

## RAPPORT DE STAGE



***Le développement de services web géographiques dédiés à la consultation et au traitement des données géospatiales et agrométéorologiques sur Internet.***



***Sous la direction du Docteur Alain VIAU du 2 mars au 23 Août 2005, Québec***

*Un grand merci au Docteur Alain Viau, à Mr Richard Picard et à toute l'équipe du GAAP, ainsi qu'à la direction du Master SIG et à ma famille qui ont rendu cette expérience possible.*

## RESUME

---

Ce document présente le stage professionnel réalisé par Nicolas Bozon au Laboratoire de Géomatique Agricole et Appliquée à l'Université Laval (Québec), du 2 mars au 23 août 2005. Il expose de manière synthétique les concepts abordés et les compétences acquises durant la période de travail, en proposant une description globale de la démarche de stage, laquelle s'est déroulée en rapport au projet Mésonet-Montréal.

La spécialisation en SIG Web résultant du stage est particulièrement mise de l'avant, à travers notamment la présentation des travaux de développement Internet, mais aussi par l'explication générale du lien entre le SIG du laboratoire et le réseau Internet.

La démonstration des apports du stage aussi bien théoriques que pratiques est l'objectif premier de ce rapport. Il met en valeur les compétences géomatiques et informatiques acquises tout en conservant une vision pré-professionnelle et un certain recul vis-à-vis de la mission de stage.

## CONSEILS DE LECTURE

---

Le produit final du stage est un SIG en ligne, sous la forme d'un site Internet disponible à l'url suivante : <http://www.mesonet-montreal.ca/cartonico> .

La compréhension du système mis en place, et notamment lors de la lecture de la quatrième partie doit se faire parallèlement à la visite du site Internet. Cela permet d'illustrer les propos tenus de manière concrète.

Pour des raisons de lisibilité, le code inclut dans le texte apparaît en italique, et les sigles, les noms de langage Internet (sauf javascript) et les noms de formats informatique apparaissent en majuscules.

Le texte fait appel à 11 annexes présentées à la fin de ce rapport. Elles comportent notamment une partie du code source utilisé pour mettre en place le SIG en ligne.

Dans les annexes, le code apparaît en caractère normal (et non en italique), et les commentaires en caractères gras.

# SOMMAIRE

---

<b>Introduction</b> .....	7
<b>1) Présentation du stage</b> .....	8
1.1) La structure d'accueil.....	8
1.1.1) Raison sociale et activités du laboratoire.....	8
1.1.2) Hiérarchie et fonctionnement.....	8
1.1.3) Collaborations et financements.....	9
1.1.4) Ressources et équipements.....	10
1.2) La Présentation du Projet Mésonet Montréal.....	11
1.2.1) Nature et Objectif du Mésonet.....	11
1.2.2) Aspects institutionnels et collaboratifs.....	12
1.3) Les caractéristiques de la mission.....	13
1.3.1) Objectifs du sujet de stage.....	13
1.3.2) Méthodes de travail .....	14
1.3.3) Outils privilégiés.....	15
<b>2) Le Moteur SIG agrométéorologique</b> .....	16
2.1) Le réseau de stations.....	16
2.1.1) Localisation et caractéristiques.....	16
2.1.2) Observations et automatisation.....	17
2.2) Les bases de données.....	18
2.2.1) La base de données météorologiques.....	18
2.2.2) La base de données spatiales.....	19
2.3) Structuration du SIG.....	20
2.3.1) Problématiques agrométéorologiques.....	20
2.3.2) Schéma conceptuel de données.....	21
2.3.3) Extrapolation/Interpolation des données météo .....	22
2.3.4) Création d'indicateurs agricoles spatialisés.....	22

<b>3) Les SIG sur Internet : État de l'art et standardisation.....</b>	<b>23</b>
3.1) Présentation de l'outil Internet .....	23
3.1.1) Histoire et évolutions.....	23
3.1.2) Les protocoles Internet.....	24
3.1.3) La relation client / serveur.....	24
3.1.4) L'Internet pour la diffusion cartographique.....	25
3.2) Inventaire des techniques de webmapping existantes.....	26
3.2.1) Solutions statiques vs solutions dynamiques.....	26
3.2.2) Les serveurs cartographiques.....	26
3.2.3) Les formats vectoriels.....	27
3.3) Les standards libres de diffusion de l'information géographique.....	28
3.3.1) Le SVG, outil de représentation de contenu vectoriel en 2D.....	28
3.3.2) Le PHP, lien entre la carte et la base de données.....	30
3.3.3) L'ECMA Script, agent d'interactivité avec l'utilisateur.....	31
<b>4) Conception et développement de l'application cartographique.....</b>	<b>32</b>
4.1) Concertation et conception de la carte interactive.....	32
4.1.1) Rappel des objectifs du projet Internet.....	32
4.1.2) Définition des fonctionnalités à développer.....	32
4.1.3) Schéma conceptuel du développement Internet.....	33
4.2) Construction du fond cartographique .....	34
4.2.1) Conversion des shape files en fichiers SVG avec Arc GIS 9.....	34
4.2.2) Mise en page des fichiers SVG : Le canvas ou viewbox.....	36
4.2.3) Design de la carte : Éléments de graphisme SVG.....	37
4.2.4) Mise en forme de l'image Landsat <i>quegaap.tiff</i> .....	38
4.3) Constitution d'une bibliothèque de fonctions ECMAScript.....	39
4.3.1) La compatibilité du SVG avec ECMAScript.....	39
4.3.2) Les fonctions Zoom et Pan.....	39
4.3.3) Les fonction <i>getData.js</i> .....	40
4.3.4) La fonction <i>getWidget.js</i> .....	40
4.3.5) Le fichier <i>getWidget.js</i> .....	41
4.3.6) Les autres fonctions javascript.....	41
4.4) Liaison à la base de données MySQL.....	42
4.4.1) Connexion du SVG a la base de données MySQL.....	42
4.4.2) Appel des valeurs sur la carte : exemple de la température.....	43
4.4.3) Appel des valeurs dans le widget : <i>prevision.php</i> .....	45

4.5) Développement de l'interface de numérisation.....	46
4.5.1) De la vue régionale au plan de ferme.....	46
4.5.2) Récupération et modification du script.....	46
4.5.3) Fonctionnement de l'application.....	47
4.5.4) Potentiel du SVG pour numériser en ligne.....	48
4.6) Instauration d'une zone membre et d'un accès privilège.....	49
4.6.1) Diffusion et niveaux de permission.....	49
4.6.2) Tests en local et obstacles à la mise en ligne.....	49
4.7) Intégration de l'ensemble dans une architecture XHTML conforme.....	50
4.7.1) Pourquoi XHTML et les CSS ?.....	51
4.7.2) Un SIG en ligne conforme ou valide ?.....	51
4.8) Bilan de la phase de développement Internet.....	52
Conclusion.....	53
Annexes.....	54
- Carte du Projet Mésonet Montréal.....	55
- Exemple d'un plan de ferme géoréférencé.....	56
- Schéma simplifié d'une station météorologique.....	57
- Code source : <i>mapmesonet.svg</i> ( extraits ).....	58
- Code source : La barre de navigation SVG / ECMAScript.....	60
- Code source : <i>zoom.js</i> .....	62
- Code source : <i>pan.js</i> .....	63
- Code source : La légende SVG / PHP .....	64
- Code source : <i>Temperature.php</i> .....	65
- Code source : <i>Previson.php</i> .....	66
- Code source : <i>index.php</i> ( xhtml / css ) .....	67
Bibliographie.....	68

## INTRODUCTION

---

La période de stage au Laboratoire GAAP s'est déroulée selon plusieurs phases de travail, ces dernières ayant suivi le cheminement logique d'un projet contingenté par une approche collective et participative. L'intégration à une équipe multiculturelle de géomaticiens / développeurs dans le contexte québécois en font une expérience hautement formatrice et inoubliable.

Des recherches bibliographiques et électroniques aux aspects de développement Internet, de nombreux aspects relatifs à la géomatique et à la cartographie numérique ont été abordés. C'est dans un cadre pluridisciplinaire que la démarche de stage s'est inscrite et les compétences à déployer, les outils à utiliser ainsi que les points forts de la mission ont été déterminés et fixés lors de mon arrivée au laboratoire. Cela premièrement dans l'objectif de cerner les objectifs pédagogiques du stage mais aussi de répondre aux besoins actuels du projet que j'ai intégré.

La position pionnière du laboratoire GAAP dans le domaine de géomatique agricole, la qualité des équipements et des ressources informatiques, ainsi que l'encadrement offert par le Docteur Alain Viau et l'informaticien Richard Picard, ont permis de dépasser les aspects strictement universitaires propres aux activités de recherche en laboratoire. En effet, toute la dynamique du stage fut fondée sur la conception et la création d'un produit destiné à être publié sur Internet, ce qui est venu stimuler et concrétiser le travail personnel en procurant des objectifs précis et en instaurant une finalité à atteindre.

Outre les principes et les concepts de la géomatique agricole, ainsi que les compétences métiers reliés à l'agriculture de précision et à l'environnement qui ont été acquises, ce stage concrétise une réelle spécialisation dans les technologies Internet appliquées aux systèmes d'information géographique (SIG). Bon nombre de concepts et d'habiletés de programmation pour la diffusion de l'information géographique ont été mis en œuvre de manière concrète. On désigne l'ensemble de ces activités par le terme relativement récent et générique de « webmapping ».

L'utilisation d'Internet dans la perspective de diffuser de l'information géographique, et les divers travaux de programmation menés ont aussi orienté la réflexion sur la situation des SIG dans le contexte de montée en puissance des technologies open source, ainsi que sur le potentiel du web sémantique vis-à-vis de la diffusion et du partage des données géographiques.

Ce rapport s'organise en quatre parties qui résument les apports théoriques et pratiques que le stage a procurés. La première partie s'attache à présenter les objectifs du stage et le laboratoire GAAP. Le moteur SIG et les bases de données sur lequel le projet repose sont ensuite présentés au sein de la deuxième partie. La troisième partie synthétise quant à elle les phases de recherche et d'apprentissage en webmapping, à travers un état de l'art de la cartographie numérique sur Internet.

Enfin, les travaux de conception et de développement Internet sont exposés au sein d'une dernière partie. La présentation du produit final est appuyée par l'explication des scripts principaux et de certaines fonctions ayant été développées. Le script intégral de l'application étant trop volumineux pour être inclus dans ce rapport, des extraits de codes sources choisis de manière ciblée sont placés en annexe. L'intégralité du script est de plus la propriété du laboratoire. Bien que fondée majoritairement sur des ressources libres, la période de développement représente 6 mois de travail et la mise en œuvre de compétences avancées de programmation, ce qui justifie cette protection par copyright.

## 1) Présentation du stage

### 1.1) La structure d'accueil

#### 1.1.1) Nature et raison sociale du laboratoire GAAP

Le laboratoire de Géomatique Agricole et Appliquée (GAAP) est une institution de recherche de la Faculté de Foresterie et de Géomatique de l'Université Laval, à Québec. Il dépend du Département des Sciences Géomatiques de l'Université et fut fondé en avril 2001. Il est aussi associé au Centre de Recherche en Géomatique (CRG) et au réseau d'excellence GEOIDE, lesquels regroupent depuis 1989 les laboratoires de quatre universités québécoises (l'Université Laval, l'Université de Sherbrooke, l'Université du Québec à Montréal et l'Université Mac Gill de Montréal).

La mission du laboratoire est d'assurer le développement de la géomatique dans le secteur de l'agriculture, de l'environnement, de la gestion de l'eau ainsi que dans l'étude du territoire par bassin versant. Les travaux du laboratoire visent aussi à favoriser le développement de l'usage des technologies à référence spatiale, comme par exemple les SIG, le GPS, les outils de traitement d'images, ou d'autres géotechniques permettant d'utiliser l'information géographique comme un outil d'aide à la décision dans les domaines de l'agriculture de précision et de l'environnement.

Les activités de recherche sont principalement liées à l'usage des SIG et de la télédétection pour la gestion des territoires et des exploitations agricoles, mais comportent aussi bien souvent des aspects propres aux disciplines connexes, à savoir la météorologie, l'hydrologie, la foresterie ou encore l'écologie. Il est possible de qualifier l'ensemble des activités du laboratoire comme ayant trait à la géomatique environnementale, mais les recherches appliquées à l'agriculture prévalent largement (agronomie, agrométéorologie, agroforesterie...).

#### 1.1.2) Hiérarchie et fonctionnement

Dirigé par le Docteur Alain Viau, le laboratoire regroupe 5 à 6 chercheurs et ingénieurs de recherche, ainsi qu'une dizaine de chercheurs associés provenant d'horizons variés. Le parc informatique et le réseau sont gérés par Richard Picard (informaticien administrateur réseau) et le SIG est par Gilles Lavoie (géomaticien responsable SIG). Les travaux liés à l'optique et à la télédétection sont quant à eux menés par Jae Dong JANG (chercheur post-doctoral).

Les activités de recherche sont aussi clairement impulsées par la présence et l'investissement de doctorants, de chercheurs post-doctorants, d'étudiants de maîtrise, de contractuels et de stagiaires, lesquels réalisent une partie de leur cursus universitaires ou professionnels en intégrant les différents projets du Laboratoire GAAP. Ils contribuent à ses avancées technologiques ainsi qu'à sa production scientifique.

L'équipe du GAAP est enfin largement mobile et évolutive. Outre les déplacements fréquents du Directeur et des Doctorants en Europe et dans le Monde, la plupart des membres du laboratoire effectuent une partie de leur travail sur le terrain, qu'il s'agisse aussi bien de relevés GPS, de campagne d'orthophotographies, d'observations et d'inventaires ou encore de rencontre avec les acteurs des différents projets.

### 1.1.3) Collaborations et financements

En tant que structure de recherche innovante de l'Université Laval, le laboratoire GAAP est financé à la fois par le gouvernement provincial québécois et par le gouvernement fédéral canadien. Il perçoit notamment des fonds du Ministère des Ressources Naturelles du Québec, ainsi que des crédits de recherche provenant du Fond Canadien d'Innovation.

Au delà des importants efforts de collaboration interuniversitaire et/ou internationale qui lui permette d'intégrer ou d'initier des projets de géomatique agricole d'envergure, le laboratoire entretient donc des rapports privilégiés avec les instances gouvernementales, mais aussi avec un certain nombre de collectivités locales et d'entreprises privées.

Le Docteur VIAU est fondateur et titulaire de la Chaire de Recherche Industrielle en optique de précision et en géomatique environnementale (OGEN). Ce type de regroupement est unique en Amérique du Nord et regroupe des acteurs variés. Le laboratoire collabore par exemple avec la Financière Agricole (Institution québécoise de financement et d'assurance agricole), avec les entreprises TELOPS et INNOTAG (télédétection) ou encore avec la coopérative agroalimentaire DYNACO. Une part des activités du laboratoire est donc étroitement liée aux intérêts des acteurs de la Chaire, et cette mise en commun de compétences représente un apport financier non négligeable pour le laboratoire GAAP.

Par ailleurs, le laboratoire est membre actif du Regroupement HYMAP, lequel est dédié à la mise en œuvre de la politique canadienne de l'eau d'une manière intégrée et durable. Celui-ci regroupe 49 chercheurs provenant de 9 universités canadiennes et travaillant tous dans le domaine de la gestion de l'eau, mais selon des centres d'intérêt différents. Un certain nombre de projets du laboratoire, comme le projet Mésonet par exemple, sont donc impulsés et chapotés par HYMAP.

Le laboratoire GAAP mène aussi des actions de recherche avec des institutions et laboratoires européens. A titre d'exemple, il collabore avec des organisations françaises telles que l'Institut de Recherche pour le Développement (IRD), le Centre de coopération internationale en recherche agronomique pour le développement (CIRAD) ou encore le Centre National du Machinisme Agricole, du Génie Rural, des Eaux et Forêts (CEMAGREF).

D'importants travaux ont notamment été réalisés dans le cadre du programme « Géo-traceagri », lequel réunit le Québec, la France et le Benelux dans un projet de traçabilité géographique appliquée à l'agriculture (<http://www.geotraceagri.net/>). Cet exemple de collaboration internationale démontre la volonté du Laboratoire de mettre en œuvre des partenariats scientifiques, mais prouve aussi ses ambitions quant à l'exportation de ses compétences et de son expertise en géomatique agricole.

#### 1.1.4) Ressources et équipements

Afin de mener ses différents projets et de répondre aux appels d'offres de ses divers partenaires, le laboratoire GAAP se doit d'investir dans les technologies de pointe reliées à la géomatique et à l'informatique. Les équipements nécessaires sont présentés ici de manière succincte afin d'apporter des précisions quant à l'environnement de travail dans lequel le stage s'est déroulé.

Le parc informatique représente l'outil de travail principal du Laboratoire. Il se constitue d'une trentaine de puissants ordinateurs connectés en réseau, lequel est géré à la fois par un système Windows NT et par une distribution Linux de type Fedora 4.0. Cette combinaison des systèmes d'exploitation permet de mettre en oeuvre toutes les opérations et les routines d'administration nécessaires aux projets géomatiques. Les machines sont aussi connectées aux différents serveurs du laboratoire, administrés eux aussi à la fois par une solution Windows Server et par des solutions open-source, à l'instar de SQL server et du serveur Apache.

Les principaux outils logiciels SIG, tous sous licence multipostes et multi-utilisateurs, sont Arc Info, Arc GIS 9, Arc View 3.2 et Surfer 8. Le laboratoire dispose bien sûr d'autres logiciels comme MatLab, Adobe Photoshop ou d'autres outils dédiés au calcul algorithmique, à l'infographie ou à la bureautique.

Le second outil, ou « ressource » est bien sûr l'ensemble des données géographiques et statistiques manipulées par le laboratoire, qui proviennent de sources variées. Les principaux fournisseurs de données spatiales sont le Ministère des ressources naturelles du Québec ainsi que le Ministère de l'environnement canadien, qui je tiens à le préciser propose des prix inférieurs à ceux pratiqués par l'IGN dans le cas français. Le laboratoire GAAP investit donc régulièrement auprès de ses institutions pour acquérir des données à jours, essentiellement des fichiers de formes, des couvertures thématiques et des images satellitaires

De plus, le GAAP acquiert lui même des données pour subvenir à ses besoins de recherche, et ce de plusieurs manières. Il mène premièrement des campagnes de relevés GPS sur le terrain, doté de tout le nécessaire GPS (récepteurs, antennes, théodolites...) et d'un véhicule 4x4 de type « pick-up » pour pouvoir accéder à tous les types de terrains.

Enfin, et c'est un investissement de taille, des campagnes de photographies aéroportées sont aussi menées de manière annuelle. Le GAAP est équipée de caméras numériques haute résolution et d'un ordinateur de bord, qu'il utilise lorsqu'il peut se procurer un avion et que les conditions atmosphériques sont clémentes. Il dispose ainsi d'images hyper-spectrales, infrarouges ou proches infrarouges multi-vues du territoire, ce qui s'avère très intéressant pour l'analyse détaillée du territoire agricole et la traçabilité géographique des cultures.

## 1.2) Présentation du projet Mésonet Montréal

### 1.2.1) Nature du projet

Mésonet-Montréal est un projet de recherche en météorologie et dans les domaines connexes, initié par le Regroupement HYMAP. Impulsé notamment par la réunion du laboratoire GAAP et du Laboratoire de Météorologie Physique et de Télédétection de l'Université Mac Gill, la constitution progressive du Mésonet est basée sur le développement d'un réseau de stations météorologiques automatisées sur le territoire Québécois. Celui-ci est géré et entretenu par le Service Météorologique Canadien, qui est un organe du Ministère de l'Environnement Canadien.

Le terme Mésonet, d'origine anglo-saxonne, résulte de la contraction des mots « Meso-scale » et « Network », et désigne ainsi un réseau établi à échelle méso. C'est le cas du Mésonet Montréal, puisque celui-ci comporte 39 stations météorologiques proches les unes des autres (plus ou moins 30 km) et établies dans un rayon approximatif de 100 kilomètres au sud de Montréal. C'est en Montérégie, principale région agricole du Québec, que le réseau prend place, et cette localisation correspond par ailleurs avec le champ d'observation du radar météorologique de l'université Mac Gill.

Les sites choisis et les installations faites permettent d'ores et déjà aux différents acteurs du projet de bénéficier de relevés météo en temps réels, ce qui multiplie les applications possibles. On peut aussi préconiser une expansion éventuelle du rôle que le Mésonet-Montréal pourra jouer. Par exemple, l'étude des précipitations hivernales et de la détection du verglas ou encore les problématiques liées à la qualité de l'air dans la grande région de Montréal pourront bénéficier des apports que procurent les stations. Les intérêts du projet sont donc multiples et le concept même de réseau agrométéorologique est relativement nouveau en Amérique du Nord.

Notons toutefois l'existence de projets similaires aux États-Unis, développé notamment par les Universités du Texas et de l'Oklahoma. Ces Mésonet s'appuient également sur des réseaux de stations automatisées, mais leurs objectifs restent généralement cantonnés aux prévisions et à la recherche en météorologie ou en physique atmosphérique. Ces réseaux diffusent l'information via par exemple des applications Mapserv sur Internet, comme par exemple à l'URL suivante : <http://www.mesonet.ou.edu/public/current.html>.

Les Mésonet américains ont clairement insufflé l'idée du Mésonet-Montréal, mais ce dernier se distingue par les applications envisagées. Les activités communes du Laboratoire GAAP et du Laboratoire de Météorologie Physique et de Télédétection constituent ainsi un réseau de compétences pour mieux appréhender les aléas climatiques relativement extrêmes au Québec, ainsi que leurs impacts sur l'agriculture et sur l'environnement en général.

### 1.2.2) Aspects collaboratifs et financiers

Soutenu par le Ministère de l'Environnement canadien ainsi que par le Ministère des Ressources Naturelles du Québec, le principal apport financier du Mésonet provient du Fond Canadien d'innovation, qui alloue respectivement 525 000\$ canadiens à la télédétection atmosphérique et un 550000\$ canadiens pour la géomatique agricole. Il s'agit donc d'un projet de recherche largement favorisé par des financements fédéraux et provinciaux, dont les retombées vise l'ensemble des acteurs de l'agrométéorologie, et l'ensemble de la communauté agricole québécoise à plus long terme.

Les activités de recherche du Laboratoire relatives au Mésonet sont aussi étroitement liées aux intérêts de la Financière Agricole du Québec, présentée plus haut. Celle-ci n'intervient pas financièrement dans le projet Mésonet mais a déjà collaborer avec le Laboratoire dans d'autres projets, et est fortement susceptible d'être intéressée par les services agrométéorologiques dont la mise en place est prévue (Notamment vis-à-vis du potentiel du Mésonet en tant qu'outil de prévention des intempéries et de suivi des cultures).

