionnalité d'édition par WFS-T. MapBuilder est souvent décrit comme une boîte à outils basée sur Internet, ou un framework, qui permet aux développeurs d'insérer une sélection de widgets dans une page web. Par exemple, le panneau de cartes (MapPanels), la liste des fonctions (FeatureLists), des outils de navigation, des éditeurs de style et d'autres encore.

L'année dernière, les trois clients de cartographie leader, Mapbuilder, OpenLayers et MapBender ont travaillé activement ensemble partageant des idées et du code. En particulier, OpenLayers est en train d'être inclus dans Mapbuilder comme un moteur d'affichage. Notre dernière version, mapbuilder-1.5alpha1 inclut un affichage par OpenLayers et notre prochaine version complètera l'intégration en liant les outils Mapbuilder et OpenLayers.

OpenLayers se focalise sur l'affichage de la carte. Mapbuilder étend cela pour offrir des widgets supplémentaires comme l'éditeur de la Description du style des couches (SLD), Les services de cartographie sur Internet des séries temporelles, le traitement des documents de contexte des cartes sur Internet (WMC) et plus.

Les développeurs de Mapbuilder contribuent maintenant au code d'OpenLayers lors de l'ajout des fonctionnalités de base de cartographie.

Fusionner les codes entre les projets est délicat autant émotionnellement que techniquement. Le catalyseur pour la fusion entre Mapbuilder/OpenLayers était qu'il y avait quatre projets différents pour développer l'affichage de vecteur en utilisant SVG/VML fin 2006. Après une discussion nous sommes tombés d'accord sur une même base de code. Cela signifiait que chacun de nous devez jeter environ les 3/4 de notre code. À court terme, cela signifiait des efforts supplémentaires de nous tous, mais à long terme, nous bénéficieront tous de cette fusion. Nous aurons plus de développeurs pour maintenir le même code de base et des utilisateurs moins perdu lorsqu'ils choisiront un client.

Les développeurs qui jettent du code ressentent une grosse perte de status et de crédibilité. Le mérite va à ceux qui ont été assez flexibles pour jeter leur code ainsi qu'aux développeurs qui ont généreusement remercié les travaux passés.

En interne, Mapbuilder stocke ses données spatiales dans un document Context. Initialement, nous utilisons un document Web Map Context (WMC) qui décrit une liste de couches WMS. Cependant, le WMC ne vous autorise pas à insérer d'autres couches, comme WFS, GML, GeoRSS, Google Maps, etc.

Dans mapbuilder-1.5alpha1 nous gérons les Context OWS (pour l'instant un document en état 'Brouillon' de l'OGC) qui étend le WMC pour inclure de multiples types de couches. Cela améliore la structure de nos fichiers de codes et de configuration, tout en continuant avec la gestion des standards de l'OGC. La branche 1.5 sera stabilisée dans les tous prochains mois avec des versions candidates et la version finale.

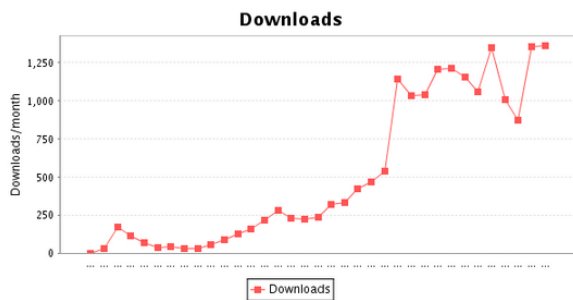
Mapbuilder a réussi la procédure d'incubation de la Fondation Open Source Géospatiale (OSGeo) en octobre 2006 devenant le second et le plus récent projet à y parvenir. La procédure a été un processus très long impliquant un audit du code et une remise en ordre des procédures de développement. Cependant, comparé aux autres projets en court d'incubation, Mapbuilder est jeune et a beaucoup moins d'histoire à fouiller, ce qui explique pourquoi il est l'un des premiers projets à être intégré (après Mapbender).

L'association avec OSGeo a été un mouvement positif pour la popularité de Mapbuilder. Les téléchargements de Mapbuilder ont doublé dans le mois qui ont suivi la création de l'OSGeo avec Mapbuilder comme un des projets de la fondation.

En résumé, Mapbuilder continue à avoir une large base d'utilisateurs et de développeurs, il a une base de code stable, de bonnes procédures de développement, un futur sain est devant lui.