D'autres collaborations permettent la mise en place et l'entretien du réseau de stations, notamment grâce à l'engagement de municipalités et d'agriculteurs impliqués dans le projet. C'est en effet sur des territoires municipaux ainsi que sur des parcelles agricoles privées que sont installées la majorité des stations. HYMAP et le Laboratoire GAAP ont dû par conséquent solliciter l'intérêt des agriculteurs et des décideurs locaux, en menant divers sondages et phases de concertation publique.

Étant donné les nombreuses possibilités que les équipements du Mésonet peuvent procurer, un certains nombres de partenaires privés se sont également joint au projet. On notera par exemple les apports non négligeables du groupe Hydro Québec (l'équivalent québécois d'EDF-GDF) qui a investi quelques 200000 \$, ou encore ceux de la coopérative agricole DYNACO et du Consortium PRISME, dont la part du financement reste inconnue.

C'est donc dans un contexte d'économie mixte que se met progressivement en place le réseau. Les principaux financements sont maintenant acquis et la plupart des stations fonctionnelles. Cependant, l'avenir du réseau est en question, notamment vis-à-vis de son financement et de sa gestion. Il est possible que la Financière Agricole reprenne la responsabilité du Mésonet et l'élargisse en implantant des stations supplémentaires. Cette hypothèse pourrait donc déboucher sur un changement de statut du réseau dans un avenir relativement proche, tout en conservant les applications pour la recherche et la collaboration avec Environnement Canada.

### 1.3) Présentation de la mission

#### 1.3.1) Sujet de stage

Le sujet du stage s'intitule « développement de services web géographiques dédiés à la consultation et au traitement des données géospatiales et agrométéorologiques sur Internet ». Cet énoncé est directement relié aux besoins immédiats du projet Mésonet-Montréal, et contient plusieurs concepts qu'il convient d'expliquer avant d'entrer dans le vif du sujet.

La notion de service web (webservice) désigne tout d'abord un ensemble de protocoles et de normes utilisés pour échanger des données entre des applications. Des logiciels ou des sites Internet, écrits dans divers langages de programmation et sur diverses plateformes, peuvent en effet employer des services web pour échanger des données à travers des réseaux informatiques comme Internet. Cette interopérabilité est dûe à l'utilisation de normes ouvertes dites « libres ». L'OSI (Open Source Initiative) et le W3C (World Wide Web Consortium) sont les comités de coordination responsables notamment de l'architecture et de la standardisation des services web au niveau planétaire.

Dans le cas du Mésonet, l'objectif est de rendre disponibles les données météorologiques d'une zone donnée, sur un site Internet cartographique et en temps réel, via l'utilisation de services web. Le but des recherches menées pendant le stage est donc de concevoir des services web géographiques, destinés aux chercheurs, aux agriculteurs et aux gestionnaires en agronomie et en agrométéorologie. Moyennant une adhésion et une éventuelle participation financière, tout agriculteur ou chercheur intéressé sera en mesure de consulter un ensemble d'informations météorologiques et hydrologiques relatives à sa zone d'exploitation ou de recherche, via le site Internet. La mise en place d'un certain nombre d'indices de prévision et de planification agricole, d'une interface de numérisation fonctionnelle en ligne sont aussi envisagés dans un avenir assez proche.

De manière concrète, la mission consiste à élaborer un site Internet cartographique selon les normes établies, en réponse à des besoins de consultation/gestion des utilisateurs. Cette étape du projet dans laquelle j'interviens est dédiée à la publication de l'information géographique et agrométéorologique contenu dans les bases de données auprès du grand public. La création de liens entre le SIG du laboratoire et Internet induit la création de services web à composante spatiale, capables de restituer les données en temps réel et sur n'importe quel type de plateforme informatique.

Les analyses spatiales et le calcul d'indices agrométéorologiques étant en cours de réalisation sur le SIG du Laboratoire, la réalisation du site Internet s'est avérée indispensable pour répondre aux besoins immédiats des utilisateurs en terme de consultation des données. Le projet Mésonet Montréal se devait de disposer d'un site Internet accessible, et c'est l'une des raisons pour lesquelles j'ai été engagé comme stagiaire. Même si toutes les fonctionnalités du projet ne sont pas encore au point ou pleinement abouties, le travail réalisé servira de base aux développements futurs, et sera enrichi d'indices et de fonctionnalités supplémentaires au fur et à mesure de l'avancement des recherches géomatiques et algorithmiques.

### 1.3.2) Méthodes de travail

Comme je l'ai déjà évoqué dans l'introduction, la période de stage s'est organisée en plusieurs étapes de travail, et mon intégration au projet Mésonet a impliqué diverses collaborations. J'ai bien sûr communiqué avec l'ensemble de l'équipe pluridisciplinaire travaillant sur les divers aspects du projet (notamment au cours de réunions ou d'ateliers spécifiques), mais des rapports privilégiés ont été développés au fur et à mesure du stage.

Les aspects administratifs et pédagogiques de la mission ont été assumés par le Docteur VIAU. La prise en main des bases de données spatiales et des logiciels SIG s'est quant à elle déroulée avec l'aide du géomaticien. Enfin, l'installation de la station de travail ainsi que la majeure partie du travail de développement ont été effectuées avec l'informaticien, en compagnie duquel j'ai beaucoup appris.

Les deux premiers mois du stage ont été consacrés d'une part à la présentation et à la compréhension du projet, et d'autre part à une phase de prospection et d'apprentissage relative aux techniques de webmapping existantes et aux outils disponibles. Cette période de travail a notamment impliqué une série de test, à savoir un essai avec le serveur cartographique Mapserver, quelques manipulations sur Arc IMS, d'autres essais à l'aide de la librairie de script open-source Géoclient, ainsi que des bribes de développement en SVG et ECMAScript.

Les 4 mois suivants ont ensuite été consacrés au développement du SIG web à proprement parler, ce dernier ayant impliqué la conception et la mise en oeuvre de différentes fonctions et de scripts pour atteindre les objectifs fixés. De plus, certains temps forts de la phase de développement ont été particulièrement formateurs, à savoir premièrement le déploiement de l'application SVG à l'aide d'Arc GIS 9, puis la mise en place de liens entre la base de données et le site Internet. Celle-ci a notamment nécessité l'utilisation du langage PHP, lequel fait largement appel aux concepts de la programmation orientée objet et aux notions d'héritage et de fonctions natives.

L'ensemble de la période de développement a été très évolutive, comportant impasses, obstacles et avancées significatives. La mise en place du SIG en ligne a été menée conjointement aux travaux de Richard Picard, lequel m'a guidé et conseillé à maintes reprises. Nous avons en fait réalisé chacun un site Internet, en partageant et reprenant les mêmes fonctionnalités, mais en empruntant des méthodes et des designs différents. Il s'est de son côté plus penché sur la mise en place des fonctions PHP ainsi que sur l'automatisation des bases de données en Perl, tandis que je me suis davantage intéressé au format SVG et ses interactions avec l'ECMAScript. J'ai aussi orienté mes travaux vers la conformité du code source, en insistant sur l'importance du respect des normes du W3C pour optimiser l'accessibilité du SIG en ligne.

Mon site a finalement été mis en ligne sur un sous domaine du site officiel du Mésonet, car ce dernier relève de la responsabilité de l'informaticien du laboratoire. Le produit final de mon stage constitue donc une annexe au projet, mais m'aura néanmoins permis de comprendre l'ensemble du moteur SIG et du système Internet (tout langages confondus). Cette vue d'ensemble et la synthèse du projet géomatique permettent aujourd'hui de proposer un exemple de site interactif et opérationnel basé sur des technologies et des langages modernes (<http://www.mesonet-montreal.ca/cartonico>).

### 1.3.3) Outils privilégiés

Afin de répondre aux besoins d'interopérabilité et de portabilité de l'application, le choix d'utiliser des solutions open source s'est rapidement avéré comme pertinent. Celui-ci a notamment été impulsé par le Docteur VIAU, pour des raisons économiques et pratiques, puis approuvé par l'informaticien et moi même. On peut aussi clairement percevoir par ce biais un choix éthique, notamment quant au refus du Directeur de développer le site Internet via la solution ArcIMS de la société ESRI, dont le laboratoire possède pourtant la licence. La volonté de ne pas dépendre totalement d'outils commerciaux représente ainsi une avancée expérimentale pour la suite de ce projet. C'est aussi et surtout un moyen d'évaluer le potentiel des outils libres en matière de webmapping, dans le contexte de montée en puissance de l'open source dans le domaine de la géomatique, et dans les technologies informatiques en général.

L'avantage des solutions commerciales, outre leur rapidité de mise en oeuvre et l'automatisation de certaines tâches qu'ils procurent, réside dans le fait qu'elles sont bien souvent accompagnées de services après-vente et de supports techniques (l'exemple du support technique par téléphone d'ESRI en est la meilleure illustration). Ce n'est pas le cas pour les logiciels ou les langages libres, lesquels sont en développement constant et résultent bien souvent du travail de développeurs bénévoles ou de communautés d'utilisateurs. Cela implique donc premièrement d'écrire le code source de l'application dans son intégralité, et deuxièmement de trouver des substituts de supports techniques.

Dans notre cas, les nombreux sites dédiés au SVG et au PHP, et au développement web en général, ainsi que les forums de discussion ont été une aide précieuse. J'ai notamment communiqué plusieurs fois par courrier électronique avec Andréas Neumann, auteur du site « Cartonnet, cartographes on the web » et spécialiste du SVG, afin de trouver des solutions aux obstacles rencontrés. Le groupe de discussion Yahoo « SVG developpers », auquel je me suis inscrit a aussi été une source d'informations non négligeable. Il est aussi à souligner que les recommandations SVG et ECMAScript du W3C ont été consultées à maintes reprises afin de produire un code propre et bien formé.

Par ailleurs un certain nombre de fonctions open-source, surtout en ce qui concerne le PHP et l'ECMAScript ont été récupérées sur Internet, provenant soit de banques de scripts libres d'utilisation soit de tutoriaux divers, puis adaptées à nos besoins de développement. Les fonctions PHP étant à la base du fonctionnement des services web ont quant à elles été développées intégralement par Richard Picard, puis reprises et adaptées aux besoins spécifiques de mon site Internet.

Les outils logiciels libres utilisés pour la création du site Internet, ont été principalement EasyPHP et PHPMyAdmin (outils de développement orientés serveur comprenant le serveur Apache, l'interpréteur PHP et les outils de gestion MySQL). D'autres outils open source ont aussi été utilisés par Richard Picard, comme par exemple l'IDE (Integrated Development Environment) Komodo pour la réalisation des fonctions Perl.

Enfin, l'intégralité du code a quant à elle été rédigée avec l'éditeur Ultra-Edit, lequel est délivré sous licence GPL par la société IDM. Basé sous un noyau open-source, c'est un shareware très appréciable pour le développement de pages HTML ou PHP.

## 2) Le moteur SIG agrométéorologique

### 2.1) Le réseau de stations

#### 2.1.1) Localisation et caractéristiques

L'installation d'un réseau de stations météorologiques requiert de grands efforts de planification et de coordination au niveau technique et scientifique. Une importante phase de concertation et l'intervention de plusieurs personnes (chercheurs, experts, décideurs, gestionnaires, techniciens...) ont été des facteurs nécessaires à la sélection de procédures optimales pour l'implantation rapide et efficace des stations du Mésonet-Montréal.

Le réseau se compose de 35 stations distantes en moyenne de trente kilomètres les unes des autres. Quinze d'entre elles appartiennent à l'Université Laval et les 20 autres sont la propriété de l'Université Mac Gill, mais l'ensemble du réseau est géré et entretenu par le Service Météorologique du Ministère de l'Environnement Canadien.

Le principal objectif de recherche des stations de l'Université Mac Gill est de fournir des données météorologiques ponctuelles permettant la validation et la calibration du radar météorologique de Sainte-Anne De Bellevue. Ces stations ont été dispersées à intervalle régulier de 25 Km dans la zone de couverture du radar. Les sites sont généralement établis sur des terrains municipaux.

Dans le cas des stations de l'Université Laval, celles utilisées par le laboratoire, les objectifs de recherche sont davantage orientés vers l'hydrologie, la météorologie et l'agrométéorologie. Le bassin versant de la Rivière Des Anglais, en Montérégie, a été choisi pour accueillir une plus grande concentration de stations. Dans cette région, les stations y sont installées à intervalle de 5 à 10 kilomètres. Les stations de l'Université Laval sont localisées en bordure de champs, sur les propriétés d'agriculteurs partenaires du regroupement.

Les stations occupent typiquement un terrain plat gazonné d'environ 15 par 25 mètres (50 par 75 pieds). L'équipement le plus haut est un mât basculant de 10 m requis pour les données éoliennes. Cette hauteur de mesure du vent est établie conformément aux normes de l'Organisation Mondiale de la Météorologie. Les instruments sont installés sur des bases distantes de 8 mètres. Chaque base est constituée d'un poteau d'aluminium de 3 pouces de diamètre fixé sur un pieu visé ou sur une base de béton (indépendamment du type de sol rencontré).

Enfin, l'installation d'une station représente un investissement d'environ 50 000 \$, comprenant l'achat des instruments et leur mise en place, ainsi que les frais induits par les travaux d'excavation et de mise à niveau du terrain. L'entretien des stations nécessite quant à lui des coûts variables, en fonction des dégâts matériels pouvant être occasionnés durant la période hivernale et de l'usure générale des équipements. La fourniture de l'électricité et du téléphone, et la coupe régulière du gazon durant l'été et l'automne représentent quant à elles des frais fixes inhérents au projet.

### 2.1.2) Observations et automatisation

Chaque station regroupe un ensemble d'instruments de mesures, et enregistre un certain nombre d'observations dont la nature peut varier en fonction d'une part de l'environnement immédiat de la station et d'autre part selon les caractéristiques culturelles de la zone qu'elle recouvre. Les 35 stations ne disposent pas systématiquement des mêmes capteurs ou instruments, mais elles permettent toutes de mesurer au minimum la température, les précipitations, l'humidité relative et la vitesse du vent en des points donnés.

L'ensemble des instrumentations sont reliées à un système d'acquisition des données (datalogger) par tranchées souterraines, lequel est branché à un modem téléphonique relié au réseau informatique d'Environnement Canada. Le Ministère récupère les données du datalogger toutes les 5 minutes en moyenne par l'intermédiaire d'Internet. Les données sont au format ASCII et dirigées vers un serveur FTP ministériel hautement sécurisé. Depuis ce serveur, elles subissent ensuite un contrôle qualité (utile à la fois pour repérer les stations défectueuses et pour déterminer les mesures anormales ou non effectuées). Les données contrôlées sont ensuite mises à la disposition des acteurs autorisés de la météorologie ou des disciplines connexes, tel est le cas du laboratoire GAAP.



L'exemple de la station météorologique d'Alfred (Ontario, Canada).

## 2.2) Les bases de données

### 2.2.1) La base de données météorologiques

Comme je l'ai précisé précédemment, le laboratoire est autorisé à télécharger les données météo d'Environnement Canada. Ces mesures étant fortement évolutives, le laboratoire doit les télécharger plusieurs fois par jour, en moyenne toutes les dix minutes, s'il veut bénéficier de données à jour.

Cette tâche n'étant pas matériellement réalisable par le seul informaticien, celui-ci a récemment développé un système d'acquisition automatique des données. De complexes requêtes en langage Perl (Practical Extraction and Reporting Language), ont donc été mises en places sur le serveur du GAAP. Ce langage puissant basé sur les langages C et Shell permet l'écriture de scripts CGI (Common Gateway Interface). Il est très adapté pour la manipulation de chaînes de caractères et de fichiers. Les requêtes Perl fonctionnent parallèlement à l'utilisation des commandes Unix *Cron Tab* et *Batch File*.

Grâce à cette « boucle » réalisé par Richard Picard, le serveur GAAP va donc récupérer automatiquement les données météo sur le FTP d'Environnement Canada à intervalles de temps réguliers. A la manière d'une commande *ping*, la requête vérifie l'accessibilité au FTP ainsi que la présence de données, puis déclenche le transfert des données vers le serveur GAAP. Cette routine d'administration s'effectue six fois par heure, 24 heures sur 24 et 365 jours sur 365.

Les données sont acquises au format ASCII puis stockées dans une base de données SQLServer hébergée sur le serveur du laboratoire. N'étant pas utilisables telles quelles, d'autres routines Perl s'occupent de convertir les données aux formats désirés. Dans le cas du projet Internet, c'est au format XML que la conversion s'effectue, afin d'une part de répondre aux standards de stockage et de diffusion en vigueur, et d'autre part de disposer de données relativement faciles à intégrer au SIG en ligne.

En ce qui concerne l'intégration des données dans le SIG du laboratoire, il n'y a pas de conversion de format. Les données ASCII sont directement implémentées dans des bases de données, généralement sous environnement Microsoft Access. Tous les membres du laboratoire ayant accès au réseau, ils peuvent alors sélectionner les données d'une station ou d'un groupe de station correspondant aux périodes auxquelles leurs différentes analyses s'intéressent. Il ne leur reste ensuite plus qu'à intégrer localement les données dans le système de gestion de base donnée de leur choix. La plupart des chercheurs utilisent Access pour cette tâche, mais il serait aussi parfaitement possible de le faire avec des outils open-source, comme PostGreSQL par exemple.

L'évolutivité caractéristiques des données météorologiques rend ces téléchargements automatisés obligatoires, et cela représente un volume très important de données à stocker. Réalisés manuellement de 2002 jusqu'à Juillet 2005, le nombre d'enregistrement a d'ores et déjà dépassé le nombre de 100 000 000. Pour plus de facilité, les données sont archivées par station au sein de dossiers annuels, et de sous-dossiers mensuels. Il est alors simple de retrouver les données d'une station donnée pour une période donnée.

### 2.2.2) La base de données spatiales

La base de données spatiale relative au projet Mésonet contient des données de natures variées, correspondant aux divers besoins d'analyse des différents axes du projet.

Les données principales et de référence sont les fichiers vectoriels que le laboratoire achète régulièrement au Ministère des Ressources Naturelles du Québec. Les fichiers de formes sont délivrés sous la forme de couvertures Arc Info, que le géomaticien convertit ensuite aux formats désirés. Ils fournissent une grande variété d'informations sur le territoire, qu'il s'agisse de limites administratives ou cadastrales, de parcelles agricoles, de réseaux routiers ou hydrographiques, de données altimétriques ou de tout autre objet présent sur le territoire et utiles aux analyses spatiales environnementales et à la gestion de territoire agricole par bassin versant.

Le Laboratoire acquiert aussi des fichiers rasters délivrés par le même Ministère. Ceux-ci sont principalement des images provenant des satellites Landsat, Spot, Radarstat ou Ikonos. De plus, l'achat d'ortho-photographies argentiques ou numériques vient enrichir la variété d'informations disponibles, et permettent de disposer d'une précision accrue concernant le territoire québécois, et spécialement à propos des zones agricoles de la Montérégie.

Comme il a été dit dans la présentation du laboratoire, ce dernier acquiert également des ortho-photographies par ses propres moyens. Concernant le projet Mésonet, une campagne de vols aéroportés a été planifiée de mars à juin puis effectuée en juillet 2005. Équipé d'une caméra haute résolution Dulkan MS4100 et d'un ordinateur portable spécialement conçu pour être embarqué à bord de l'avion, le laboratoire a donc photographié l'ensemble de la zone d'étude, selon différentes résolutions et à l'aide de différents filtres étant placés devant l'objectif. Ainsi, il dispose à présent d'images hyper-spectrales, infrarouges ou proche infrarouges multi-vues de la Montérégie, qu'il utilise pour la télédétection et la photo-interprétation. L'exemple le plus parlant étant un sous projet du projet Mésonet basé sur la caractérisation des pommiers et la détection des carences dans un bassin versant contenant un grand nombre de vergers. Les images sont utilisées pour les analyses spatiales et agricoles, et soumises à des filtres d'identification, le plus souvent développés en langage C ou Fortran. De plus, les images multi-vues récemment acquises s'avèrent bénéfiques pour la réalisation de modèles numériques de terrains (MNT) très précis.

Une campagne de relevés GPS sur le terrain a aussi été menée, avec l'aide de géomaticiens et d'arpenteurs contractuels que le laboratoire a spécialement engagé. Les relevés ont été acquis au format CSV (Coma Separated Values) à l'aide de GPS et d'antennes portables, corrigées, puis implémentés dans le SIG à l'aide des fonctionnalités GPS d'Arc GIS 9. Il s'agit surtout de relevés caractérisant les parcelles agricoles, les stations météorologiques et les instruments du réseau Mésonet.

Enfin, l'ensemble des données vectorielles et matricielles du laboratoire sont administrées par le géomaticien et stockées sur des disques durs spécifiques, prenant l'air d'entrepôt de données géospatiales, tant le volume de données est important. Les données du Mésonet en font partie, et sont accessibles par tous les membres du laboratoire, comme pour les données météo. Ceux-ci les acheminent localement via le réseau et constituent diverses « geodatabases personnelles » sur Arc Gis 9, en fonction de leurs besoins d'analyse.

## 2.3) Structuration du SIG

### 2.3.1) Problématiques agrométéorologiques

A la fois outil de recherche pour le laboratoire GAAP et ses partenaires et outil d'aide à la décision pour les agronomes et les agriculteurs, le Mésonet est un projet de géomatique environnementale évolutif. Basé sur un modèle de gestion des données défini, il est cependant ouvert sur de nombreuses applications et propositions des différents acteurs concernés, ce qui le rend flexible et adaptable.

L'idée prédominante du projet, ou du moins l'idéal vers lequel les recherches tendent, est celle de traçabilité géographique des cultures. Le concept de traçabilité caractérise l'aptitude à retracer l'historique, l'utilisation ou la localisation d'une entité au moyen d'identifications enregistrées. La traçabilité géographique permet donc d'identifier et de suivre spatialement un produit agricole depuis sa mise en terre jusqu'à sa récolte.

L'évolution biologique et le bien-être des cultures étant intimement lié aux conditions météorologiques, les données des stations du Mésonet procurent des informations précieuses pour surveiller et prévoir l'état des champs cultivés. Simples mesures météorologiques ponctuelles à l'état brut, elles prennent tout leur sens et permettent d'établir différents calculs utiles à la gestion lorsqu'elles sont implémentées dans le SIG.

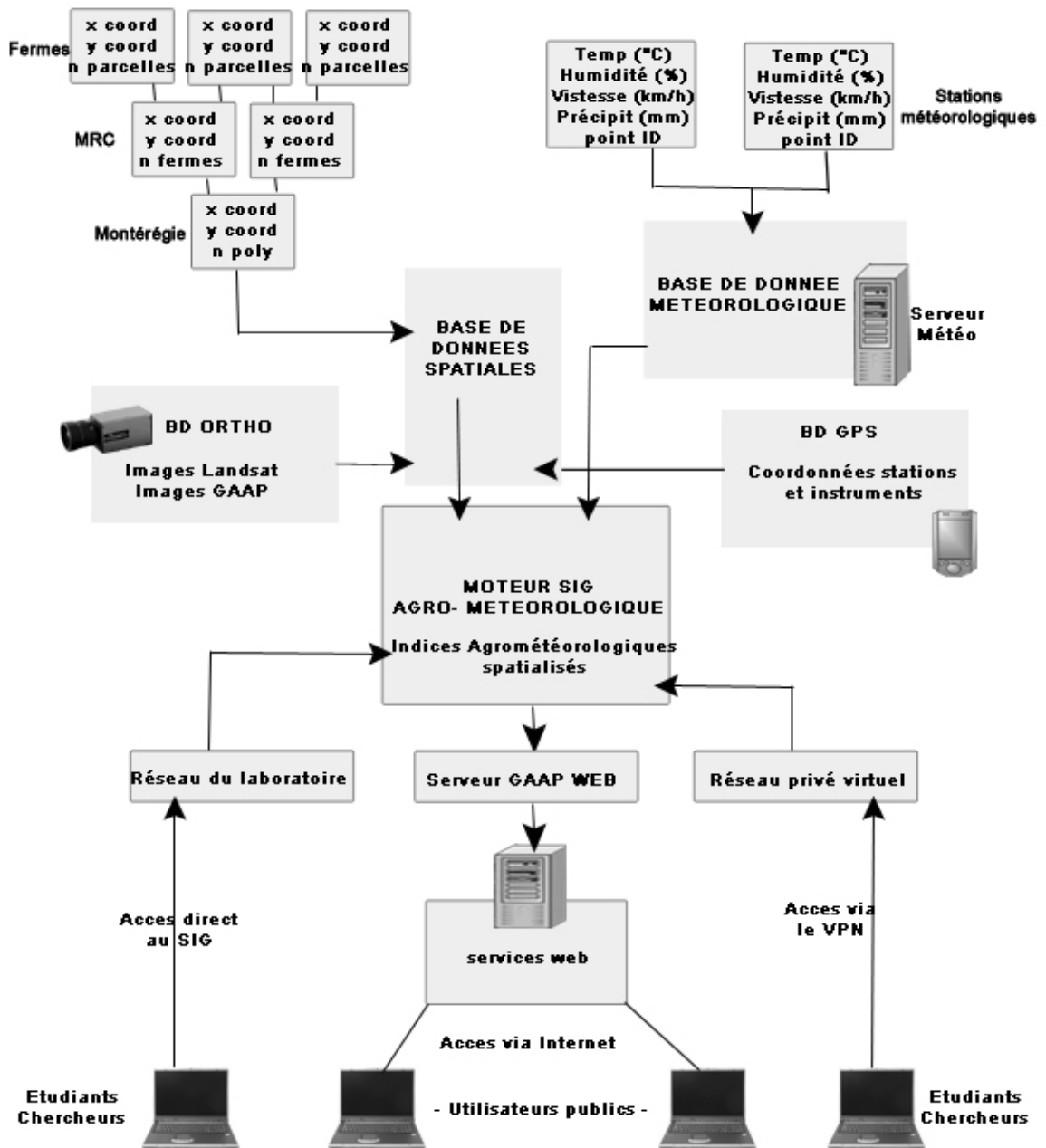
La problématique spatiale réside dans l'application de mesures ponctuelles à des entités surfaciques que sont les parcelles agricoles. Il s'agit donc d'être capable d'interpoler ou d'extrapoler les données météo afin d'établir scientifiquement des correspondances spatiales. Si chacune des parcelles est caractérisée spatialement comme entrant dans le rayon de représentativité des mesures de telle ou telle station, il est alors possible de leur assigner les données météo correspondantes.

### 2.3.2) Schéma conceptuel de données

Le schéma conceptuel de données présenté à la page suivante procure une vision d'ensemble du SIG dédié au projet Mésonet-Montréal. Son objectif est de proposer une organisation simplifiée du fonctionnement du moteur SIG agrométéorologique, des bases de données jusqu'à l'utilisation du SIG.

La réalisation du schéma est basée à la fois sur le schéma conceptuel de données du laboratoire GAAP et sur les différents aspects propre au projet Mésonet, comme la base de données images (BD ORTHO), la base de données GPS ou encore l'interface web. Il synthétise la compréhension du système que j'ai développé au début de mon stage.

Il ne prend pas en compte le détail des services Internet, qui font l'objet d'un schéma présenté dans la partie 4, puisque ces derniers constituent l'objectif majeur de la période de stage.



### 2.3.3) Extrapolation/Interpolation des données météo

La mise en place de couples parcelles/stations représentatifs nécessite à la fois de prendre en compte les composantes spatiales du terrain (relief, hydrologie, exposition, trous d'airs, présence de haies coupe-vent, microclimats caractéristiques...), et également les caractéristiques culturelles et phytosanitaires des parcelles.

L'établissement de zones tampons ou « buffers » identiques pour chaque station et leur intersection (« intersect ») avec les parcelles environnantes ne sont donc pas directement valables. Chaque station doit être étudiée individuellement, afin de caractériser son environnement proche, et ainsi d'appréhender les évolutions spatiales possibles des conditions météorologiques pour une surface donnée.

L'extrapolation ou l'interpolation des données météo selon les cas permettent d'approcher la représentativité, qui reste forcément relative étant donné la forte évolutivité et la variabilité des données climatiques, hydriques ou éoliennes dans l'espace et le temps. Cependant, la densité du réseau de station nous permet d'affirmer qu'il est possible d'appliquer les données ponctuelles aux entités polygonales, avec une marge d'erreur réduite.

Ce travail est en cours de réalisation et fait l'objet du sujet de thèse du doctorant Serge Olivier KOTCHI, avec lequel j'ai collaboré à plusieurs reprises durant mon stage. La solution retenue pour l'extrapolation/interpolation est celle du krigage, qui sera réalisé avec le logiciel Surfer 8.0. Ce dernier permet non seulement d'appliquer facilement une matrice à l'espace étudié, mais aussi de prendre en compte automatiquement des facteurs comme le relief ou l'hydrographie dans l'analyse.

### 2.3.4) Calcul d'indicateurs agricoles spatialisés : exemples

Une fois que chaque parcelle sera reliée aux données de telles ou telles stations, il sera alors possible d'établir des calculs à partir des données ponctuelles de base. L'objectif est de mettre au point des indices agricoles spatialisés, utiles à la gestion des parcelles et à la prévision des besoins des cultures. Le calcul d'indices permettant d'identifier les carences ou les besoins en eau est également envisagé dans les étapes futures du projet.

Ces travaux entrent également dans le mandat de Serge Olivier KOTCHI, qui utilisera probablement le logiciel MatLab pour arriver à ses fins. La mise au point de ses indices nécessitera d'établir des algorithmes complexes prenant en compte les différents facteurs spatiaux et climatiques abordés plus haut, puis de les implémenter dans le SIG du laboratoire GAAP.

A titre d'exemple, les premiers indices qui seront calculés seront l'indice de végétation normalisé (NDVI), l'indice degré jours de croissance (DJR) ou encore l'indice d'unité thermiques – maïs (UTM). Une fois établis et implémentés dans le SIG, ils permettront d'établir des cartographies évolutives, très utiles aux agronomes et aux agriculteurs.

### 3) Les SIG sur Internet : État de l'art et standardisation

#### 3.1) Présentation de l'outil Internet

##### 3.1.1) Histoire et évolutions

Internet est un réseau mondial d'ordinateurs reliés entre eux via les lignes téléphoniques, de la fibre optique, du câble GSM ou encore par des satellites. L'idée de la mise en place d'un tel réseau planétaire remonte à 1969, date à laquelle le réseau expérimental ARPANET fut créé par l'Advanced Research Projects Agency (ARPA) au sein du Département de la Défense du gouvernement fédéral des États-Unis d'Amérique. Ce réseau d'origine militaire avait alors pour but de relier quatre instituts universitaires américains et de faciliter l'échange d'informations confidentielles. Le réseau ARPANET est en fait l'ancêtre d'Internet.

Le développement massif d'Internet depuis la fin des années 1980 est lié d'une part à la démocratisation de l'informatique et d'autre part aux progrès réalisés dans le domaine des télécommunications et des technologies numériques. La naissance de protocoles et l'apparition de nouveaux moyens techniques, à l'instar des serveurs, des routeurs et des modems, ont permis à la « toile » de s'agrandir et d'être de plus en plus performante en terme d'échange de données et de rapidité de connexion.

Ancrée dans nos modes de vie tel un besoin préconçu au même titre que la télévision numérique ou le téléphone cellulaire, Internet est un produit de consommation de masse obéissant aux lois du marché. Le nombre d'opérateurs et de fournisseurs de services Internet ne cesse d'augmenter, et la forte concurrence mène à la baisse progressive des prix d'accès, allant de pair avec une augmentation croissante des débits de connexion.

Bien plus qu'un simple moyen de communication, Internet tend à devenir omniprésent dans les sociétés développées, de part sa croissance rapide et ses avantages nombreux en terme de rapidité et de partage. De plus en plus utilisé pour le commerce et la publicité, il induit des bouleversements sociétaux et éthiques importants, qu'il s'agisse d'habitudes de consommation, de téléchargements illicites ou encore de sites à contenus illégaux ou mensongers.

Outil puissant et universel, l'utilisation massive d'Internet soulève aujourd'hui des questions relatives à la justice, à la sécurité ou encore au respect des droits d'auteurs, qui influence de plus en plus les gouvernements à agir en terme de limitation des droits ou de répression des crimes informatiques.

Malgré ses aspects néfastes, ou du moins les applications nuisibles qu'il rend possible, le réseau Internet constitue un outil incontestablement pratique pour l'échange d'information et de données numériques. Les services web et l'interactivité permettent des progrès énormes, qui ont d'ores et déjà des applications dans la géomatique et les SIG, mais aussi dans ceux des systèmes d'information, de la gestion, de la robotique et de l'intelligence artificielle.

### 3.1.2) Les protocoles Internet

Les ordinateurs connectés à Internet peuvent s'échanger de l'information grâce au protocole TCP/IP et un ensemble de matériels spécifiques tels que la carte réseau, le modem, le serveur et le router.

Un protocole est une méthode standard qui permet la communication entre des processus (s'exécutant éventuellement sur différentes machines), c'est-à-dire un ensemble de règles et de procédures à respecter pour émettre et recevoir des données sur un réseau.

TCP/IP est en fait une suite de protocoles. Le sigle TCP/IP signifie «Transmission Control Protocol/Internet Protocol» et représente d'une certaine façon l'ensemble des règles de communication sur Internet. Il se base sur la notion adressage IP, c'est-à-dire le fait de fournir une adresse IP à chaque machine du réseau afin de pouvoir acheminer des paquets de données. C'est ainsi que la communication est possible entre deux ou plusieurs ordinateurs donnés au milieu d'un réseau de plusieurs millions de machines connectées.

Il existe de nombreux autres protocoles dont les plus connus sont HTTP (Hyper Text Transfert Protocol), FTP (File Transfert protocole), DHCP (Dynamic Host Configuration Protocol), ou encore SMTP (Simple Mail Transfert Protocole).

### 3.1.3) La relation client / serveur

Les différents services Internet comme World Wide Web fonctionnent selon la relation client/serveur. Cela signifie qu'un logiciel serveur approprié doit être installé sur un ordinateur hôte, pour que celui-ci puisse proposer ses services à d'autres ordinateurs distants. Pour se faire, le logiciel serveur doit être constamment actif sur l'ordinateur hôte, lequel doit être connecté à Internet. L'ordinateur client peut également disposer d'un logiciel de protection (dit pare-feu) qui empêche ou restreint les accès venant des serveurs extérieurs.

Les Serveurs sont donc des programmes qui veillent en permanence l'arrivée de requêtes concernant leurs services, depuis les ordinateurs clients. Par exemple un serveur www attend des demandes d'appel de pages web qui sont sauvegardées sur la machine offrant le service.

Les Clients sont des programmes logiciels qui réclament des données au serveur. Les navigateurs Internet explorer, Firefox ou Safari sont des logiciels clients par excellence. Si l'on clique sur un lien qui conduit à une adresse HTTP, le navigateur lance alors une requête au Serveur correspondant. Le Serveur évalue la demande et renvoie les données désirées. C'est le concept même du protocole http, qui régie la communication client / serveur et le fonctionnements des différents services web.

### 3.1.5) L'Internet pour la diffusion cartographique

La géomatique et des SIG ont toujours évolués parallèlement aux progrès de l'informatique, des technologies numériques et du déploiement d'Internet. C'est lorsqu'un SIG est établi en réseau et qu'il est relié à une connexion Internet haut débit qu'il prend tout son pouvoir. Il devient alors à la fois un outil de recherche et d'analyse, et un média rendant possible la communication et la publication de l'information géographique. L'utilisation fréquente de réseaux Intranet ou d'espaces ftp dans les entreprises ou les institutions tend à faire d'Internet un outil privilégié de gouvernance et de standardisation, de plus en plus utile aux décideurs, aux gestionnaires de toutes sortes ainsi qu'aux gouvernements locaux.

Le rôle de la carte en tant qu'outil universel dans un nombre grandissant de domaines, permet à la cartographie numérique sur Internet d'évoluer fortement. De plus en plus d'acteurs et d'institutions expriment le besoin de diffuser et/ou partager l'information, qu'il s'agisse aussi bien de thématiques reliées à l'environnement, à l'aménagement, au cadastre, au calcul de d'itinéraires, ou encore au géomarketing. C'est ainsi que les activités reliées au webmapping constituent une discipline géomatique de plus en plus prisée et évolutive, alliant les savoir-faire du géomaticien (analyses spatiales, connaissances métiers, cartographie...) et ceux de l'informaticien développeur (base de données, réseaux, systèmes, langages...)

Bien plus que la simple diffusion de cartes descriptives, les nouvelles technologies Internet permettent aujourd'hui l'édition de réels SIG en ligne. Il s'agit de reproduire les algorithmes couramment employés par les logiciels SIG à des fins d'analyse spatiale et de calcul. Pour ce faire, il est nécessaire d'opter pour un type d'architecture et un mode de fonctionnement adaptés aux besoins du projet et des utilisateurs potentiels.

D'autre part, il est aussi possible de développer des modules logiciels en ligne, à l'aide de langages objets tels que le Python, le C++, le Fortran ou encore le Java. Les opérations courantes de conversion de format ou de projection, les analyses spatiales simples, la numérisation ou encore les opérations de sémiologie graphique avancées deviennent alors possibles en ligne, et peuvent permettre d'augmenter le rôle et l'impact d'un SIG de manière significative.

Tout comme les sites Internet traditionnels, les SIG en ligne connaissent par ailleurs une relative standardisation, initiée notamment par l'Open Geospatial Consortium (OGC) et le W3C. L'utilisation grandissante de la norme XML et des langages dérivés comme le SVG, le X3D ou encore le XHTML laisse penser que la diffusion de l'information géographique doit obéir à des règles pré-établies, si l'on veut tendre vers des méthodes de stockage et de partage optimales. Ce courant de pensée ne se développe cependant que timidement, face à la profusion des solutions commerciales de diffusion (serveurs cartographiques) et à la variété des langages web utilisés.

## 3.2) Inventaire des techniques de webmapping existantes

### 3.2.1) Solutions statiques vs solutions dynamiques

Comme tout site Internet, une application cartographique en ligne peut être dynamique ou statique. Cependant, il existe plusieurs niveaux d'interactivité entre une carte complètement figée et une cartographie dynamique interactive. Il revient au géomaticien/développeur (Est-ce d'ailleurs plus légitime de le nommer cartographe Internet, ou encore webmappeur?) de juger du niveau d'interactivité nécessaire.

La carte statique dite « morte » repose le plus souvent sur l'ancestrale fonction html « usemap ». Cette technique n'est que peu recommandée pour les cartes contenant beaucoup d'objets, et est plutôt longue à mettre en œuvre. Il faut en effet déterminer à l'œil nu, ou via des générateurs d'images map, les coordonnées en pixels de chacun des éléments que l'on veut rendre interactifs, c'est-à-dire cliquables. Notons toutefois que cette technique a été actualisée dans la recommandation du langage XHTML 1.0. Il est aussi possible et recommandable d'utiliser les feuilles de styles (CSS2) pour « mapper » une image. Lesquels procureront un code plus clair et surtout plus conforme.

Dans le cas d'une cartographie dynamique et interactive, la carte et les données attributaires sont bien souvent générées par un serveur à la demande de l'utilisateur. Il est cependant possible de distinguer différents niveaux de dynamisme. Une application est qualifiée de dynamique si elle a recours uniquement à un serveur (cartographique ou non). Elle est dite semi dynamique lorsqu'une partie de l'intelligence est déportée du côté client, grâce à des langages de programmation tel que le javascript ou le DHTML (HTML dynamique).

### 3.2.2) Les serveurs cartographiques

Le recours aux serveurs cartographiques est aujourd'hui de plus en plus répandu dans les applications de cartographie numérique. Ils représentent des technologies serveur dédiées à la diffusion cartographique et viennent compléter les solutions serveurs traditionnelles dont les capacités sont souvent vites limitées en terme de cartographie et de lien aux bases de données spatiales.

Les principaux éditeurs de logiciels SIG ont d'ores et déjà développé des applications serveur dans leur gammes de produits commerciaux. C'est le cas d'ESRI avec Arc IMS, d'Apic avec Apicweb, de Claritas avec MapExtreme ou encore de Star Informatics avec la solution StarNext. Leurs principaux avantages résident dans la facilité de mise en œuvre, leur évolutivité et les méthodes de mises à jour par exemple, ainsi que dans l'automatisation de certaines tâches de programmation. Ils permettent ainsi de générer rapidement un SIG en ligne et de créer des pages HTML automatiquement, sans connaissances réelles des langages web.

Leurs inconvénients sont leurs prix élevés, s'ajoutant au coût de la licence du SIG de base, et bien sûr des coûts de rediffusion des données géographiques. Notons aussi que les serveurs cartographiques commerciaux produisent du code HTML souvent non conforme, ou tout simplement basé sur le HTML 4.0, qui tend à devenir un langage web obsolète.

Par ailleurs, de plus en plus de serveurs cartographiques libres voient le jour et sont couramment utilisés dans les projets géomatiques. Ces derniers sont conçus sur des noyaux open-source, délivrés généralement gratuitement sous les principes de la Licence Publique Générale, et fonctionnant aussi bien sur des plateformes Windows ou Linux.

C'est le cas par exemple du désormais bien connu MapServer ou encore du serveur cartographique Alov Map. Ceux-ci sont un peu moins simples d'installation et d'utilisation, mais possèdent les mêmes fonctionnalités que les solutions commerciales. Il faut cependant souvent développer quelques habilités de programmation pour paramétrer l'interface graphique qu'ils procurent.

Notons enfin que de plus en plus de serveurs cartographiques sont conçus sur la base de MapServer et du PHP, au travers d'applications commerciales ou libres. C'est le cas par exemple du serveur CartoWeb, développé par la société rhônalpine Camp2camp. Un très bel exemple de ce que peut faire CartoWeb est consultable sur le site de l'inventaire forestier national à cette url : [http://www.ifn.fr/spip/rubrique.php3?id\\_rubrique=67](http://www.ifn.fr/spip/rubrique.php3?id_rubrique=67)

### 3.2.3) Les formats vectoriels

Les deux formats vectoriels utilisés aujourd'hui sont le Flash (.swf) et le SVG (.svg). Conçus initialement pour mettre en place des animations graphiques et sonores pour sites web, ces formats ont peu à peu intégré les applications de webmapping. Ils tendent à prendre le dessus sur les langages GML (Geographic Markup Language) et SVF (Simple Vector Language), déjà anciens et de moins en moins utilisés en cartographie numérique.

Le SVG est un standard XML (eXtended Markup Language) relativement bien documenté et ouvert. Il tend à concurrencer le format Flash (.swf) qui est un langage propriétaire développé par la société Macromédia. Cette dernière détient le monopole en matière de dynamisme web, dans la mesure où la majorité des applications dynamiques sont aujourd'hui réalisées avec le logiciel FlashMX. Il est quand même important de noter que Macromédia développe présentement des outils permettant d'intégrer le SVG, rendant notamment la conversion Flash / SVG possible. Cela démontre bien le développement important du SVG, ainsi que la volonté des éditeurs de logiciels à s'aligner de plus en plus sur les solutions open-source.

D'autres formats vectoriels sont aussi utilisés pour diffuser de l'information géographique, comme par exemple le GML (Geographic Markup Language). Celui-ci est dédié au transport et au stockage des données spatiales, et également basé sur la norme XML. Bien que relativement peu utilisé aujourd'hui, il est mis à jour régulièrement par l'OGC, la version actuelle étant la 3.1.0.

Notons enfin le fort développement des langages web en trois dimensions (3D) dans les applications de webmapping, à l'instar du VRML (Virtual Reality Modelling Language) et de son successeur, le langage X3D. Leur objectif est de permettre la représentation d'univers 3D virtuels sur des sites Internet. Ces outils tendent à devenir de puissants médias pour les SIG 3D, permettant de publier des modèles numériques de terrain, des scènes urbaines ou rurales géoréférencées, ou encore des animations 3D en temps réel.

### 3.3) Les standards libres de diffusion de l'information géographique

#### 3.3.1) Le SVG, outil de représentation de contenu vectoriel en 2D

La norme ou « format » SVG (Scalable Vector Graphics), est le standard Internet en vigueur pour la diffusion de contenu vectoriel en deux dimensions, défini par le World Wide Web Consortium (W3C), depuis le 14 Janvier 2003. Pouvant être aujourd'hui considérée comme une extension du Langage GML, cette norme est en fait née du regroupement stratégique des principaux acteurs du graphisme informatique intervenants sur le web à l'instar d'Adobe, Apple, AutoDesk, Corel, HP, IBM, Macromedia, Microsoft, Netscape, Sun Microsystems, ou encore Xerox. Cela bien sûr dans le but de garantir le développement de SVG a travers des produits commerciaux (larges possibilités d'importation et d'exportation, développement de viewers et de plug-ins...).