Mesure des téléchargements

Ce graphique montre un accroissement uniforme de l'intérêt depuis le démarrage du projet, avec un doublement du nombre de téléchargements lors de la création de l'OSGeo (avec Mapbuilder comme projet fondateur) en mars 2006. Il y a eut un creux à Noël 2006, puis un retour en début 2007 (le visualisateur gml et mapbuilder-1.5alpha1 ont été sortis à la fin du mois de décembre 2006).



Mesure des commit dans Subversion

Ce graphique récupère le nombre de ligne de code dans le dépôt de Mapbuilder. Le graphique montre que Mapbuilder a beaucoup grossi depuis décembre 2003. Nous avons un support conséquent de la part de la communauté des développeurs qui inclut un cœur constant, ainsi qu'un nombre de nouveaux développeurs qui rejoignent la communauté, développent un temps et s'en vont.

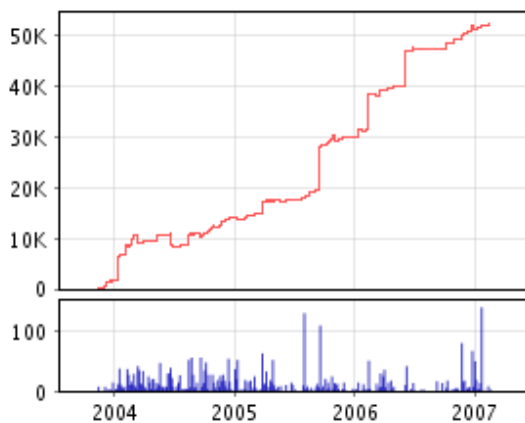
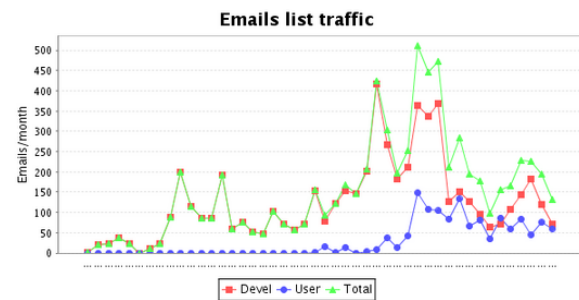


FIG. 2 – Source data <http://fisheye.codehaus.org/browse/mapbuilder/trunk/mapbuilder/mapbuilder/lib>

Mesure du trafic des emails

Ce graphique montre le nombre mensuel d'email sur les listes des développeurs et des utilisateurs. Les emails des développeurs explosent fin 2005, début 2006 pendant le développement de la version 1.0 avant de revenir au niveau précédant. La stabilisation du trafic email peut être mise sur le compte de plusieurs éléments :



1. Un effort important a été réalisé sur la documentation. Souvent les réponses des questions de débutants renvoient vers la documentation existante. Cela ne signifie pas que notre documentation est merveilleuse, elle s'est juste améliorée.
2. Le travail sur le rendu vecteur réalisé depuis mi-2006 a été réalisé en conjonction avec des développeurs externes. Une grande partie des échanges s'est déplacée vers d'autres listes ou les canaux IRC (En particulier `irc://freenode.net#openlayers`)
3. Avril - Octobre 2006, une part importante de l'effort des développeurs s'est focalisée sur l'obtention de la graduation de l'OSGeo plutôt que dans le développement. Une chute dans l'activité des mails sur ces derniers mois est probablement dû à cela.

Mesure du téléchargement du visualiseur GML

En décembre 2006, nous avons sorti la version du visualiseur GML de Mapbuilder qui peut être distribué avec un jeu de données GML. Il a été distribué sur un CD en tant que "artefacts" de l'Open Geospatial Consortium Testbed 4. Il a été également rendu disponible au téléchargement et les statistiques sont disponibles ci-dessous. Le nombre de téléchargements est le 1/10e des téléchargements de mapbuilder-libs ce qui suggère que le marché cible a obtenu leur visualisateur GML du CD ou que les personnes sont plus intéressées par d'autres fonctionnalités de Mapbuilder.

Date	Téléchargements
Dec-06	75
Jan-07	104
Fev-07	115

Source des données : ⁸¹

⁸¹Téléchargement du visualiseur GML : http://sourceforge.net/project/stats/detail.php?group_id=35246&ugn=mapbuilder&type=prdownload&mode=alltime&package_id=116388

À propos de l'auteur

Cameron Shorter fait parti du Comité de Gestion du Projet Mapbuilder.