Public et facilement ouvert sur les langages web tels que le javascript, le XHTML et le PHP, le SVG s'impose peu à peu dans la conception et la diffusion de contenu dynamique sur Internet. Son développement allant de paire avec la diffusion croissante de pages web XML valides, il devrait logiquement permettre une standardisation certaine dans la diffusion d'applications dynamiques sur le web, et par conséquent favoriser l'échange des informations avec des temps de chargements de plus en plus faibles (ces derniers étant encore assez long avec la technologie Flash, indépendamment de la taille du fichier swf et de la vitesse de connexion du poste client).

SVG est un langage de description d'objets bidimensionnels basé sur une grammaire XML. Il admet trois types d'objets graphiques: des contours graphiques vectoriels (lignes ou polylignes, droites ou courbes, rectangles, cercles, polygones ou chemins (path)), des images et du texte. L'élément le plus important est le chemin, qui permet le plus de possibilités graphiques. Il permet de dessiner des lignes ou des polygones au moyen d'une liste de coordonnées. La syntaxe est proche de celle utilisée en PostScript ou en PDF car le chemin est dessiné avec des commandes comme "moveTo" ou "lineTo", auxquelles s'ajoutent des courbes de Bézières cubiques et carrées ainsi que des portions d'ellipses.

« Scalable » veut dire adaptable, c'est-à-dire dont la résolution ne se limite pas à une valeur fixe de pixels, d'un point de vue graphique. Par rapport à Internet, adaptable signifie qu'une technologie donnée peut s'étendre à un grand nombre de fichiers, à un grand nombre d'utilisateurs, ou encore à une grande variété d'applications. SVG est adaptable dans les deux sens du mot. Les graphiques SVG s'adaptent aux différentes résolutions d'affichage pour, par exemple, permettre qu'une sortie imprimée utilise la pleine résolution de l'imprimante ou qu'une sortie écran puisse s'afficher avec la même dimension sur des écrans ayant des résolutions différentes. Ils sont adaptables parce qu'un même contenu SVG peut être un graphique autonome, ou bien être référencé ou être inclus dans d'autres graphiques SVG, ce qui permet ainsi la construction d'une illustration complexe par couches ou « calques ».

Les viewers SVG fonctionnent selon une simulation de "l'algorithme des peintres". Si deux calques se superposent, alors celui du dessus masque celui du dessous, intégralement ou non dépendamment du degré de transparence définit. Les objets jusqu'à 2,5 dimensions (c'est le cas par exemple des modèles numériques de terrain) peuvent être représentées correctement par cet algorithme et l'ordre d'affichage des calques correspond à l'ordre des objets présent dans le document SVG.

Par ailleurs, le SVG ouvre de nouveaux horizons pour la diffusion de l'information géographique et tend à fournir des solutions nouvelles aux cartographes et aux géomaticiens. Les avantages qu'il apporte, tant au niveau de son adaptabilité aux autres standards Internet que de par son interopérabilité entre les plateformes, laisse penser qu'il possède un fort potentiel et qu'il sera de plus en plus appliqué pour mettre en place des SIG en ligne.

Tout comme le XML s'impose comme la norme officielle de diffusion des métadonnées géographiques, le SVG pourrait prendre le dessus pour la diffusion des cartes dynamiques. Cette hypothétique progression nécessiterait bien sûr le développement de logiciels SVG et d'extension SIG fonctionnels afin de rendre l'emploi de ce langage accessible à tous. C'est notamment grâce à son interface graphique et à sa facilité d'utilisation que le Flash s'est avéré devenir une référence chez les webmasters, ce qui n'est pas encore le cas du SVG. Si les possibilités qu'il assure sont très riches, il reste un langage Internet plutôt abstrait et nécessitant la connaissance des bases du XML. La nécessité de rédiger du code pour mettre en place une application SVG est un facteur limitant pour sa diffusion auprès du grand public.

Il est cependant possible d'émettre l'hypothèse que le SVG sera de plus en plus soutenu par les éditeurs de SIG dans un avenir assez proche ( prenons par exemple la collaboration entre ESRI et Adobe pour faire de SVG Builder une extension fonctionnelle d'Arc GIS 9). Comme les formats Illustrator (.AI) et PDF, le SVG est l'un des rares formats graphiques au format ASCII. L'affichage de documents SVG sur le web nécessite donc l'installation du plug-in Adobe SVG sur le poste client. Cependant, il est d'ores et déjà possible de penser que les versions futures des navigateurs tels qu'Internet Explorer ou Mozilla Firefox intégreront directement cette fonctionnalité. C'est le cas de Firefox 1.1 qui sera disponible très prochainement.

SVG offre un haut niveau d'interactivité, permettant de développer des applications cartographiques à part entière, totalement interconnectées avec des outils de bases de données. SVG autorise une interaction importante entre les éléments de la carte et la page HTML, et permet par exemple d'afficher ou de masquer certains objets ou groupes, de modifier dynamiquement le style, la transparence ou la couleur, de se localiser sur un lieu donné à une échelle donnée, d'afficher des labels, de régir des événements de souris (onmouseover, onclick ...) ou encore de se connecter sur des bases de données distantes.

C'est lorsqu'il est couplé avec d'autres standards que le SVG prend tout son pouvoir et qu'il permet de développer des pages Internet interactives de qualité. C'est grâce au langage PHP qu'il est possible de relier les objets du document SVG avec des bases de données spatiales, via un système de gestion de bases de données relationnelles de type MySQL par exemple. Enfin, l'un des avantages incontestable de SVG est qu'il permet de s'affranchir d'un serveur cartographique, et surtout de répartir les opérations à réaliser du côté serveur et celles à envoyer du côté client

### 3.3.3) Le PHP, lien entre la carte et la base de données

PHP (*Hypertext Preprocessor*) est un langage objet de scripts, spécialement conçu pour le développement d'applications Internet. Il s'intègre facilement au HTML et constitue un langage s'exécutant du côté serveur. Il se distingue donc des langages s'exécutant côté client (comme le DHTML ou le Javascript par exemple). Le poste client ne reçoit que le résultat du script, sans avoir accès au code source qui est à l'origine du contenu affiché.

C'est un langage CGI (Common Gateway Interface), au même titre que le Perl, le Soap ou le C. Cependant, il ne nécessite pas d'écrire un programme constitué de nombreuses lignes de commandes afin d'afficher une page HTML. Le code PHP est inclus à l'intérieur de la page HTML afin de réaliser une action déterminée. Le code est inclus entre des balises prédéfinies qui permettent au serveur web de passer en mode PHP. Le grand avantage de PHP est qu'il est relativement simple à mettre en oeuvre par un débutant, mais offre également des fonctionnalités avancées pour un développeur expérimenté.

Trois composants sont nécessaires pour diffuser du PHP sur Internet : un analyseur PHP (CGI), un serveur Internet et un navigateur Internet. Le serveur web exécute le PHP et le script PHP est accessible avec le navigateur web. Ces opérations peuvent aussi être menées en local, ce qui est pratique pour faire des tests de développement avant la mise en ligne. De plus il supporte de nombreuses bases de données comme par exemple PostgreSQL, ODBC, Informix, SQLite, IBM DB2, dBase, Oracle, Ingres,...

Par ailleurs, le PHP s'adapte à n'importe quel système d'exploitation (Windows, Linux, Machintosh...) ainsi qu'à différent type de serveur web (Apache, Microsoft IIS...). De plus, il laisse également le choix entre un mode de programmation procédurale ou objet, ou encore un mélange des deux. Bien que les applications orientées objet ne soient pas très standard en PHP 4, beaucoup de bibliothèques (ou librairies PHP) ont été écrites uniquement avec du code orienté objet.

Le potentiel de PHP va bien au delà de la génération dynamique de pages HTML. Il peut aussi générer des images, des fichiers PDF, des animations SVG (ou Flash), ou encore des documents XML « à la volée », c'est-à-dire créés par le serveur à la demande du client, puis renvoyés instantanément au navigateur web. On retrouve donc PHP dans les applications Internet manipulant beaucoup de contenu et/ou nécessitant des outils d'interactivité, comme par exemple les SIG en ligne, les sites associatifs et collaboratifs, ou encore les sites commerciaux (solutions e-commerce).

Enfin, qu'il soit installé manuellement ou automatiquement sous Windows, Mac OSX, Linux ou autres ODBC, PHP nécessite un certain nombre de bibliothèques de fonctions pour pouvoir fonctionner à son plein potentiel. Certaines sont directement intégrées dans la dernière version de PHP, et d'autres doivent être téléchargées et ajoutées au répertoire du serveur web de la plateforme de développement.

### 3.3.4) L'ECMA Script, agent d'interactivité avec l'utilisateur

ECMAScript est un langage de programmation Internet orienté objet issu des technologies Javascript et Jscript. Ce langage a été développé par Netscape en 1996, puis repris par Microsoft depuis Internet Explorer 3.0. Il correspond en fait à un effort de standardisation du javascript, qui est l'un des langages les plus utilisés dans les pages web. Il est particulièrement adapté pour dynamiser les documents SVG.

C'est un langage de script, c'est-à-dire un langage de programmation permettant l'automatisation de certaines tâches, la manipulation ou l'amélioration du fonctionnement d'un système. Javascript utilise certains outils des langages objets, mais il n'en est pas complètement un. Il ne propose pas exactement sur la même logique de développement des objets notamment au niveau du concept d'héritage.

Le javascript est un langage pour le web qui permet de dynamiser les pages HTML en interprétant les actions de l'internaute, comme le clic, le survol d'un lien ou encore le clic droit. Le traitement du document réalisé en javascript ou en ECMA est réalisé côté client, et c'est pourquoi tous les navigateurs web possèdent un interpréteur javascript. Il est possible de désactiver ces interpréteurs via certaines commandes utilisateurs, ce qui remet d'ailleurs en cause l'interopérabilité du Javascript. Un des effets pervers de l'interprétation côté client est d'obtenir des réactions différentes selon le navigateur utilisé, notamment avec Microsoft Internet Explorer qui propose sa propre version de Javascript, le Jscript. C'est donc la variété des interpréteurs et la certaine souplesse du langage qui ont mené à la proposition de la norme ECMA 262, dont le but est de standardiser les programmes écrits en javascript.

Un langage objet est un langage de programmation qui propose un ensemble de méthodes (fonctions) et de propriétés généralistes qui peuvent s'appliquer aux différents objets définis dans le langage. Il existe des classes d'objets, qui hiérarchisent les objets qui la constituent. Chaque objet hérite des propriétés et méthodes de l'objet parent. Cela n'est pas le cas en javascript et c'est pourquoi c'est un langage « orienté » objet, mais pas à part entière comme le C, le Java ou le Python.

Les objets javascript sont en fait des entités qui décrivent les éléments contenus dans une page HTML (frame, formulaires, image ...) ainsi que les éléments qui structurent cette page, comme le type de navigateur Internet par exemple. Un objet javascript n'est pas instance d'une classe car il n'y a pas de notion de typage, comme en C par exemple. Il est créé soit lors de sa première utilisation, soit lors d'une déclaration avec la commande *var*. Un objet est en fait une structure comportant des champs pouvant être des attributs ou des fonctions. Un attribut contient une donnée, pouvant être soit un élément d'un type de base (une chaîne de caractères, une valeur numérique, ...), soit un autre objet. Une fonction est quant à elle capable de prendre en compte des paramètres et d'effectuer des opérations. La différence est que les fonctions sont propres aux objets. Les données utilisées par une fonction sont généralement les attributs de l'objet et non des variables quelconques du programme.

Un très bel exemple d'interactivité obtenu par le javascript et basé sur une cartographie simplifiée est disponible à cette adresse : [http://www.route-industriekultur.de/primaer/karte\\_e.htm](http://www.route-industriekultur.de/primaer/karte_e.htm)

## 4) Conception et développement de l'application cartographique

### 4.1) Conception de la carte interactive

#### 4.1.1) Objectifs du projet Internet

La compréhension du moteur SIG agrométéorologique, et l'important travail d'apprentissage des langages web m'ont permis de participer au projet de développement et de proposer mes idées. Les objectifs ont été clairs dès le début de ma mission : Développer un SIG en ligne performant et rendre possible la consultation des données météorologiques en temps réel.

La découverte de certains langages et du caractère infini et évolutif d'Internet est cependant venue modeler le projet, tant par rapport aux tendances de programmation actuelles, que par les normes, scripts et possibilités techniques que Richard Picard et moi-même avons découverts.

La mise au point de l'échéancier du projet a été proposée par le Directeur, que nous avons suivi l'informaticien et moi-même, en partageant nos ressources et les tâches à accomplir. La période de développement correspond à la durée de mon stage et la mise en ligne du site s'est faite peu avant mon départ du Québec.

#### 4.1.2) Définition des fonctionnalités à développer

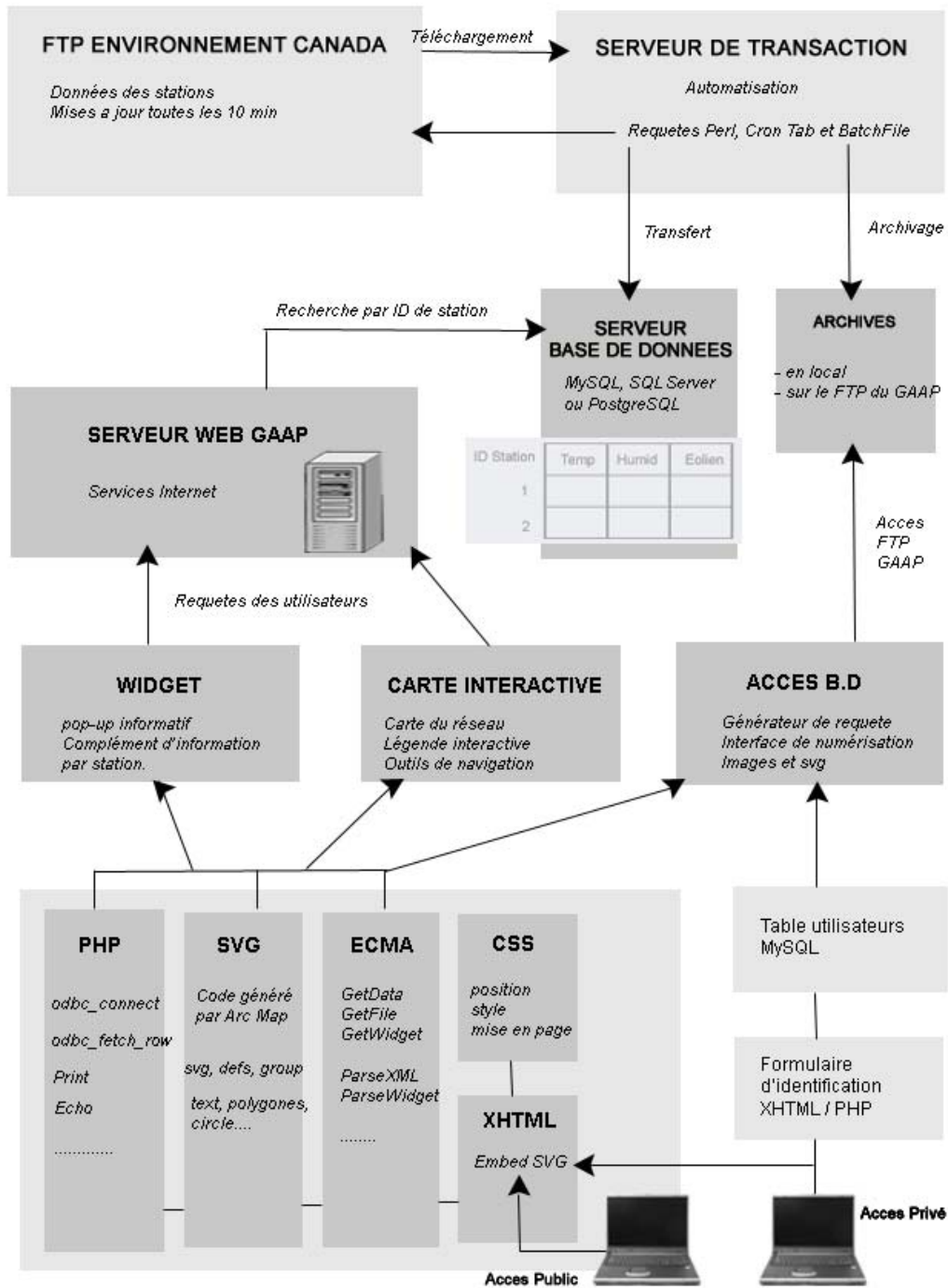
Plusieurs réunions de travail avec le Directeur, l'informaticien, le géomaticien et les autres membres de l'équipe ont permis de planifier la phase de développement Internet. C'est sur un ton consultatif et participatif que s'est donc mis en place le programme de travail, bien que le choix de l'open-source ait été pris préalablement par le Directeur, grâce notamment aux capacités de veille technologique de l'informaticien.

Il a donc été admis de développer l'application selon la norme SVG. Le premier point soulevé fut de développer les qualités de navigation fondamentales d'un SIG en ligne, à savoir les fonctionnalités de zoom et de panoramique, ainsi que le retour à la vue d'origine. Il fut aussi proposé d'intégrer un degré d'interactivité avec l'utilisateur suffisant pour rendre le SIG attrayant, notamment en proposant des temps de téléchargements les plus faibles possible.

La mise en place des services web a été planifiée, en mettant l'accent d'une part sur les paramètres indispensables à diffuser (température, précipitations, vitesse du vent et humidité relative), et d'autre part sur le choix de la méthode de développement. Les services web auraient pu être mise en place selon la norme WSDL du W3C, mais le PHP s'est avéré plus simple et rapide à mettre en œuvre. Ce choix s'est révélé comme une alternative au serveur cartographique, et ils représentent un lien infallible à la base de données. Leur développement s'est effectué progressivement et parallèlement aux avancées réalisées avec le SVG et l'ECMAScript.

J'ai enfin proposé au mi-développement de doter le site d'une architecture conforme aux spécifications du W3C. Le code SVG que nous avons généré étant conforme, et les fonctions javascript de l'application s'approchant de la recommandation de l'ECMA, il m'a paru indispensable de les intégrer dans un site web conforme XHTML et CSS. Cette normalisation du code permet une accessibilité optimale à la carte et aux services web. Sa mise en place représente par ailleurs une réelle formation au web sémantique, qui quoi qu'on en dise représente l'avenir d'Internet.

### 4.1.3) Schéma conceptuel du développement Internet



## 4.2) Construction du fond cartographique

### 4.2.1) Conversion des shapefiles en fichiers SVG avec Arc GIS 9

La première étape du développement a consisté en l'élaboration d'un fond cartographique pour l'application, basé sur la norme SVG. Plusieurs tentatives furent d'abord menées avec le logiciel Arcview 3.2, pour convertir un shapefile en fichier SVG. Les extensions SVGBuilder (version d'essai d'Adobe), toWKT (extension open source du projet Géoclient) et shp2svg (extension open source du projet Cartonnet) ont été testées. Les résultats obtenus étaient satisfaisants mais le potentiel de rendu cartographique étant relativement limité sous Arcview, il ne permettait pas de générer des cartes SVG de qualité suffisante pour le site Internet.

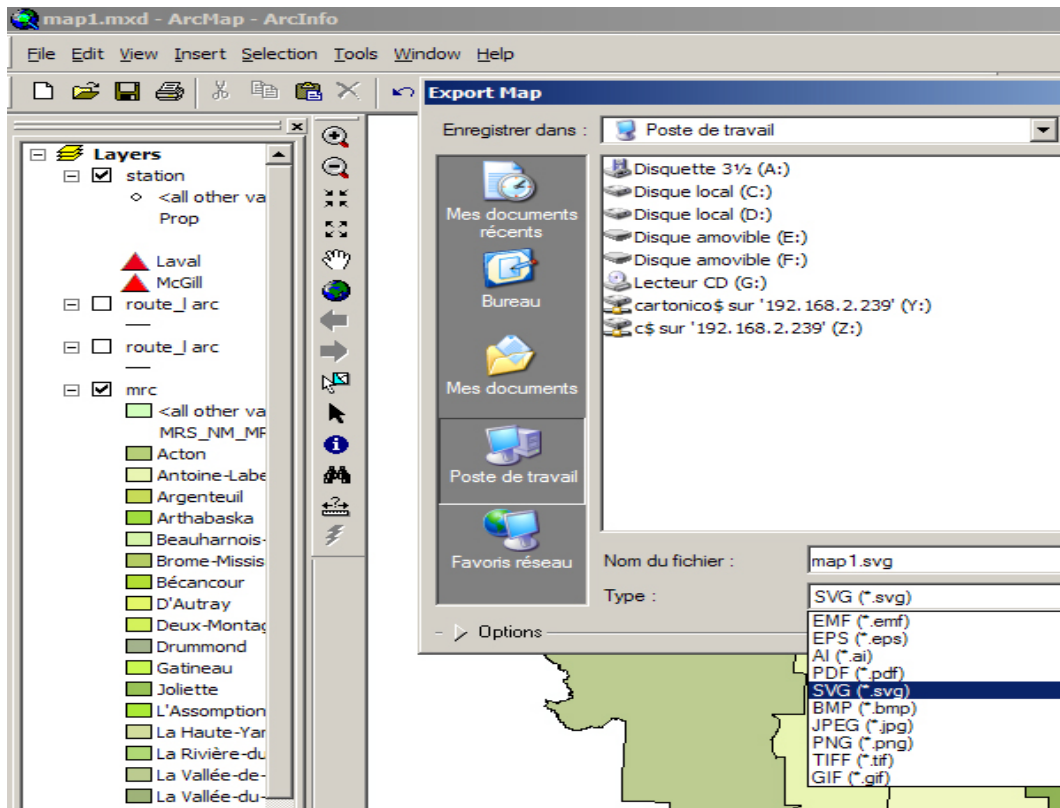
Ce n'est que lorsque la fonction native d'export SVG fut découverte sous Arc GIS 9, que la confection du fond de carte a pris une autre tournure. La dernière version d'Arc Map permet effectivement d'enregistrer directement un shapefile ou une vue composée de plusieurs shapefiles, sous la forme d'un document SVG basique. Il suffit d'emprunter la fonction « *Export Map* » de l'onglet File du menu principal et de choisir le format SVG. Cette manipulation est illustrée par la capture d'écran à la page suivante.

Les SVG générés par Arc GIS sont enregistrés en local sous la forme de pages web possédant l'extension .svg. Leurs codes sources sont établis par tuilage de l'image (correspondant à la vue d'Arc Map). Chacun des shapefiles présent dans la vue sont interprétés sous forme de complexe code SVG et empilés selon l'ordre dans lequel ils ont été implémentés dans Arc Map. Tous les SVG générés doivent appartenir à la même vue et obéir au même système de projection si l'on veut que les objets des shapefiles restent géoréférencés au sein d'un même document SVG.

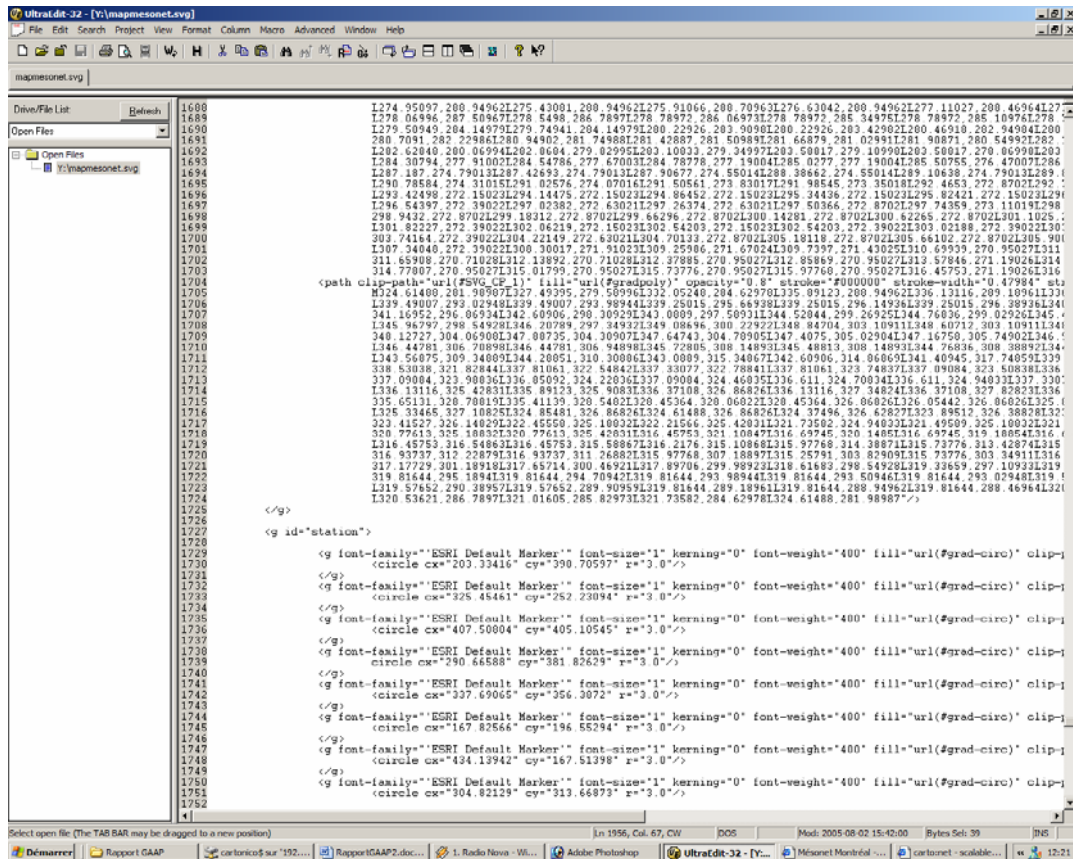
Le code produit par ESRI (dont des extraits sont présentés à la page suivante), s'il apparaît quelque peu indigeste à la lecture (suites de caractères abstrait que sont les chemins ou « path » SVG...), reste conforme et surtout exploitable depuis une page au format HTML ou PHP. Les polygones et les points, ici le réseau de stations météorologiques et les municipalités régionales de comté (MRC) correspondent en fait à des groupes SVG. Ceux-ci sont eux-mêmes imbriqués au sein d'éléments « group », « defs » et/ou « svg » qui caractérisent le code en paragraphe. Les objets géographiques deviennent alors des couches XML qu'il est possible de configurer en intervenant sur les propriétés de « group », « defs » ou « svg ».

Le problème de la conversion des labels ou « étiquettes » des objets géographique s'est par ailleurs posé. Il est en effet déconseillé d'exporter une vue comportant des objets étiquetés, au risque de compliquer encore le code et de ne pas pouvoir exploiter les groupes. Les labels ont donc été exportés au format SVG selon le même modèle que pour les polygones et les points. sous forme de couches distinctes. Les fichiers SVG obtenus ne comportent que du texte, mais sont issus de la même vue. Ils possèdent les mêmes propriétés et donc des coordonnées en pixel adaptées a celles des polygones et des points.

Cette première étape permet donc d'automatiser la création des éléments SVG de base de la carte numérique, et de créer des documents svg conforme en des temps record. Cependant, la compréhension du code est nécessaire pour parvenir à modifier l'ordre et le style des couches, ainsi que pour générer des couches toponymiques.



Export des shapefiles au format SVG sous Arc Map



Extrait du code SVG généré par Arc Map

#### 4.2.2) Mise en page des fichiers SVG : Le viewBox

Une fois les fichiers SVG générés (*mrc.svg* et *stations.svg*), il a fallu intervenir sur le code ESRI, afin d'une part d'intégrer les fonctions nécessaires à l'interactivité et d'autre part de respecter une résolution, une présentation et une charte graphique prédéfinie.

Le développement et l'intégration des fonctions PHP et ECMAScript s'étant déroulé ultérieurement, il s'agissait dans un premier temps de définir une présentation générale pour l'ensemble de l'interface utilisateur et pour la carte, laquelle correspond au mode de navigation principal des futurs services web.

Le canevas SVG est un espace infini où s'affiche le contenu SVG, défini par la DTD (Document Type Definition) de SVG. Celle-ci correspond en fait au modèle de document au format SGML décrivant la structure d'un document svg conforme. Le "viewbox" SVG est quant à lui le rectangle visible à l'écran par l'utilisateur. Il repose sur un système de coordonnées en pixel, dont l'origine se situe en haut à gauche du rectangle composant le viewBox.

Dans notre cas, la taille du viewBox généré par Arc Map a été conservée. Les propriétés de notre « fenêtre » SVG sont donc *width="532.48819"* et *height="437.24409"*, auquel on a rajouter une bordure grise de 1.5 de large, soit le code *stroke:lightgrey; stroke-width:1.5*". Tout objet dont la taille et ou la position dépasse les limites du viewBox est tronqué.

La carte est positionnée dans le document SVG selon les coordonnées *x="10"* et *y="50"*, c'est-à-dire à 10 pixels à droite et à 50 pixels en dessous de l'origine du viewBox. Elle est accompagnée d'une barre de navigation, d'une légende et d'une carte de repérage appelé « overview ».

La barre de navigation, dont le code source est présenté dans l'annexe 5, est un groupe SVG faisant appel à des fonctions ECMAScript et PHP. Elle est placée au dessus de la carte, à la manière d'une barre d'outil d'un SIG Micro.

L' « overview » est fixé en haut à droite du viewBox selon les coordonnées *x="565"* et *y="50"* et les propriétés *width="120"* et *height="112"*. Il renseigne quant au niveau de zoom et à la localisation courante sur la carte. Il n'est autre qu'un rectangle SVG dans lequel on appelle le viewBox, celui s'adaptant à la taille lui étant imposé. Cela est obtenu par le code suivant :

```
<!-- SVG OVERVIEW -->
<svg id="overview" x="565" y="50" width="120" height="112" viewBox="0
0 532.48819 437.24409">
<use xlink:href="#mrc"/>
<rect id="OverViewRect" onmousedown="startPanOM(evt)"
onmousemove="panOM (evt)" onmouseup="endPanOM(evt)"
onmouseout="endPanOM(evt)" x="2122588" y="871883"
width="8068898" height="7595814"
style="fill:limegreen;opacity:0.2"/>
</svg>
```

Enfin, la légende est présentée à droite de la carte, et simplifiée au maximum. Elle se divise en deux blocs : Les couches de bases (MRC et stations) et les couches thématiques (température, humidité relative, vitesse du vent et précipitations) Les fonctions permettant d'appeler les couches XML sont abordées plus loin

#### 4.2.3) Design de la carte : éléments de graphisme SVG

Afin de rendre la carte plus attrayante et le design du site plus esthétique, le document *mapmesonet.svg* a subi des modifications de style. La charte graphique du site est basée sur une ambiance claire où la couleur blanche domine largement, ainsi que sur trois couleurs de polices (vert, orange et noir). L'objectif fut alors d'adapter le style du SVG à celui imposé par l'ensemble du site. Il a donc fallu intervenir sur les propriétés de groupe, en changeant les couleurs selon l'encodage du document (codes couleurs hexadécimaux ou noms SVG), en harmonisant la taille des polices de caractère et en veillant à générer des styles répondant aux critères de la charte graphique pour lesquels j'avais opté au préalable.

L'apparence même des objets géographiques a été modifiée. Par exemple, les symboles des stations (triangles) générés par Arc Map ont été manuellement remplacés par des cercles, auxquels on a ajouté un dégradé radial. Cette manipulation est décrite par le code suivant :

```
<g font-family=""ESRI Default Marker"" font-size="1" kerning="0" fontweight="400"
fill="url(#grad-circ)" clip-path="url(#SVG_CP_1)" xml:space='preserve'>
  <circle cx="203.33416" cy="390.70597" r="3.0"/>
```

Les polygones ont également été dotés de dégradés radiaux afin de donner plus de consistance aux municipalités régionales de comté. La nature du dégradé est en fait définie dans une fonction native SVG définie dans l'entête du document, que l'on assigne au groupe contenant les polygones.

Le code suivant présente les dégradés utilisés pour le remplissage des points et des polygones, dont les identifiants sont appelés par la commande *fill="url(#nom du dégradé)"* dans les propriétés de groupe.

```
<radialGradient id="grad-circ">
  <stop offset="2%" stop-color="white" />
  <stop offset="90%" stop-color="forestgreen" />
</radialGradient>

<radialGradient id="gradpoly">
  <stop offset="2%" stop-color="white" />
  <stop offset="90%" stop-color="palegoldenrod" />
</radialGradient>
```

Par ailleurs, le design de la barre de navigation repose quant à lui sur des images au format PNG. Celles-ci ont été créées avec le logiciel Gimp, qui est l'analogue open-source du logiciel de traitement d'image Adobe Photoshop. On appelle ces images dans le script SVG avec une commande *xlink*, comme dans cet exemple du bouton « Zoom out » :

```
<image id="minus" x="15" y="11" width="17" height="18"
xlink:href="images/moins.png"/>
```

Ces aspects purement graphiques peuvent sans doute paraître secondaires pour un géomaticien non initié au web. Ils restent cependant primordiaux pour le développeur de SIG en ligne, aussi importants que la séméiologie graphique l'est pour le cartographe. Le trafic et le succès d'un site Internet (cartographique ou non) passe en majeure partie par un design clair et harmonieux.

#### 4.2.4) Mise en forme de l'image Landsat *quegaap.tiff*

L'objectif premier de cette étape du travail fut de répondre à des besoins de lisibilité et de clarté de la carte interactive. Disposant de l'image Landsat au laboratoire, l'équipe a trouvé intéressant de l'intégrer à l'interface cartographique du site internet. Celle-ci procure en effet une précision accrue à la carte régionale et possède des qualités graphiques certaines.

L'image satellitaire au format TIFF couvre un vaste domaine selon une forme rectangulaire, et j'ai donc entrepris de la découper. En d'autres termes, il s'agissait de traiter l'image pour qu'elle se superpose exactement à la forme du shapefile (converti en SVG pour l'application), et donc à la zone d'étude du Mésonet. Cet exercice de style a permis d'obtenir une information claire et évite surtout que l'image remplisse l'intégralité du viewbox en masquant les limites territoriales.

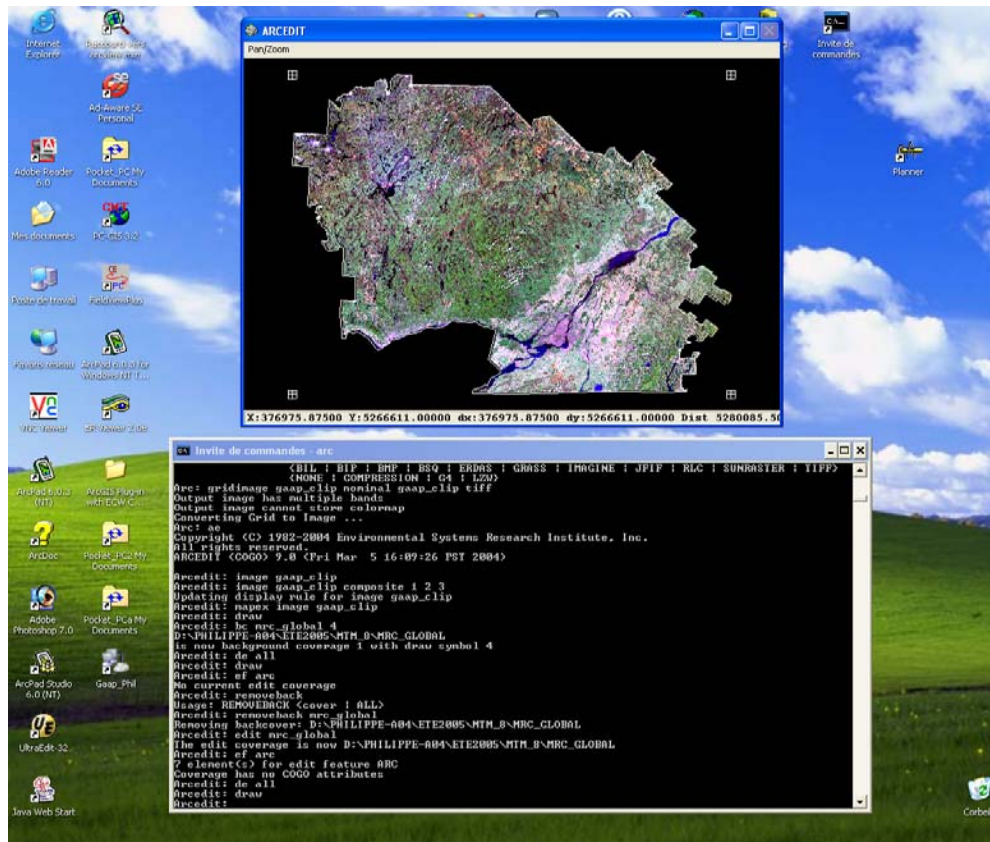
Le travail s'est déroulé sur le logiciel Arc Info, ce qui m'a permis d'une part de découvrir ce logiciel SIG surpuissant et surtout d'aborder les bases du langage commande développé par ESRI, depuis une vingtaine d'années maintenant.

L'opération a consisté en la création d'une couverture polygonale sur l'image satellite à partir du shapefile des MRC, puis au découpage du TIFF selon les nouvelles limites créées. Le meilleur moyen d'illustrer cette manipulation relativement technique est de présenter les commandes utilisées comme suit :

- Initialisation de Arc Info :  
*Command prompt :>arc*
- Conversion de *mrc.shp* en couverture Arc info :  
*shapearc <in\_shapefile> <out\_cover>*
- Construction de la topologie linéaire:  
*build mrc arc*
- Initialisation de Arc Edit :  
*ARC :> ae*
- Edition de la couverture des MRC nouvellement créée :  
*AE :> edit mrc*  
*AE :> display 9999* (fenêtre de visualisation)  
*AE :> ef arc* (édition des arcs)  
*AE :> de arc* (affichage des arcs)  
*AE :> draw* (affichage)
- Sélection des arcs utiles :  
*AE :> select many* (sélection des arcs de la zone)  
*AE :> put mrc\_global* (copie dans une nouvelle couverture)  
*AE :> edit mrc\_global*  
*AE :> ef arc*  
*AE :> select all* (sélection de tous les arcs)  
*AE :> unsplit none* (suppression des pseudo nœuds)  
*AE :> save*  
*AE :> quit*

Une fois cette sélection sauvegardée dans la nouvelle couverture, il ne restait alors plus qu'à construire la topologie polygonale via la commande *ARC* :> *build mrc\_global poly*. Le découpage du TIFF s'est quant à lui déroulé à l'aide du module GRID de Arc Info, selon la manipulation suivante :

- Implémentation de l'image :  
`image <quegaap.tiff composite 1 2 3>` (déclaration domaine RVB)
- Conversion du TIFF en grid :  
`ARC :> imagegrid <in_image> <out_stack>` (création de trois bandes)
- Initialisation du module GRID:  
`ARC:> grid`
- Découpage du grid avec la couverture polygonale (répété pour chaque bande)  
`GRID :> gridclip <in_grid> <out_stack (bande 1, 2, 3)> cover <path>`
- Création d'un stack (pile des trois bandes créés par image grid) :  
`GRID :>makestack <stack> list <grid(bande1)> <grid(bande2)>`  
`<grid(bande3)>`  
`GRID:>save`  
`GRID:>quit`
- Conversion du nouveau grid en image TIFF:  
`ARC:> gridimage <stack>nominal tuile tiff`



L'environnement Arc Info utilisé pour l'image *quegaap.tiff*

## 4.3) Constitution d'une bibliothèque de fonctions ECMAScript

### 4.3.1) La compatibilité du SVG avec ECMAScript

Les fonctions interactives intégrées du viewer SVG sont le zoom, le pan(oramique), le retour à la vue initiale et le lissage (anti-aliasing). Le plug-in d'Adobe comporte également une fonction complémentaire de recherche de texte, d'activation/désactivation du lissage, de copie et d'affichage du code-source. Il est donc possible d'affecter ces fonctions à des touches (raccourcis clavier), en tenant compte des autres événements simultanés, ce qui peut être utile dans une application cartographique.

Le SVG reconnaît trois catégories d'événement : les événements de souris, des événements de clavier, et les changements d'état (état du chargement et de l'affichage du fichier SVG). Tous les événements peuvent être combinés entre eux et les possibilités sont donc très étendues. Les événements complexes sont élaborés en combinant des scripts aux objets SVG. Avec ECMAScript, le développeur peut accéder à tous les éléments du DOM (SVG et navigateur) de l'application. Il est donc possible de relier les objets de la carte à des fichiers javascript, placés à la racine du serveur.

### 4.3.2) Les fonctions Zoom et Pan

Elles représentent les fonctions les plus élémentaires d'un SIG en ligne et permettent à l'utilisateur de changer d'échelle de visualisation et de se déplacer sur la carte. Dans le cas du site du Mésonet, ces fonctionnalités ont été développés sur la base de celles offertes par la librairie open-source Cartonet.

La fonction de zoom repose dans le fichier *zoom.js*, que l'on appelle dans le SVG avec la commande `<script type="text/javascript" xlink:href="zoom.js"/>`. La fonction est basée sur la déclaration de trois types (minus, plus et defaultview) qui correspondent aux actions de « zoom in », « zoom out » et de retour à la vue initiale. En fonction du type (donc du bouton sur lequel l'utilisateur clique), le viewbox est recalculé selon vingt degrés de zoom différents. Tout les groupes SVG sont donc agrandis ou rapetissés simultanément, ce qui permet de zoomer sur une partie de territoire ou sur une station en conservant les distances et le géo référencement des objets. Le code source de cette fonction est présenté dans l'annexe 6 à la page 62.

La fonction de panoramique repose quant à elle dans le fichier *pan.js*, appelé de la même manière que *zoom.js*. Elle aussi basée sur des types correspondant aux différentes directions possible pour le panoramique, la fonction recalcule le viewbox selon les coordonnées x et y courantes à l'écran et le cas choisi, et provoque le changement d'affichage directionnel. Elle gère aussi la localisation du viewbox courant dans la fenêtre de l'overview, complétant ainsi la fonction *overView*. Le code source de cette fonction est présenté dans l'annexe 7 à la page 63.

La compréhension des deux fonctions est appuyée par la présentation de leur code source dans les annexes 6 et 7, mais aussi par le code de la barre de navigation présenté en annexe 8. Celui-ci montre comment les fonctions Javascript sont intégrées au SVG, notamment avec les commandes *onclick*, *onmouseover*, *onmouseup* ou *onmouseout*.

#### 4.3.3) Le fichier *getdata.js*

Ce fichier se compose d'un ensemble de cinq fonctions Javascript sur lesquelles est basé l'appel dynamique des couches XML. Elles sont donc à l'origine du fonctionnement des services web. Elles constituent l'interactivité de la légende en proposant divers choix à l'utilisateur, lesquels sont interprétés puis utilisés pour extraire les informations désirées dans la base de donnée SQL server , via des requêtes PHP.

La fonction *getData* fonctionne selon l'état de cases à cocher ou « checkbox ». Si la case est cochée (c'est-à-dire cliquée), alors la fonction appelle les données ou le fichier reliés à l'information (ou la couche) désirée. Si la case est décochée, alors aucune information n'est appelée.

La fonction *getData* appelle uniquement un document XML présent à la racine du serveur. C'est le cas par exemple de l'image Landsat, qui a été convertie en SVG par Arc Map, puis enregistrée dans un document XML de base. Elle possède les mêmes propriétés SVG que la carte et s'adapte donc parfaitement à l'ensemble, par simple appel de l'utilisateur.

Le déclenchement de la requête vers le serveur GAAP est assuré par la fonction *request*, contenant les commandes *getUrl* qui sont appliquées à la carte et à la légende, et qui réfèrent aux deux fonctions suivantes.

Les fonctions *callbackMap* et *callbackLegend* s'occupent de convertir l'information demandée au format XML, et assure la mise au jour automatique de la carte et de la légende lorsqu'une couche est appelée. Enfin, les fonctions *statusChange* et *loadingBG* viennent compléter et ordonner le dynamisme généré.

#### 4.3.4) Le fichier *gaap.js*

Ce fichier se compose de quatre autres fonctions javascript nécessaires au fonctionnement des services web. Il comporte les fonctions *getFile* et *parseFile*, qui fonctionnent toutes deux grâce aux fonctions événementielles *Highlight* et *Unhighlight*.

En effet, la fonction *getFile* utilise le même modèle que la fonction *getData*, selon lequel le choix de l'utilisateur est interprété en fonction de l'état de cases à cocher ou « checkbox ». La différence est que la fonction *getFile* appelle un fichier de scripts, et nom un simple document XML comme le fait la fonction *getData*.

*GetFile* est ainsi utilisée pour tout les services nécessitant des requêtes pour accéder la base de données, et qui impliquent par conséquent des scripts PHP. La fonction a besoin de trois critères pour être effective, selon le modèle « événement, couche, url » (*evt, layer, url*), et identifie le champs visé de la base de données par la fonction *getElementbyID*.