MapGuide Project Update

par le comité de direction du projet MapGuide, traduit par Gwenael Bachelot

À propos de MapGuide

MapGuide Open Source est une plate-forme évoluée pour le développement d'applications web géospatiales, permettant la publication et la mise à jour de données. MapGuide est très flexible : multi plate-forme (Windows et Linux) et multi langages (PHP, Java, et ASP.Net). MapGuide intègre une puissante API géospatiale de traitement, et un grand nombre de formats de sortie, comme des formats images (PNG, JPG, etc.), WMS, WFS, et KML. De plus, en exploitant la bibliothèque Open Source d'abstraction d'accès aux données FDO, MapGuide peut accéder à plusieurs sources de données comme MySQL, PostGIS, Oracle, ArcSDE, fichiers Shape, SDE, WMS, WFS, et de nombreux formats d'images.

Événements récents

Depuis les débuts de ce projet, on recense plusieurs événements significatifs :

- Novembre 2005
 - Autodesk annonce la mise à disposition en open source du code source du nouveau MapGuide.
 - Sortie de la pré-version de MapGuide Open Source (sous le nom MapServer Enterprise, qui sera ensuite modifié).
- Février 2006
 - Le code de MapGuide est mis à disposition à travers l'OSGeo (Open Source Geospatial Foundation).
 - Sortie de MapGuide Open Source 1.0.0
 - L'équipe de développement de MapGuide commence à utiliser des mécanismes publics de suivi des problèmes et de remontée d'informations par les utilisateurs.
- Juillet 2006
 - Sortie de MapGuide Open Source 1.0.1

- Septembre 2006
 - Le passage en revue de la provenance du code source est terminée.
- Octobre 2006
 - Sortie de MapGuide Open Source 1.0.2
 - Le comité de direction du projet MapGuide Open Source est créé. MapGuide Open Source est maintenant contrôlé par un groupe de développeurs et d'utilisateurs, dont la majorité d'entre eux ne sont pas des employés Autodesk.
 - La première RFC (Request For Change - demande d'évolution) est publiée pour évaluation.
- Janvier 2007
 - Sortie de MapGuide Open Source 1.1.0
 - Le fournisseur open source Raster GDAL remplace le fournisseur Raster Autodesk propriétaire.
 - Un outil de suivi (Trac) est mis en place. Cet outil permet aux utilisateurs de soumettre des bugs et des demandes d'améliorations.
- Mars 2007
 - MapGuide a passé le processus d'incubation de l'OSGeo, en montrant le niveau requis de participation, de liberté sur la propriété du code source et d'usage des méthodologies standard de développement open source.

En développement

Ces derniers mois, plusieurs fonctionnalités nouvelles ont été ajoutées à MapGuide Open Source.

- Publication native au format KML, avec le support des polygones (régions), de l'affichage en fonction du niveau de zoom, et de l'extrusion.
- Introduction d'un moteur de symbologie très paramétrable, autorisant des styles complexes pour les points et les lignes.

Statistiques

- Durant les trois premiers mois de 2007, les statistiques de téléchargement montraient approximativement :
 - 2 300 téléchargements de la version Windows
 - 900 téléchargements de la version Linux

- 1 200 téléchargements des exemples pour les développeurs PHP
- 1 200 téléchargements des exemples pour les développeurs .Net
- 750 téléchargements des exemples pour les développeurs Java
- Liste de diffusion mapguide-users
 - 470 membres
 - Une moyenne de 20 messages/jour
- Liste de diffusion mapguide-internals (développement du moteur)
 - 130 membres
 - Une moyenne de 8 messages/jour
- 17 RFCs (Requests for Change - demande d'évolution) soumises et adoptées
- Des statistiques plus utiles sont disponibles sur le site sur le site Ohloh : <http://www.ohloh.net/projects/4656>

Projet MapServer

par Daniel Morissette, traduit par Yves Jacolin

À propos de MapServer

MapServer est un environnement de développement pour construire des applications et des services de cartographie sur Internet. Il est rapide, flexible et sûr, il peut être intégré à n'importe quel environnement SIG. Il gère beaucoup de formats de données spatiales courants incluant les services web de l'OGC.

MapServer propose MapScript, un puissant environnement de script, qui gère PHP, Python, Perl, C#, Java, etc. L'utilisation de MapScript rend la création d'applications web géospatiales complexes facile et rapide.