Par exemple, pour le cas du service web permettant d'appeler les données de températures, la syntaxe suivante est utilisée :

```
<g id="checkbox7" transform="translate(5,35)">
  <a><rect x="0" y="0" height="15" width="15" onclick="getFile
    (evt,'Temperature','Temperature.php')" fill="white" stroke="url
    (#bgbutton)" stroke-width="1.5"/></a>
  <text x="20" y="12" font-size="10" fill="black">Température (°C)</text>
</g>
```

#### 4.3.5) Le fichier *getwidget.js*

Ce fichier se compose de deux fonctions Javascript qui régissent le fonctionnement du « widget », ou pop-up informatif. Celui-ci permet de visualiser les données propre a chaque station, en cliquant sur le son nom lorsqu'il a été appelé sur la carte. C'est une présentation des données complémentaire à l'interface cartographique qui fonctionne aussi selon le principe de service web.

La fonction *getWidget* est fondée sur la position du curseur de l'utilisateur. Le « widget » apparaît lorsque le nom le toponyme de station est cliqué (*onclick*) et disparaît quand le nom est cliqué de nouveau. La fonction *parseWidget* rend l'exécution l'Événement javascript possible en convertissant les données du « widget » en enfants et nœuds XML.

De plus, l'interactivité assignée aux toponymes est renforcée par la fonction *toolTip*, développée dans un autre fichier. Celle-ci permet d'obtenir un événement supplémentaire appelé communément bulle d'information. En effet, lorsque le curseur de la souris survole une station (*onmouseover*), une infobulle apparaît pour préciser à l'utilisateur qu'il doit cliquer sur le toponyme pour activer ou désactiver le « widget ».

#### 4.3.6) Les autres fonctions javascript

D'autres fonctions javascript ont été développées ou reprises d'applications existantes, pour assurer le fonctionnement global de l'application cartographique, mais l'objectif n'est pas ici de toutes les présenter. Notons toutefois l'importance des fonctions *parseXML* et *parseWidget* qui convertissent les résultats du PHP au format XML, sans quoi les données renvoyées par le serveur ne seraient pas interprétables dans le document SVG.

Signalons aussi qu'il aurait été parfaitement possible de renforcer l'interactivité entre la carte et l'internaute, via des scripts javascript supplémentaires. Il aurait été par exemple intéressant d'assigner des commandes *onmouseover* et *onmouseout* aux polygones de la carte, afin que ceux-ci s'illuminent ou change de couleurs lors du passage de la souris. Cette fonctionnalité est souvent native sur les serveurs cartographiques, mais il est aussi possible de l'obtenir par l'interaction du SVG et du javascript.

Un des manques de l'application est aussi l'affichage dynamique des coordonnées spatiales en fonction de la position du curseur. Cette fonctionnalité, également native sur les serveurs cartographiques, aurait pu voir le jour mais des problèmes techniques se sont posés. En effet, la fonction ECMA *getCoord*, découverte sur le site Pilat Informative Educative, permet cette manipulation. Basée sur les fonctions *getClient(x)* et *getClient(y)*, elle procure cependant la position du curseur selon le système de coordonnées en pixel du *viewbox*, et non en coordonnées géographique. Après plusieurs tentatives infructueuses, la décision fut prise de laisser de coté cette fonctionnalité, la méthode de conversion des coordonnées n'ayant pas été trouvée.

Enfin, d'autres choix plus animés auraient pu être faits, notamment concernant l'affichage et le fonctionnement du « widget ». Il aurait été par exemple possible de le rendre déplaçable par l'utilisateur, pour éviter de masquer systématiquement la légende de la carte. Il aurait aussi pu facilement présenter du texte animé, ou défilant. Le développement du site ne s'est pas spécialement attardé sur ce type de fonctionnalités, la priorité étant d'ériger les services web et de rendre les données consultables.

#### 4.4) Liaison à la base de données MySQL

##### 4.4.1) Connexion du SVG a la base de données MySQL

Nous abordons ici de manière concrète le lien entre l'interface cartographique et la base de données contenant les données météorologiques évolutives. Cette liaison est possible grâce au langage CGI que constitue le PHP, et représente le coté serveur de l'application cartographique. Si l'ECMAScript permet d'identifier la demande de l'utilisateur, de générer des actions et de convertir les données comme il a été vu précédemment, c'est le PHP qui va chercher les données dans la base MySQL par l'intermédiaire du serveur Apache et qui les renvoie du coté client, au sein du document SVG.

Les fichiers PHP sont stockés à la racine du serveur et appelés soit au sein des scripts javascript, soit intégrés directement dans le SVG dans certains cas. Dans les deux cas, ils doivent être écrit selon les règles de syntaxe du PHP et nécessite une connexion à la base de données en question.

La connexion à MySQL se fait simplement par le code suivant, ou *mesonet* est le nom de la base de donnée, et où *nbozon* et *webmapp* correspondent à mon nom d'utilisateur et à mon mot de passe sur le serveur dédié au projet.

```
<?php
    $conn = odbc_connect('mesonet','nbozon','webmapp');
?>
```

##### 4.4.2) Appel des valeurs sur la carte : l'exemple de la température

C'est le premier service web qui a été développé, pour être ensuite appliqué aux autres mesures disponibles selon la même méthode. Il s'agissait alors de rédiger une suite d'instructions permettant d'aller chercher les données les plus récentes du champ température de la base de données, pour une station précise.

Le PHP doit être capable d'une part de trouver et de renvoyer l'information, mais aussi d'écrire les résultats sous forme de code SVG. Pour cela, il faut déclarer l'opération d'écriture du XML selon l'encodage choisi dans l'entête du document PHP, par le code suivant :

```
<?
header("Content-type: image/svg+xml");
print('<?xml version="1.0" encoding="utf-8" standalone="no"?>');
?>
```

L'opération de recherche dans la base se fait ensuite avec le code suivant, lequel a été répété pour chacune des stations. Il prend en compte l'identifiant de la station (ici 5 dans cet exemple de la station Saint-Anicet), l'identifiant de la mesure (ici 112 pour la température) et la date de capture (heure et date de mesure).

```
$query_stAnicet= odbc_exec($conn,"SELECT Valeur FROM Tampon
WHERE NbrStation=5 AND CodeMesure =112 AND DateCapture >
(getutcddate()- 0.9) ORDER BY DateCapture DESC");
```

Afin que le PHP écrive le SVG par la commande *print xml*, il est nécessaire d'inclure directement du code SVG dans le fichier PHP. Ceci apparaît très clairement dans l'annexe 9 qui présente le code source du fichier *temperature.php*. Conservons l'exemple de la station Saint-Anicet, pour démontrer comme le SVG est imbriquable, ici avec du PHP et du Javascript. Les balises SVG *text* et *a* définissent la position et les propriétés du texte à afficher, et le rendent cliquable, la balise javascript *onmousemove* appelle l'infobulle au passage de la souris, tandis que la boucle PHP *if echo else* affiche les valeurs de température. Si la donnée n'est pas disponible dans la base, le message « no data » apparaît.

```
<g id="Temperature">
<g font-family="Arial" font-size="1" kerning="0" font-weight="400"
fill="red" clip-path="url(#SVG_CP_1)" onmouseout="toolTip(evt)"
xml:space='preserve'>

<a
xlink:href="http://mesolindev.gaap.ulaval.ca/prevision/prevision.php?id=5"
target="_blank">

<text transform="matrix(7.91744 0 0 7.91972 210.2919
382.06792)"><tspan x="0 0.66686 0.93966 1.27309 1.93995 2.51588
2.72806 3.24336 3.81928 4.09209 4.36489 "
onmousemove="toolTip(evt,'St-Anicet Lat: 45.1208 Long: -74.2895
Elevation: 47.6 metres')">

<?php
if(odbc_fetch_row($query_stAnicet)){ $temp=odbc_result($query_stAnicet
,1);echo "$temp";}else{ $temp="no data";echo "$temp";}?>

</tspan></text>
</a>
</g>
</g>
```

#### 4.4.3) Appel des données dans le widget : le fichier *prevision.php*

Le « widget » constitue un complément d'information à la carte et synthétise les données de la station cliquée. C'est un objet SVG obtenu par un événement javascript et faisant appel à des requêtes PHP. Son affichage (position et style) est standardisé pour les 39 stations, grâce au SVG qu'il faut encore une fois inclure dans le fichier PHP. Le code source de ce fichier est présentés dans l'annexe 10, à la page 65.

Les 39 cas pouvant être rencontré doivent être déclarés avec la balise PHP *case* selon la syntaxe suivante, pour que l'analyseur PHP puisse interpréter la demande de l'utilisateur :

```
case "1" :  
    $stationID="1"; break;  
  
case "2" :  
    $stationID="2"; break;  
  
case "3" :  
    $stationID="3"; break; .....
```

On appelle ensuite toutes les valeurs de la base de données que l'on veut faire apparaître sur le « widget », par des requêtes (ici, le code va chercher les données dans les champs latitude et longitude de la base de données) :

```
$query_lat= odbc_exec($conn,"SELECT Latitude FROM description WHERE  
NbrStation=$id");
```

```
$query_long= odbc_exec($conn,"SELECT Longitude FROM description WHERE  
NbrStation=$id");
```

Les mêmes requêtes sont ensuite répétées pour les différents champs de la base de données, à savoir le propriétaire de la station, l'altitude, la température, l'humidité relative, la pointe de vent, la vitesse moyenne du vent et le niveau des précipitations. L'heure et la date de mesure apparaissent également à l'aide du code : `<? print (strftime("%Y-%m-%d, %H:%M:%S")); ?>`.

Comme pour la carte, on assemble enfin des commandes SVG *text* et des requêtes PHP *if echo else*, pour chacune des informations à publier. Pour la longitude et la latitude, le code correspondant est le suivant :

```
<text id="latitude" font-family="Arial" font-size="11" fill="limegreen" x="85" y="85">  
<?php if(odbc_fetch_row($query_lat)){ $lat=odbc_result($query_lat,1);echo  
"$lat";}else{ $lat="no data";echo "$lat";} ?>  
</text>
```

```
<text id="longitude" font-family="Arial" font-size="11" fill="limegreen" x="85"  
y="105">  
<?php if(odbc_fetch_row($query_long)){ $long=odbc_result($query_long,1);echo  
"$long";}else{ $long="no data";echo "$long";} ?>  
</text>
```

## 4.5) Développement de l'interface de numérisation

### 4.5.1) De la vue régionale au plan de ferme

Le développement de l'interface de numérisation s'est déroulé une fois que l'intégralité des services Internet étaient rédigés et fonctionnels. Ce travail représente un test et n'est pas proposé sur la version officielle du site du Mésonet. J'ai tout de même tenu à la présenter car elle procure des fonctionnalités avancées au SIG en ligne, et est utile pour juger du potentiel du SVG en matière de numérisation en ligne.

Des besoins de numérisation pourraient apparaître chez les utilisateurs, dépendamment de l'avenir du réseau. Ce sera le cas si la Financière Agricole reprend le projet et l'utilise pour ses travaux relatifs à la gestion des parcelles et à la prévision des intempéries. C'est dans la perspective d'éventuellement répondre un jour à ses besoins non encore effectifs, mais aussi dans l'objectif de tester les possibilités du SVG, que cette expérience a été initiée par l'informaticien.

Le code source de cette application constitue une complexe interaction entre des documents SVG, des fonctions PHP et une multitude de fonctions ECMA. Il est issu du site Pilat informative Educative, dédié au SVG et à la cartographie sur Internet. Ce dernier propose en effet au téléchargement un outil de dessin open-source et complet, parmi d'autres scripts et routines SVG et javascript. Nous nous en sommes donc emparé puis nous l'avons adapté à nos besoins cartographiques.

Ce module expérimental de numérisation est par ailleurs a considéré comme fonctionnant en parallèle a l'accès ftp du laboratoire. En effet, si les utilisateurs ont accès aux images et aux cartes svg stockées sur le site ftp, ils pourront les télécharger puis les exploiter en local depuis le site Internet. C'est tout l'intérêt de cette application, qui permet de numériser, dessiner ou encore écrire sur n'importe quelle image au format SVG. L'utilisateur peut par exemple numériser des polygones sur le territoire régional (MRC), mais aussi travailler sur des orthophotographies, donc sur un territoire beaucoup plus local, contenant des informations plus précises, relatives a son exploitation agricole ou a son territoire d'étude.

Le changement d'échelle et la mise a disposition d'outils de dessin vectoriel peuvent permettront aux agriculteurs et aux gestionnaires agronomes de caractériser les champs cultivés, les espaces exploitables et l'environnement direct de leur territoire (pouvant contenir une station météorologique). La création future des indices agricoles spatialisés et leur implémentation dans le site Internet permettront d'accroître de manière significative le potentiel de l'interface de numérisation, car ils procureront des mesures toujours plus locales et donc précises.

Cela rend donc possible la création d'information géographique par l'utilisateur sur la base de documents SVG géoréférencés. Les modifications apportées peuvent être sauvegardées localement, puis réutilisées et modifiées ultérieurement par l'internaute. Le SIG en ligne prend donc toute son ampleur, en offrant des possibilités d'observation, de la vue régionale au plan de ferme personnalisé.

#### 4.5.2) Récupération et modification du script

Comme il a été dit, le code source a été conçu et rédigé par les auteurs du site Pilat (<http://pilat.free.fr/home.htm>), puis placé en libre téléchargement, selon un contrat « Creative Commons ». Il est donc protégé par le droit de la propriété littéraire et artistique (droit d'auteur, droits voisins, droits des producteurs de bases de données), mais est réutilisable et distribuable en ligne. Il n'est cependant pas exploitable au sein d'applications commerciales.

Les modifications apportées au script ont été minimales car la plupart des fonctions ont été conservées. Il s'agit premièrement d'adapter le mode de fonctionnement de l'interface aux besoins cartographiques éventuels des utilisateurs, puis d'adapter le design de l'interface de manière harmonieuse au reste du site, en respectant les critères de la charte graphique.

#### 4.5.3) Fonctionnement de l'application

Il ne s'agit pas ici d'expliquer les fonctions ECMA ou PHP étant à l'origine du fonctionnement de l'application, mais plutôt de montrer comment nous avons adapté cet outil de dessin au SIG en ligne.

L'application de base permet d'insérer une image en arrière-plan, d'ouvrir un document existant sur le poste client, de créer un nouveau document et de l'enregistrer au format SVG côté client. Les fonctionnalités de dessin vectoriel permettent ensuite de créer des formes SVG basiques, de dessiner à main levée et d'agir sur le style et l'apparence des nouveaux éléments créés.

Dans le cas de la carte régionale (*mrc.svg*), le document SVG est ouvert directement dans l'application, du moment qu'il est disponible localement sur la machine (le téléchargement des SVG ou des images se faisant via le serveur ftp du laboratoire). Les modifications graphiques apportées sont écrites par le PHP dans le corps du document, puis éventuellement sauvegardées par l'utilisateur.

Dans le cas où l'internaute souhaite travailler sur une orthophotographie, le fonctionnement diffère quelque peu, car le document de base n'est pas au format SVG, mais au format image (JPEG, BMP...). Deux fichiers doivent donc être disponibles localement, à savoir l'image à placer en arrière-plan et un document SVG lui étant associé. Ce dernier indique premièrement au serveur le chemin à suivre, donc la localisation précise de l'image à récupérer côté client, mais contient également les modifications apportées. Cette dernière est donc placée en arrière-plan mais n'est pas modifiée. Si l'utilisateur enregistre des modifications et souhaite en bénéficier ultérieurement, c'est le document SVG qu'il devra ouvrir.

Le fonctionnement de l'interface de numérisation nécessite donc d'une part l'accès aux données par l'utilisateur, et d'autre part la mise en place de documents SVG sur le serveur, pour toutes les images susceptibles d'être utilisées pour la numérisation. Si cela est valable dans le cadre d'un exemple, il est pour l'instant matériellement impossible de fournir tout le travail de programmation et les manipulations nécessaires dans le cadre du projet. C'est notamment pour cette raison que l'application n'est pas mise à disposition du public sur le site officiel du Mésonet, et qu'elle fera certainement l'objet de travaux supplémentaires si toutefois les besoins de numérisation s'avèrent effectifs.

#### 4.5.4) Potentiel du SVG pour numériser en ligne

La numérisation d'objets sur un document géographique est donc aujourd'hui possible grâce aux SVG et aux possibilités vectorielles qu'il procure. Ce potentiel est toutefois à modérer et ce pour plusieurs raisons.

Le SVG est un langage relativement nouveau et encore peu utilisé par les sites comportant des modules de cartographie dynamique et interactive. L'absence d'outils performants permettant d'automatiser la programmation, et mettre en place des services web, empêche cette technologie d'évoluer et de se répandre comme une référence en terme de cartographie dynamique interactive.

Bien que l'expérience menée à partir du script open-source ait été concluante, plusieurs limites à son opérabilité sont à formuler. Elle nécessite premièrement un accès FTP ou un espace de téléchargement protégé, ce qui n'est pas le cas de tous les SIG en ligne. La mise à disposition des données spatiales est même plutôt rare dans les projets de webmapping, étant donné l'importance des frais de diffusion et le caractère « protégé » des données de référence.

D'autre part, le volume de données à diffuser s'avère être un facteur limitant pour la numérisation en ligne. En effet, le travail de dessin sur les documents SVG est très lent face à une connexion Internet bas débit. La prise en compte d'images dans l'application (nativement plus lourdes qu'un document texte) est aussi limitée selon le type de connexion et l'espace disque disponible du côté client.

De plus, notons que le mode de navigation reste quelque peu « artisanal ». Bien qu'il permette à l'utilisateur de se déplacer sur la carte ou l'image, ainsi que de créer et de caractériser des polygones, la précision « géographique » reste à désirer. Cela est dû d'une part à l'aspect basique des formes que le SVG peut générer, mais résulte surtout de la non prise en compte des contraintes topologiques induites par les entités surfaciques. La création de polygones et de polygones à main levée permet certes d'éditer des nœuds, mais ceux-ci n'apparaissent pas en tant que tel et ne sont pas sauvegardés au sein du document SVG.

Par ailleurs, la navigation au « clic » reste assez lente et fastidieuse par rapport à d'autres solutions de numérisation en ligne. Cela devient vite un problème si le nombre d'objets à numériser est important et que le poste client fonctionne en bas débit et avec peu de mémoire vive.

Il est donc possible d'émettre des réserves quant au potentiel du SVG pour numériser, bien qu'il permette des fonctionnalités avancées. La mise en place d'une interface de numérisation en SVG et ECMAScript nécessite d'importants travaux de programmation et reste moins performante que les extensions web des logiciels SIG courant. Arc IMS est par exemple beaucoup plus rapide et précis pour ce faire, pouvant être couplé avec des modules logiciels en langage Java. Une bonne part de la programmation est automatisée et les fonctionnalités qu'il permet d'obtenir sont quasiment semblables aux outils de numérisation disponibles sous Arc Map. Le SVG pourrait cependant un jour permettre de s'approcher des résultats obtenus par les solutions commerciales, si des logiciels performants basés sur cette norme apparaissent.

## 4.6) Instauration d'une zone membre et d'un accès privilège

### 4.6.1) Diffusion et niveaux de permission

La mise en place d'une zone de membres sur le site Internet constitue également un test utile pour l'avenir du réseau. La zone membre n'est cependant pas encore opérationnelle sur le site officiel du Mésonet, car cela nécessite premièrement de différencier des profils utilisateurs, donc de réaliser une enquête statistique sur le trafic et le type d'utilisateurs que concernera le Mésonet, mais également de développer des scripts de sécurité pour protéger la base de données des intrusions.

Il sera donc nécessaire de différencier les utilisateurs (par exemple agriculteurs, agronomes, gestionnaires, chercheurs et étudiants) si leur nombre augmente dans les mois à venir. Cette différenciation permet d'établir des niveaux de permissions relatifs à l'accès aux données et aux services web, ce qui est indispensable dans un SIG en ligne touchant un large public, ou tout du moins concernant des utilisateurs de nature variée.

C'est encore une fois grâce au PHP qu'il est possible d'instaurer une zone membre sécurisée sur un site Internet. Il faut pour cela créer une table MySQL utilisateurs sur le serveur, laquelle enregistre les internautes ayant rempli un formulaire d'identification disponible sur le site. Une fois enregistrés, les membres du site ont accès aux pages protégées en s'identifiant par leur adresse de courriel et un mot de passe choisi lors de l'enregistrement, et enregistrés dans la table MySQL.

### 4.6.2) Tests en local et obstacles à la mise en ligne

Afin de perfectionner mes compétences en PHP et de pouvoir répondre aux différents besoins d'un site cartographique dans l'avenir, je me suis donc attelé à la réalisation d'une zone membre pour le site Internet. Pour cela, j'ai utilisé la suite EasyPHP et récupéré un script open-source sur le site <http://www.phpcs.com/>. Il a donc fallu placer ce script à la racine du serveur et l'intégrer aux différents pages que j'avais décidé de sécuriser. La création d'une table utilisateur sur PHPMyAdmin et la mise en place d'un formulaire correspondant sur le site furent les premières étapes du travail. Il a ensuite fallu placer un script de validation sur toutes les pages à protéger, ce dernier ayant pour but de vérifier la conformité et l'existence dans la base de données MySQL des paramètres utilisateurs entrés par l'internaute.

La zone membre s'est avérée fonctionnelle en local, mais il n'a pas été possible de créer la table sur le serveur du Mésonet pour des raisons de conflits entre SQLServer et MySQL. Les niveaux de permission ne sont donc pas opérationnels sur le site Internet. De plus, le script récupéré, bien que performant, présente des manques en termes de sécurité. Il ne prend par exemple pas en compte les « cookies sessions » et le cryptage des url des pages protégées, ce qui est aujourd'hui indispensable pour éviter des intrusions non désirées dans la base de données, voire des attaques sur le serveur.

Bien qu'inaboutie, la mise en place de la zone membre a tout de même permis de comprendre le processus général de sécurisation des pages web via le langage PHP. Il m'a aussi permis de comprendre l'importance de différencier les utilisateurs dans un système de diffusion de l'information géographique proposant un accès en ligne aux bases de données.

## 4.7) Intégration de l'ensemble dans une architecture XHTML conforme

### 4.7.1) Pourquoi XHTML et les CSS ?

Successeur du HTML, le XHTML est issu des spécifications du W3C. L'écriture de pages web XHTML constitue le premier pas à accomplir vers le Web de demain. A la fois compatible avec les navigateurs d'hier et avec le XML, sa rigueur en fait un puissant langage de structuration de contenu, laissant le soin de la présentation aux feuilles de styles en cascade (« Cascading Style Sheets » ou CSS).