Événements récents

Octobre 2006

- Sortie de la version 4.10.0 de MapServer

Février 2007

- Sortie de la version 4.10.1 de MapServer

Développements futurs - MapServer 5.0

L'équipe de développement de MapServer prévoit de sortir de la version 5.0 pour la fin de l'été 2007. La [plannification de la version 5.0](#) décrit un certain nombre de mises à jour prévues, incluant :

- Gestion de la bibliothèque AGG, un moteur de rendu nouveau et performant comme alternative à GD
- Mise à jour de la gestion de WMS à la version 1.3.0
- Amélioration de la gestion du Serveur SOS de l'OGC
- Nouvelle gestion de l'OWS commun de l'OGC en version 1.0.0
- Possibilité de créer des graphiques dynamiquement
- Prioritisation des étiquettes
- Corrections de la gestion de la mémoire de MapScript
- Restructuration du mécanisme de rendu des LOG/DEBUG
- ... et beaucoup d'autres choses ...

Statistiques

Statistiques de la liste Mapserver-Users list (pour le premier trimestre 2007) :

- Nombre d'inscrit : 2 344
- Nombre de pays représenté : 67 (Sur tous les continents sauf l'Antartique)
- Nombre de message par mois : 415 (moyenne pour Janvier-Mars 2007)

Jetez aussi un oeil sur les statistiques de développement de MapServer sur [Ohloh](#). En voici quelques aspects :

- **27 développeurs** : Sur les 12 derniers mois, 12 développeurs ont contribué à l'ajout de code nouveau à MapServer. C'est une équipe relativement grande, mettant ce projet dans le top 5% de toutes les équipes de projet sur Ohloh. Pour cette mesure, Ohloh a uniquement considéré les changements récents du code. Sur l'historique complet du projet, 27 développeurs ont contribué.
- **Mature, base du code bien établie** : Les premières lignes du code source ont été ajoutées à MapServer en 2000. C'est une période d'activité relativement longue pour un projet open source et cela peut être un très bon signe. Un long historique du contrôle des sources tel que celui-ci indique que le projet a assez de mérite

pour pouvoir capter l'intérêt des contributeurs sur une longue période. Cela indiquerait une base du code relativement mature et dépourvue de bug et serait le signe d'une équipe de développement organisée et dédiée.

Remarque : Le code source de MapServer est en réalité plus vieux que l'historique du contrôle des sources. le projet a démarré autour de 1996 mais l'utilisation de CVS a débutée en 2000.

Mise à jour du projet Quantum GIS

Gary Sherman, traduit par Cyril de Runz

Version actuelle et développements futurs

L'équipe de développement de QGIS a sorti la version 0.8.0 le 29 décembre 2006 à l'issue d'une longue période de développement. La majorité du temps a été dépensée au portage de la base du code de Qt 3.3 à Qt 4.2.x. Cet effort accompli, le projet va produire des versions à une fréquence plus actuelle. La version 0.8.1 est prévue pour la fin du premier trimestre 2007. Pendant la phase préliminaire de résolution des bugs, il y aura quelques améliorations mineures, incluant un ensemble élargi d'outils dans la boîte à outils "GRASS toolbox".

Modifications au niveau du système de construction

Le projet QGIS a migré vers l'utilisation de CMake plutôt que celle des GNU autotools. Cela deviendra officiel avec la version 0.9 mais les changements ont d'ors et déjà été reportés sur la branche 0.8.1. CMake a prouvé sa facilité d'utilisation, sa souplesse et les résultats donnent des temps de compi-

lations du code de base de QGIS vraiment plus rapides.

Qu'est ce qui est en cours dans la version 0.9

Dans la version 0.9, les liaisons Python sont incluses dans la plupart des classes de l'API de QGIS. Cela signifie que vous pouvez écrire à la fois des plug-ins et des programmes pour station autonome en utilisant Python et PyQt. De nombreuses personnes développent d'ors et déjà des plug-ins. D'ailleurs, les fameux "Tim Sutton C++ tutorials" ont aussi été adaptés pour Python par Martin Dobias. Regardez sur le blog dédié à QGIS ⁸² pour rechercher les tutoriels et plus d'informations.

Il y a un nombre d'évolutions substantielles dans la version 0.9, notamment un effort appuyé sur le support d'utilisation des bibliothèques QGIS pour vos projets de programmation. L'équipe de développement travaille toujours sur les derniers détails de la version 0.9. Veuillez regarder le site web pour plus d'informations sur l'état des travaux.

Si vous êtes aventuriers, vous pouvez essayer les nouvelles fonctionnalités ainsi que l'outil Python. Pour cela, vous devez tester les derniers développements du code (qui sont donc instables) en utilisant le svn ⁸³.

QGIS entre dans l'OSGeo

QGIS a postulé et a été accepté dans le processus d'incubation de l'OSGeo en février dernier. Une élection au sein de la communauté a indiqué que la majorité des personnes était en faveur de cette adhésion. Actuellement, la phase de travail avec un mentor OS-Geo afin de devenir un projet OSGeo à part entière a commencée.

⁸²qgis blog : <http://blog.qgis.org>

⁸³qgis svn : <https://svn.qgis.org/repos/qgis/trunk/qgis>



**2007 FREE AND OPEN SOURCE SOFTWARE
FOR GEOSPATIAL (FOSS4G) CONFERENCE**
VICTORIA CANADA  SEPTEMBER 24 TO 27, 2007

FOSS4G - Ouverture des Inscriptions à la Conférence

Nous sommes heureux de vous annoncer l'ouverture des inscriptions en ligne à la Conférence Free and Open Source Software for Geospatial 2007 (FOSS4G 2007). FOSS4G est l'évènement annuel qui réunit les personnes et les sociétés qui créent, utilisent, et gèrent des logiciels géospatiaux open source. Inscrivez-vous dès maintenant en ligne.⁸⁴

Inscrivez-vous avant la date limite du 27 Juillet, pour économiser sur les frais d'inscription! Tirez profit de l'opportunité que FOSS4G 2007 vous offre, de construire un réseau avec les autres professionnels des données géospatiales, de renouveler d'anciennes relations, et d'en créer de nouvelles.

Pour les dernières mises à jour, l'inscription et/ou la soumission d'une présentation, visitez le site web de la conférence.⁸⁵

OPPORTUNITES D'EXPOSITION & DE SPONSORING

Concernant les opportunités d'exposition et de sponsoring, lisez la page des partenaires ⁸⁶ ou

contactez Paul Ramsey, Président de la Conférence par email.⁸⁷

SOUMETTRE UNE PRESENTATION

Vous pouvez soumettre une présentation en ligne.⁸⁸ La date limite pour les soumissions est le 29 Juin 2007.

Les présentations FOSS4G durent 25 minutes, avec 5 minutes de questions/réponses à la fin. Les présentations concernent l'utilisation ou le développement de logiciels géospatiaux opensource. Tout le monde peut soumettre une proposition de présentation et participer à la conférence comme présentateur. Plus d'informations sont disponibles sur la page des présentations sur le site web.

Nous espérons vous voir à Victoria, au Canada en Septembre!

⁸⁴Inscription en ligne : <http://www.foss4g2007.org/register/>

⁸⁵Site web de la conférence : <http://www.foss4g2007.org/>

⁸⁶Page des partenaires : <http://foss4g2007.org/sponsors>

⁸⁷Email Paul Ramsey : pramsey@foss4g2007.org

⁸⁸Soumettez une présentation sur <http://www.foss4g2007.org/presentations/>

Rédacteur en chef :

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

Rédacteur, Actualité :

Jason Fournier

Rédactrice, Étude de cas :

Micha Silver

Rédacteur, Zoom sur un projet :

Martin Wegmann

Rédacteur, Étude d'intégration :

Martin Wegmann

Rédacteur, Documents de programmation :

Landon Blake

Remerciements

Tous les relecteurs & le project Actualités de GRASS

Le *journal de l'OSGeo* est une publication de la *Fondation OSGeo*. La base de ce journal, les sources du style $\LaTeX 2_{\epsilon}$ ont été généreusement fournies par l'équipe éditoriale de l'actualité de GRASS et R.



This work is licensed under the Creative Commons Attribution-No Derivative Works 3.0 License. To view a copy of this licence, visit :

<http://creativecommons.org/licenses/by-nd/3.0/> or send a letter to Creative Commons, 171 Second Street, Suite 300, San Francisco, California 94105, USA.



All articles are copyrighted by the respective authors. Please use the OSGeo Journal url for submitting articles, more details concerning submission instructions can be found on the OSGeo homepage.

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

Site de l'OSGeo : <http://www.osgeo.org>

Contact postal pour l'OSGeo, PO Box 4844, Williams Lake, British Columbia, Canada, V2G 2V8



ISSN 1994-1897