Basés sur la grammaire XML, et donc obéissant aux principes du SGML (« Standard Generalized Markup Language), les documents XHTML sont donc directement lisibles, éditables et validables avec les outils XML standards. Ils peuvent aussi utiliser des applications qui reposent sur le DOM HTML autant que sur le DOM XML.

De plus, XHTML est par définition modulable et stylisable. Il est possible de lui ajouter des balises supplémentaires ou d'autres attributs facilement, et d'appeler un fichier CSS au sein du document, ce qui précise les propriétés graphiques, la mise en page ou encore le positionnement des différents éléments de la page XHTML au navigateur web.

La séparation du contenu et du style permet de rendre lisible et accessible toutes les informations contenues au sein d'une page à tous les types de navigateurs. Il est ainsi possible de consulter un site conforme à XHTML à l'aide d'un ordinateur, d'un smartphone, d'une télévision, d'un pocketPC, ou encore d'un terminal alpha. La conversion est faite automatiquement par le navigateur.

Face à tout ces avantages, mais aussi en raison du nombre grandissant de pages web XHTML présentes sur Internet, j'ai donc décidé d'intégrer les travaux de développement et les services Internet au sein d'une architecture XHTML/CSS. Cette décision est d'ailleurs venue accentuer la différence entre mon site et le site officiel, ce dernier étant réalisé avec l'outil Dreamweaver de la société Macromédia, qui génère du code HTML 4.0 pas toujours conforme.

Le but de ce dernier travail était avant tout de doter le réseau Mésonet d'une interface de diffusion conforme et pouvant être interprété par n'importe quel type de plateforme et de navigateur. L'ensemble du site reposant sur des documents SVG conformes ainsi que sur des fonctions ECMA s'approchant elles aussi de la validité ISO, il était logique d'intégrer le SIG en ligne dans une architecture moderne et fonctionnelle qui permette d'assurer une accessibilité optimale ainsi qu'une certaine pérennité au code.

Cette étape finale du travail de programmation m'aura aussi permis d'approfondir ma connaissance de la norme XML et de mieux appréhender les apports du web sémantique pour la diffusion de l'information géographique et la mise en place de SIG en ligne accessibles et portables.

Les recommandations XHTML 1.0 et CSS 2.0 diffusées sur le site du W3C ont été utilisées de manière approfondie pour rédiger un code XHTML et des feuilles de styles conformes. L'outil de validation en ligne également proposé par le W3C a quant à lui été le moyen de vérifier la conformité des balises utilisées et d'émettre des choix quant à la diffusion.

#### 4.7.2) Un SIG en ligne conforme ou valide ?

Suite a plusieurs séances de vérification sur le validateur du W3C et a des efforts d'adaptation du code, un seul élément s'est révélé être un obstacle a la validité XHTML du site Internet. Il s'agit de la balise *embed* utilisée pour appeler le document SVG dans le corps de la page XHTML.

Cette balise, largement utilisée dans les pages HTML traditionnelles, n'a en effet pas été actualisée dans la recommandation XHTML, et remplacée par la balise *object*, cette dernière étant considérée comme mieux adaptée au critères standards du SGML. Son utilisation est cependant fortement conseillée pour l'intégration du SVG, que ce soit par Adobe, ou par les acteurs utilisant cette technologie, qu'il s'agisse aussi bien de webmasters expérimentés ou de cartographes Internet.

En effet, la balise *object* est difficilement configurable et pose des problèmes de positionnement du SVG que l'on appelle, ce qui est très gênant en terme d'accessibilité et de standardisation. J'ai donc du renoncer a la validité du site du Mésonet ( qui aurait pu être mise en évidence dans l'interface avec l'aide du logo du W3C, lequel informe les utilisateurs et les renvoie vers une page de confirmation sur le site officiel du W3C).

Malgré cet obstacle minime, la source du Mésonet reste conforme ou « bien formée ». Cela veut dire que le code obéit aux principales règles du XHTML, mais que certains éléments non conformes empêchent la stricte validité délivrée par le W3C. L'accessibilité du site est donc logiquement affectée par ce défaut, mais dans une moindre mesure, étant donné qu'une seule balise pêche.

Le site reste donc tout de même largement accessible (en tout cas plus que le site officiel du Mésonet réalisé avec Dreamweaver...), et s'assure d'une parfaite compatibilité avec les navigateurs web d'hier et de demain. Les utilisateurs pourront donc consulter mon travail de manière optimale sur n'importe quelle plateforme informatique équipée du plug-in SVG et configurée pour une résolution écran de 1024x768 pixels.

Cette affirmation a été prouvée par les tests d'affichage menés sur différents navigateurs. Concernant la technologie Windows, les navigateurs Internet Explorer (versions SP1 et SP2) et Mozilla Firefox (version 1.0.5 et 1.0.6) affiche le site de manière identique, malgré quelques défauts pouvant apparaître selon la configuration d'Internet Explorer. Pour la technologie Machintosh, les navigateurs Safari RSS (navigateur Apple) et Mozilla Camino (version 0.8.4) interprètent aussi bien le SVG que le XHTML, et de manière parfaitement identique. Le plug-in SVG n'étant pas encore disponible pour les systèmes Unix et Linux, la compatibilité de la carte SVG et des fonctions ECMA n'ont pas pu être testé. Le code XHTML / CSS est cependant très bien reconnu par les environnements KDE et Gnome, utilisés sous la distribution Linux RedHat 4.0 lors du test.

#### 4.8) Bilan de la phase de développement Internet

Le développement du SIG en ligne s'est avéré être un succès et m'a procuré une expertise approfondie des méthodes de webmapping actuelles. Les travaux de programmation m'ont amené à comprendre le fonctionnement des services web géographiques, mais m'ont aussi aidé à appréhender le fonctionnement général d'Internet et de la relation client/serveur.

La rédaction intégrale du code-source via un éditeur de texte non automatisé a certes été longue et parfois fastidieuse, mais c'est grâce à cela que je peux à présent prétendre d'une connaissance avancée des langages web adaptés à la diffusion de l'information géographique. Cela n'aurait pas été le cas si l'application avait par exemple été mise en place avec un serveur cartographique, dont le fonctionnement est relativement simple et ne nécessite surtout que très peu de compétences en programmation Internet.

Par ailleurs, l'utilisation d'outils et de langages open-source constitue également un apport important induit par la phase de développement. De plus en plus utilisés, voire commercialisés dans certains cas, ces technologies représentent un domaine d'avenir et sont selon moi indispensables au bagage du géomaticien comme de l'informaticien. Cet aspect du projet m'a aussi fortement encouragé à m'intéresser aux produits SIG open-source, comme par exemple GRASS, GMT, JUMP, ou encore QUANTUM GIS. Ces logiciels sont certes généralement moins performants que les SIG commerciaux, mais ils constituent les outils privilégiés de la géomatique libre, que certains perçoivent comme une piste à suivre pour l'avenir.

De plus, la conformité du code-source (tout langages confondus) engendre d'une part une accessibilité optimale pour le site, et constitue également une expertise à faire valoir. La connaissance des standards Internet et des normes du W3C enrichi de manière significative mon expérience en webmapping, et en webmastering en général.

Le fait d'avoir réalisé un site Internet de manière complète, et non d'avoir réaliser des tâches sous traitées de l'informaticien comme cela aurait pu être possible si le stage s'était déroulé en entreprise, est fortement bénéfique. J'ai en effet pu aborder le projet Internet dans son ensemble, de la compréhension initiale du SIG à la mise en ligne finale. Le développement aura donc fait appel à des compétences pluridisciplinaires pour être mené à bien. J'ai ainsi autant appris en géomatique, qu'en webmapping et qu'en infographie. La mise en place des services web m'a aussi permis d'enrichir grandement mes connaissances en bases de données, et de mieux cerner leurs interactions avec le SIG d'une part, et avec Internet d'autre part.

Enfin, le travail de collaboration avec l'informaticien du laboratoire a aussi été bénéfique en terme d'administration réseau et système. Les tâches informatiques quotidiennes, les recherches sur Internet et le travail sur le réseau m'ont procuré de nouvelles compétences aussi bien hardware que software, ainsi qu'une certaine habilité à la veille technologique, qu'il s'agisse aussi bien de géomatique, de développement Internet ou encore d'open-source.

## CONCLUSION

---

Ce stage de 6 mois au laboratoire GAAP est venu répondre avec brio aux objectifs de stage que je m'étais fixé, relatifs à une spécialisation en cartographie numérique sur Internet. Cette volonté s'était peu à peu dessinée au cours de la période de cours à Saint-Étienne, au fur et à mesure de la découverte des disciplines de la géomatique et du marché de l'emploi dans ce domaine.

Les activités de webmapping sont en effet en plein développement en France, et il existe une réelle demande de la part d'institutions de recherche, de collectivités locales, d'entreprises ou encore de SSI spécialisées en géomatique. L'expertise et les compétences que j'ai acquises dans ce domaine me permettront donc logiquement de pleinement débiter ma carrière professionnelle en tant que géomaticien spécialisé en SIG web, et formé aux solutions open-source.

Le SIG en ligne a été réalisé à l'aide de langages Internet, ce qui a induit premièrement la compréhension détaillée du code rédigé, ainsi qu'une vision globale du projet Internet. Ces aspects de « coding » ne m'ont pas pour autant limité à cet unique aspect du webmapping, puisque je considère à présent être capable d'atteindre les mêmes finalités à l'aide de serveurs cartographiques open-source ou commerciaux. Ce sont les compétences pluridisciplinaires en informatique et en géomatique acquises qui constituent mon expertise des SIG en ligne, et de la diffusion de l'information géographique de manière générale.

Par ailleurs, le stage aura aussi été très formateur vis-à-vis des connaissances métiers abordées. Géographe environnementaliste de formation, la participation au projet Mésonet Montréal m'a beaucoup appris en agrométéorologie et en agriculture, ce qui est venu renforcer mes connaissances en environnement mais m'a aussi permis de développer une vision logique et synthétique du territoire agricole. La géomatique agricole est un domaine relativement nouveau et ne se limite pas aux applications de météorologie. Ma présence au laboratoire m'a fait découvrir de nombreuses applications de la géomatique à l'agriculture et aux disciplines connexes, développées au sein du projet Mésonet ou d'autres projet en cours.

Les technologies de pointe utilisées par l'équipe, notamment à propos du GPS et des campagnes d'orthophotographie, mais aussi d'un point de vue purement informatique, sont venu clairement élargir mes connaissances pluridisciplinaires en géomatique appliquée et me permettent donc à présent de mieux cerner toutes les possibilités techniques pouvant être mobilisées pour mener un projet.

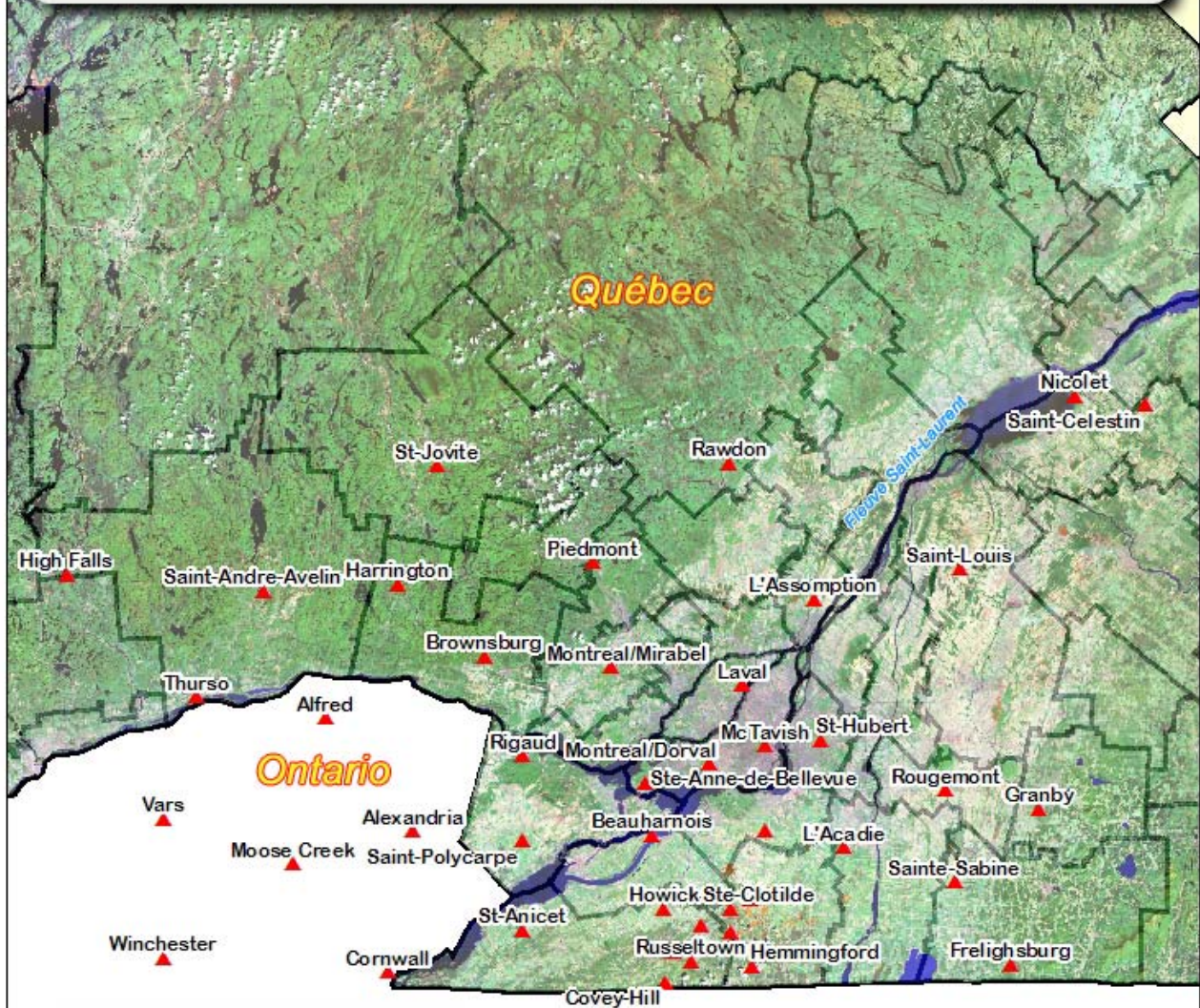
Enfin, l'intégration du laboratoire GAAP en tant que stagiaire m'aura aussi permis de d'approfondir ma connaissance du système universitaire canadien, et surtout de percevoir et de comprendre les différences entre la France et le Québec en matière de géomatique. Le monde des SIG québécois est de manière générale plus technologiquement avancé qu'en France, et cela est du principalement à l'encouragement de la recherche par l'État dans les secteurs à fort potentiel, ainsi que par la multitude d'interactions présentes entre les laboratoires de recherche et les entreprises du secteur privé. Cette expérience enrichissante a été l'occasion de travailler aux cotés d'acteurs variés dans un contexte radicalement différent, et aussi d'établir un réseau de contacts utiles pour ma future carrière en géomatique.

## ANNEXES

---

- 1) Carte du Projet Mésonet Montréal
- 2) Exemple d'un plan de ferme géoréférencé
- 3) Schéma simplifié d'une station météorologique
- 4) Code source : *mapmesonet.svg* (extraits)
- 5) Code source : La barre de navigation SVG / ECMAScript (extraits)
- 6) Code source : *zoom.js*
- 7) Code source : *pan.js* (extraits)
- 8) Code source : La légende (extraits)
- 9) Code source : *prevision.php* (extraits)
- 10) Code source : *Temperature.php* (extraits)
- 11) Code source : *index.php* (xhtml / css)

**ANNEXE 1 : Carte du projet Mésonet - Montréal**  
*Reseau de stations météorologiques automatisées, Québec*



**LEGENDE**

- ▲ Station météorologique
- Rawdon Nom de station
- Limites de Municipalités Régionales de Comté
- Image Landsat du territoire
- Québec Etats Canadiens et Américains

Vermont

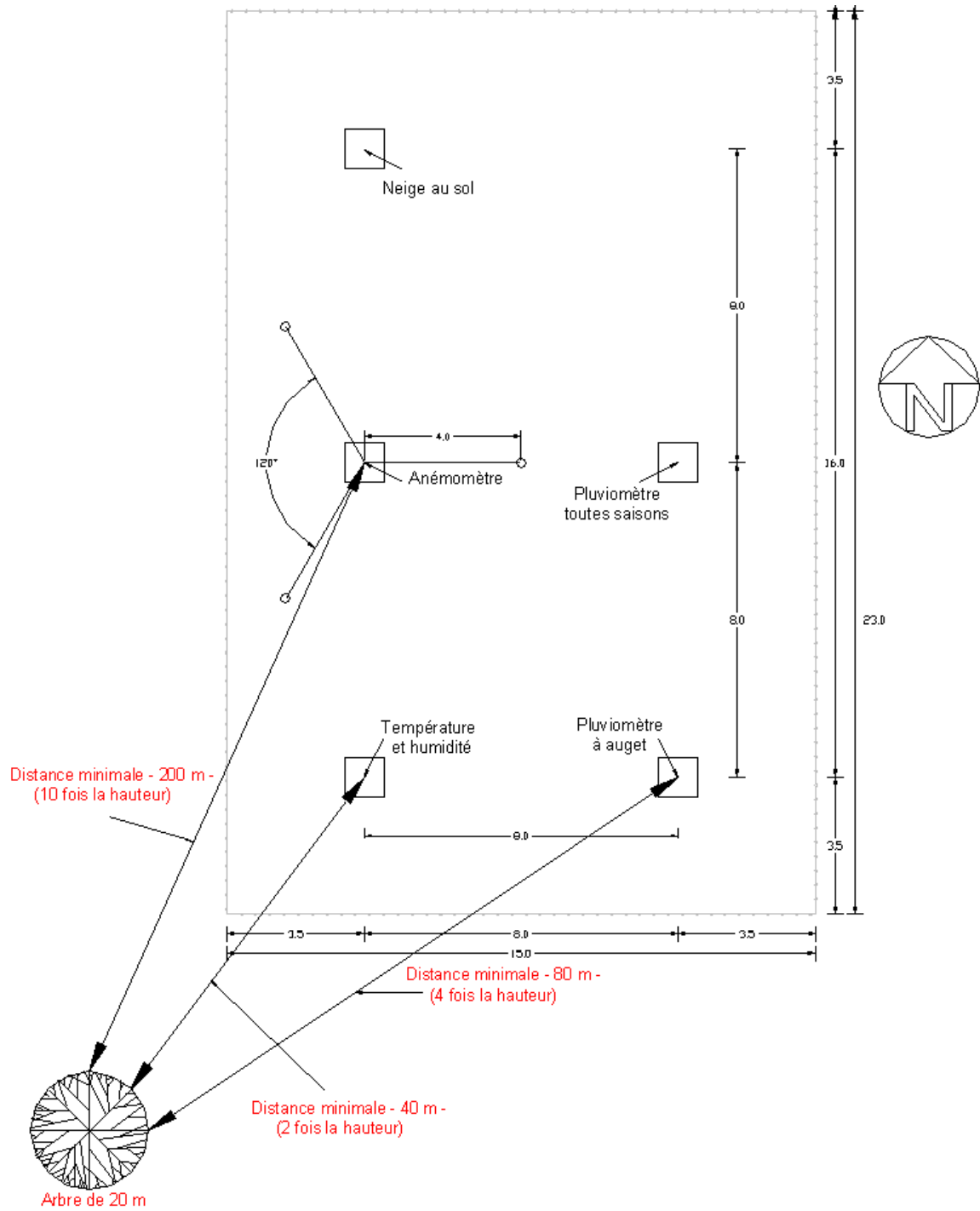
0 20000 40000 60000 80000 100000  
metres

**ANNEXE 2 : exemple d'un plan de ferme géoréférencé**



Source : BD Ortho, Ministère des ressources naturelles du Québec

### ANNEXE 3 : Schéma simplifié d'une station météorologique



Source : Projet Mésonet - HYMAP

## ANNEXE 4 : Code source du fichier *mapmesonet.svg* (extraits)

### Déclaration de la DTD

```
<?xml version="1.0" encoding="iso-8859-1" standalone="no"?>
```

```
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.0//EN"
"http://www.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd">
<svg zoomAndPan="disable">
```

### Début du code généré par ArcMap : le groupe des MRC

```
<defs>
<g id="mrc">
  <clipPath id="SVG_CP_1">
    <path
      d="M532.3876,437.24409L532.3876,0L0,0L0,437.24409L532.38
      76,437.24409z"/>
    </clipPath>

  <path clip-path="url(#SVG_CP_1)" fill="url(#gradpoly)" opacity="0.8" stroke="#000000" stroke-
  width="0.47984" stroke-miterlimit="10" stroke-linecap="square" d="M-26.63138,25.67908L-
  26.39145,25.43909L-25.91161,25.67908L-25.43176,25.91907L-24.95192,26.39905L-
  24.712,26.63904L-24.23215,26.87903L-23.51239,26.87903L-22.79262,27.35902L-
  21.83293,27.839L-21.11316,28.31898L-20.3934,29.03896L-19.43371,29.99892L-
  18.47402,30.9588L-17.27441,31.91885L-16.31472,33.11881L-15.59495,34.31877L-
  15.11511,35.51872L-14.39534,36.95867L-13.43565,38.63861L-12.47596,40.07856L-
  11.75619,40.55854L-10.55658,40.79853L-10.07674,40.55854L-9.59689,40.79853L-
  8.87713,41.03853L-8.15736,41.03853L-7.19767,41.03853L-6.71782,40.55854L-
  6.23798,40.07856L-5.27829,39.59858L-4.55852,39.35859L-3.59883,39.1186L-
  2.39922,38.8786L-1.91938,38.15863L-1.19961,37.67865L-0.95969,37.19866L-
  0.71977,36.47869L0.71977,35.99871L.....
  .....
```

### Début du code généré par Arcmap: le groupe des stations météo

```
<g id="station">

  <g font-family=""ESRI Default Marker" font-size="1" kerning="0" font-weight="400" fill="url(#grad-
  circ)" clip-path="url(#SVG_CP_1)" xml:space='preserve'>
    <circle cx="203.33416" cy="390.70597" r="3.0"/>
  </g>

  <g font-family=""ESRI Default Marker" font-size="1" kerning="0" font-weight="400" fill="url(#grad-
  circ)" clip-path="url(#SVG_CP_1)" xml:space='preserve'>
    <circle cx="325.45461" cy="252.23094" r="3.0"/>
  </g>

  <g font-family=""ESRI Default Marker" font-size="1" kerning="0" font-weight="400" fill="url(#grad-
  circ)" clip-path="url(#SVG_CP_1)" xml:space='preserve'>
    <circle cx="407.50804" cy="405.10545" r="3.0"/>
  </g>

  <g font-family=""ESRI Default Marker" font-size="1" kerning="0" font-weight="400" fill="url(#grad-
  circ)" clip-path="url(#SVG_CP_1)" xml:space='preserve'>
    <circle cx="290.66588" cy="381.82629" r="3.0"/>
  </g>.....
  .....
```

### Appel des documents javascript associés au SVG

```
<script type="text/javascript" xlink:href="globals.js"/>
<script type="text/javascript" xlink:href="getData.js"/>
<script type="text/javascript" xlink:href="overView.js"/>
<script type="text/javascript" xlink:href="others.js"/>
<script type="text/javascript" xlink:href="layerControl.js"/>
<script type="text/javascript" xlink:href="zoom.js"/>
<script type="text/javascript" xlink:href="pan.js"/>
<script type="text/javascript" xlink:href="toolTip.js"/>
<script type="text/javascript" xlink:href="gaap.js"/>
<script type="text/javascript" xlink:href="gaapwidget.js"/>
<script type="text/javascript" xlink:href="gaapGraph.js"/>
```

### Déclaration des dégradés appliqués (points et polygones SVG)

```
<radialGradient id="grad-circ">
  <stop offset="2%" stop-color="white" />
  <stop offset="90%" stop-color="forestgreen" />
</radialGradient>

<radialGradient id="gradpoly">
  <stop offset="2%" stop-color="white" />
  <stop offset="90%" stop-color="palegoldenrod" />
</radialGradient>

<radialGradient id="overgrad">
  <stop offset="2%" stop-color="red"/>
  <stop offset="25%" stop-color="orange"/>
  <stop offset="60%" stop-color="yellow"/>
</radialGradient>
```

### Début du code de la carte SVG viewBox

```
<svg id="map" x="10" y="50" width="532.48819" height="437.24409" viewBox="0 0
532.48819 437.24409" enable-background="new 0 0 532.48819 437.24409"
version="1.1" xmlns="http://www.w3.org/2000/svg">
  <use id="map_mrc" xlink:href="#mrc"/>
  <use id="map_station" xlink:href="#station"/>
</svg>
```

---

### Code de la fenêtre de localisation

```
<svg id="overview" x="565" y="50" width="120" height="112" viewBox="0 0 532.48819
437.24409">
  <use xlink:href="#mrc"/>
  <rect id="OverViewRect" onmousedown="startPanOM(evt)"
onmousemove="panOM(evt)" onmouseup="endPanOM(evt)"
onmouseout="endPanOM(evt)" x="2122588" y="871883"
width="8068898" height="7595814" style="fill:limegreen;opacity:0.2"/>
</svg>
<rect x="565" y="50" rx="0" ry="0" width="125" height="112"
style="fill:none;stroke:lightgrey;stroke-width:1.5"/>
```

## ANNEXE 5 : Code source de la barre de navigation

### Fonction de zoom

```
<g transform="translate(0,0)">
  <a id="minus" transform="translate(0,3)" onclick="zoom(evt)">
    <rect x="13" y="10" width="20" height="20" fill-opacity="0"
      style="fill:#FFFFFF;stroke:#000000;"/>
    <image id="minus" x="15" y="11" width="17" height="18"
      xlink:href="images/moins.png"/>
  </a>
  <a id="defaultView" transform="translate(30,3)" onclick="zoom(evt)">
    <rect x="13" y="10" width="20" height="20" fill-opacity="0"
      style="fill:#FFFFFF;stroke:#000000;"/>
    <image id="Defaultview" x="15" y="11" width="17" height="18"
      xlink:href="images/fullextend.gif"/>
  </a>
  <a id="plus" transform="translate(60,3)" onclick="zoom(evt)">
    <rect x="13" y="10" width="20" height="20" fill-opacity="0"
      style="fill:#FFFFFF;stroke:#000000;"/>
    <image id="plus" x="15" y="11" width="17" height="18"
      xlink:href="images/plus.png"/>
  </a>
</g>
```

### Element de décoration

```
<image id="bar" transform="translate(85,3)" x="15" y="10" width="2" height="21"
xlink:href="images/bar.png"/>
</g>
```

### Fonction de panoramique

```
<g transform="translate(0,0)">
  <g onclick="pan(evt)" transform="translate(0,0)">
    <a transform="translate(95,3)" title="Zoom -">
      <rect x="13" y="10" width="20" height="20" fill-opacity="0"
        style="fill:#FFFFFF;stroke:#000000;"/>
      <image id="Left" x="15" y="11" width="17" height="18"
        xlink:href="images/backU.gif"/>
    </a>
    <a transform="translate(125,3)">
      <rect x="13" y="10" width="20" height="20" fill-opacity="0"
        style="fill:#FFFFFF;stroke:#000000;"/>
      <image id="Top" x="15" y="11" width="17" height="18"
        xlink:href="images/pan_north.gif"/>
    </a>
    <a transform="translate(155,3)">
      <rect x="13" y="10" width="20" height="20" fill-opacity="0"
        style="fill:#FFFFFF;stroke:#000000;"/>
      <image id="Bottom" x="15" y="11" width="17" height="18"
        xlink:href="images/pan_south.gif"/>
    </a>
    <a transform="translate(185,3)">
      <rect x="13" y="10" width="20" height="20" fill-opacity="0"
        style="fill:#FFFFFF;stroke:#000000;"/>
      <image id="Right" x="15" y="11" width="17" height="18"
        xlink:href="images/fowardU.gif"/>
    </a>
  </g>
</g>
```

### Appel des toponymes des stations

```
<g transform="translate(0,0)">

<g onclick="getData(evt,'nomStation_3','nomStation_3.xml','stationlegend.xml')"
transform="translate(0,0)">

<a transform="translate(220,3)">
  <rect x="13" y="10" width="20" height="20" fill-opacity="0"
  style="fill:#FFFFFF;stroke:#000000;"/>
  <image id="lab" x="15" y="11" width="17" height="18"
  xlink:href="images/labelingstat.png"/>
</a>
</g>
```

### Appel des toponymes des MRC

```
<g onclick="getData(evt,'nomMRC','nomMRC.xml','stationlegend.xml')">

<a transform="translate(250,3)">
  <rect x="13" y="10" width="20" height="20" fill-opacity="0"
  style="fill:#FFFFFF;stroke:#000000;"/>
  <image id="labeli" x="15" y="11" width="17" height="18"
  xlink:href="images/labeling.png"/>
</a>

  <image id="deko" transform="translate(275,3)" x="15" y="9" width="350"
  height="22" xlink:href="images/deko.gif"/>
</g>
</g>
```

## ANNEXE 6 : Code source du fichier *zoom.js* (extraits)

```
function zoom(evt)
{
    var type = evt.target.parentNode.getAttribute("id");
    var viewBox = SVGMap.getAttribute("viewBox").split(" ");
    var mapCenterX = parseFloat(viewBox[0]) + viewBox[2] / 2;
    var mapCenterY = parseFloat(viewBox[1]) + viewBox[3] / 2;

    if (type == "plus")
    {
        scaleFactor = scaleFactor - 0.5;
    }

    if (type == "minus")
    {
        scaleFactor = scaleFactor + 0.5;
    }

    if (type == "defaultView")
    {
        scaleFactor = 10;

        var mapCenterX = defaultViewBoxX + ( defaultViewBoxWidth / 2 );
        var mapCenterY = defaultViewBoxY + ( defaultViewBoxHeight / 2 );
    }

    var newViewBoxWidth = defaultViewBoxWidth * scaleFactor / 10;
    var newViewBoxHeight = defaultViewBoxHeight * scaleFactor / 10;
    var newViewBoxX = mapCenterX - (newViewBoxWidth / 2 );
    var newViewBoxY = mapCenterY - (newViewBoxHeight / 2 );

    SVGMap.setAttribute("viewBox",newViewBoxX+" "+newViewBoxY+" "+newViewBoxWidth+"
    "+newViewBoxHeight);

    setOverViewRect(newViewBoxX,newViewBoxY,newViewBoxWidth,newViewBoxHeight);
}
}
```

source: <http://www.carto.net/>

## ANNEXE 7 : Code source du fichier *pan.js* (extraits)

```
function pan(evt)
{
    var panType = evt.target.getAttribute("id");
    var viewBox = SVGMap.getAttribute("viewBox").split(" ");
    var myPanFactorX = 50 * scaleFactor / 10;
    var myPanFactorY = 50 * scaleFactor / 10;

    switch(panType)
    {
        case "Top":
        {
            var viewBoxX = viewBox[0];
            var viewBoxY = viewBox[1] - myPanFactorY;
            break;
        }
        case "TopRight":
        {
            var viewBoxX = parseFloat(viewBox[0]) + myPanFactorX;
            var viewBoxY = viewBox[1] - myPanFactorY;
            break;
        }
        case "Right":
        {
            var viewBoxX = parseFloat(viewBox[0]) + myPanFactorX;
            var viewBoxY = viewBox[1];
            break;
        }
        case "BottomRight":
        {
            var viewBoxX = parseFloat(viewBox[0]) + myPanFactorX;
            var viewBoxY = parseFloat(viewBox[1]) + myPanFactorY;
            break;
        }
        case "Bottom":
        {
            var viewBoxX = viewBox[0];
            var viewBoxY = parseFloat(viewBox[1]) + myPanFactorY;
            break;
        }
        case "BottomLeft":
        {
            var viewBoxX = viewBox[0] - myPanFactorX;
            var viewBoxY = parseFloat(viewBox[1]) + myPanFactorY;
            break;
        }
        case "Left":
        {
            var viewBoxX = viewBox[0] - myPanFactorX;
            var viewBoxY = viewBox[1];
            break;.....
    }
    .....source: http://www.carto.net/
}
```

## ANNEXE 8 : Code source de la légende (extraits)

### Appel des couches de référence

```
<g transform="translate(565,180)">
  <rect x="0" y="0" rx="0" ry="0" width="125" height="312"
  style="fill:whitesmoke;stroke:lightgrey;stroke-width:1.5"/>
  <text x="11" y="20" font-size="11" fill="limegreen" font-face="trebuchet
  ms">Couches de référence</text>

  <g id="checkbox1" transform="translate(5,35)">
  <a>
    <rect x="0" y="0" height="15" width="15" onclick="mapLayerControl(evt,'mrc')
    fill="url(#checkBoxFill)" stroke="url(#bgbutton)" stroke-width="1.5"/>
  </a>
    <text x="20" y="12" font-size="10" fill="black">MRC</text>
  </g>
  <g id="checkbox2" transform="translate(5,55)">
  <a>
    <rect x="0" y="0" height="15" width="15" onclick="mapLayerControl(evt,'station')
    fill="url(#checkBoxFill)" stroke="url(#bgbutton)" stroke-width="1.5"/>
  </a>
    <text x="20" y="12" font-size="10" fill="black">Stations</text>
  </g>
</g>
```

### Début du code des couches thématiques

```
<g transform="translate(565,290)">
  <text x="11" y="0" font-size="11" fill="limegreen" font-face="trebuchet
  ms">Couches thématiques</text>
  <!-- IMAGE checkbox -->

  <g id="checkbox4" transform="translate(5,15)">
  <a>
    <rect x="0" y="0" height="15" width="15"
    onclick="getData(evt,'image','image.xml','bglegend.xml')
    fill="white" stroke="url(#bgbutton)" stroke-width="1.5"/>
  </a>
    <text x="20" y="12" font-size="10" fill="black" font-
    face="trebuchet ms">Image Landsat</text>
  </g>

  <g id="checkbox7" transform="translate(5,35)">
  <a>
    <rect x="0" y="0" height="15" width="15"
    onclick="getFile(evt,'Temperature','Temperature.php')
    fill="white"
    stroke="url(#bgbutton)" stroke-width="1.5"/>
  </a>
    <text x="20" y="12" font-size="10" fill="black">Température (°C)</text>
  </g>
```

.....  
.....

## ANNEXE 9: Code source du fichier *temperature.php* (extraits)

### Entête php

```
<?
header("Content-type: image/svg+xml");
print('<?xml version="1.0" encoding="utf-8" standalone="no"?>');
?>
```

### Connexion a MySQL

```
<?php
```

```
    $conn = odbc_connect('mesonet','nbozon','fougas');
```

### Requetes par station

```
    $query_stAnicet= odbc_exec($conn,"SELECT Valeur FROM Tampon WHERE
NbrStation=5 AND CodeMesure =112 AND DateCapture > (getutcddate()- 0.9) ORDER BY
DateCapture DESC");
```

```
    $query_assomption= odbc_exec($conn,"SELECT Valeur FROM Tampon WHERE
NbrStation=14 AND CodeMesure =112 AND DateCapture > (getutcddate()- 0.9) ORDER BY
DateCapture DESC");
```

```
    $query_freighsburg= odbc_exec($conn,"SELECT Valeur FROM Tampon WHERE
NbrStation=15 AND CodeMesure =112 AND DateCapture > (getutcddate()- 0.9) ORDER BY
DateCapture DESC");
```

```
    $query_steClotilde= odbc_exec($conn,"SELECT Valeur FROM Tampon WHERE
NbrStation=25 AND CodeMesure =112 AND DateCapture > (getutcddate()- 0.9) ORDER BY
DateCapture DESC");
```

```
    $query_acadie= odbc_exec($conn,"SELECT Valeur FROM Tampon WHERE
NbrStation=28 AND CodeMesure =112 AND DateCapture > (getutcddate()- 0.9) ORDER BY
DateCapture DESC");
```

```
    $query_stJovite= odbc_exec($conn,"SELECT Valeur FROM Tampon WHERE
NbrStation=30 AND CodeMesure =112 AND DateCapture > (getutcddate()- 0.9) ORDER BY
DateCapture DESC");
```

```
    $query_nicolet= odbc_exec($conn,"SELECT Valeur FROM Tampon WHERE
NbrStation=35 AND CodeMesure =112 AND DateCapture > (getutcddate()- 0.9) ORDER BY
DateCapture DESC");
```

---

### Appel des valeurs dans des balises SVG text

```
<g id="Temperature">
    <g font-family="Arial" font-size="1" kerning="0" font-weight="400" fill="red" clip-
path="url(#SVG_CP_1)" onmouseout="toolTip(evt)" xml:space='preserve'>
        <a
xlink:href="http://mesolindev.gaap.ulaval.ca/prevision/prevision.php?id=886" target="_blank">
            <text transform="matrix(7.91744 0 0 7.91972 285.14766
366.70847)"><tspan x="0 0.72748 1.27309 2.00058 2.24307 2.72806 "
onmousemove="toolTip(evt,'Howick')"><?php
if(odbc_fetch_row($query_howick)){ $temp=odbc_result($query_howick,1);echo
"$temp";}else{ $temp="no data";echo "$temp";}?></tspan></text>
            </a>
        </g>
    </g>
```

---

## ANNEXE 10: Code source du fichier *previsions.php* (extraits)

### Entête php

```
<?php
    header("Content-type: image/svg+xml");
    print ('<?xml version="1.0" encoding="iso-8859-1" standalone="no"?>');
?>
```

### Connexion a MySQL

```
<?php

    $conn = odbc_connect('mesonet','');
```

### Déclaration des 39 cas correspondent au 39 stations

```
if (empty($_GET["id"])) $id=NULL;
else $id = $_GET["id"];
switch ($id) {
    case "1" :
        $stationID="5"; break;
    case "2" :
        $stationID="14"; break;
    case "3" :
        $stationID="15"; break;
    case "4" :
        $stationID="25"; break;
    case "5" :
        $stationID="28"; break;
    case "6" :
        $stationID="30"; break;
    case "7" :
        $stationID="35"; break;
    case "8" :
        $stationID="49"; break;
    case "9" :
        $stationID="56"; break;
```

---

### Requetes par station

```
$query_nbr= odbc_exec($conn,"SELECT NbrStation FROM description WHERE
NbrStation=$id");
    $query_id= odbc_exec($conn,"SELECT Identifier FROM description WHERE
NbrStation=$id");
    $query_lat= odbc_exec($conn,"SELECT Latitude FROM description WHERE
NbrStation=$id");
```

---

### Appel des valeurs dans des balises SVG text

```
<text id="humidite" font-family="Arial" font-size="8" fill="#FF6600" x="85" y="205"><?php
if(odbc_fetch_row($query_hum)){ $hum=odbc_result($query_hum,1);echo
substr("$hum",0,2);}else{ $hum="no data";echo "$hum";}?>
%</text>
```

## ANNEXE 10: Code source du fichier *index.html* (extraits)

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//FR"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr" lang="fr">
<head>
<meta http-equiv="Content-Type" content="text/html; charset="ISO-8859-1" />
<title>Mésonet Montréal</title>
<link rel="stylesheet" type="text/css" href="style.css" media="screen" />
<link rel="stylesheet" type="text/css" href="banniere.css" media="screen" />
<link rel="stylesheet" type="text/css" href="contient.css" media="screen" />
<link rel="stylesheet" type="text/css" href="form.css" media="screen" />
</head>
<body>
<div id="total">
<h1>

</h1>
<div id="menu">
<h3>Projet Mésonet</h3>
<ul><li><a href="def.html" title="definition">Definition</a></li><li><a href="histo.html"
title="histoire">Historique</a></li><li><a href="partners.html" title="">Partenaires</a></li><li>
<a href="install.html" title="">Installations</a></li></ul>
<h3>Les Outils</h3>
<ul><li><a href="index.php" title="">Carte météorologique</a></li><li>Générateur de
requetes</li><li>Interface de numérisation</li><li><a href="metadata.html"
title="">Métadonnées</a></li><li>Acces FTP</li>
</ul>
<h3>Espace Membres</h3>
<form method="post" action="login.php">
<label>Courriel</label>
<input name="username" type="text" id="username" />
<label>Mot de passe</label></b><br>
<input name="pwd" type="password" id="pwd" />
<td><input type="submit" name="Submit" value="Valider" />
</form>
<ul><li><a href="registration.php">Ouvrir un compte</a></li></ul>
<h3>Liens relatifs</h3>
<ul><li><a href="SVGView.exe">Telecharger SVG Viewer</a></li>
<li><a href="http://meteo.ec.gc.ca/">Environnement Canada</a></li>
</ul>
</div>
<div id="contenu">
<h2>Services web agro-météorologiques</h2>
<embed src="mapmesonet.svg" width="750" height="600"></embed>
</div>
<div id="pied">
<p>
<a href="http://www.ulaval.ca/"></a>
<a href="http://www.mcgill.ca/" title=""></a>
<a href="http://www.ec.gc.ca/" title="site_envcan"></a>
</p>
</div>
</body>
</html>
```

## BIBLIOGRAPHIE

- Ouvrages généraux et/ou de référence :

SONNET J. - « [OpenGIS® Web Map Context Implementation Specification \(WMC\)](#)», OGC™ Implementation Specification, Document PDF, Mars 2005, 28 pages.

DE LA BEAUJADIÈRE J. - « [OpenGIS® Web Map Service \(WMS\) Implementation Specification \(WMS\)](#) », OGC™ Implementation Specification, Document PDF, Février 2004, 75 pages.

HARDER C., DANGERFIELD J. – « *Serving Maps on the Internet: Geographic Information on the World Wide Web* » - ESRI press, Juillet 1998, 130 pages.

ANTONI JP., KLEIN O., MOISY S. - « *Cartographie interactive et multimédia : vers une aide à la réflexion géographique* », Cybergéo, No. 288, Octobre 2004, 16 pages.

HELD G., SCHNABEL O, NEUMANN A. - « *Advanced webmapping with SVG* », [www.svgopen.org/2003/courses/AdvancedWebmappingWithSVG/Documentation.pdf](http://www.svgopen.org/2003/courses/AdvancedWebmappingWithSVG/Documentation.pdf), svgopen.org, Document pdf, Juillet 2003, 29 pages.

- Sites Internet officiels et environnementaux liés au projet Mésonet :

Site du Laboratoire GAAP  
<http://www.gaap.ulaval.ca/>

Site du projet Géotraceagri  
<http://www.geotraceagri.net/>

Site du regroupement HYMAP  
<http://www.hymap.ca/>

Site du Centre de Recherche en Géomatique  
<http://www.crg.ulaval.ca/default.asp>

Site du Ministère des ressources Naturelles du Québec  
<http://www.mrn.gouv.qc.ca/>

Site du Ministère de l'environnement du Canada  
<http://www.ec.gc.ca/>

Site Météo-Média, Montréal  
<http://www.meteomedia.com/Meteo/Villes/Can/Pages/CAQC0363.htm>

- Sites Internet relatifs aux autres réseaux Mésonet :

Site du Mésonet Texas  
<http://mesonet.tamu.edu/>

Site du Mésonet Oklahoma  
<http://www.mesonet.ou.edu/public/current.html>

Site du Mésonet Arizona

- <http://ag.arizona.edu/azmet/>  
Sites Internet dédiés au webmapping :

Site Pilat Informative Educative  
<http://pilat.free.fr/>

Site du projet Cartonnet  
<http://www.carto.net/>

Site du projet Géoclient  
<http://www.mycgiserver.com/~amri/geoclient.cocoon.xml>

Site généraliste sur le webmapping  
<http://sig.cwriter.org/index.php/Webmapping>

Site dédié au SVG  
<http://www.svgopen.org/2005/>

Site portail sur les SIG open-source  
<http://freegis.org/>

Site dédié au webmapping (espagnol)  
<http://www.mappinginteractivo.com/>

Site utilisé pour l'exemple d'application Cartoweb  
[http://www.ifn.fr/spip/rubrique.php3?id\\_rubrique=67](http://www.ifn.fr/spip/rubrique.php3?id_rubrique=67)

- Sites Internet liés au développement Internet :

Site d'Adobe sur le SVG  
<http://www.adobe.com/svg/main.html>

Site de ressources PHP  
<http://www.phpcs.com/>

Site de ressource Javascript  
<http://www.javascriptfr.com>

Site sur le XHTML et le web sémantique  
<http://www.alsacreation.com/>

Recommandation SVG 1.0 du W3C  
<http://www.w3.org/TR/SVG/>

Recommandation XHTML 1.0 du W3C  
<http://www.la-grange.net/w3c/xhtml1/>

Site de l'outil de validation XHTML du W3C  
<http://validator.w3.org/>

Recommandation CSS 2 du W3C  
<http://www.w3.org/TR/REC-CSS2/>

Site de la norme ECMA  
<http://www.ecma-international.org/publications/standards/Ecma-262.htm>