

**Département de géomatique appliquée**  
**Faculté des lettres et des sciences humaines**  
**Université de Sherbrooke**

**Proposition d'une boîte à outils générique pour la facilitation du développement d'applications cartographiques multiplateformes**

**Par**

**JEAN-CHRISTOPHE LEFEBVRE**

**Travail présenté à**

**Mickael Germain**

**Comme exigence finale dans le cadre de la**  
**Maîtrise en sciences géographiques de type cours,**  
**Profil géodéveloppement durable**

**Longueuil**

**OCTOBRE 2016**

## **Composition du jury**

**Proposition d'une boîte à outils générique pour la facilitation du développement d'applications cartographiques multiplateformes**

**Cet essai a été évalué par un jury composé des personnes suivantes :**

**Mickael Germain directeur de recherche**

**(Département de géomatique appliquée)**

**Yves Voirin membre de jury**

**(Département de géomatique appliqué)**

## **Remerciements**

La réalisation de ce travail a été pour moi une tâche particulièrement ardue. En effet, puisque j'occupais un emploi à temps plein de jour, j'ai dû effectuer mes recherches et rédiger une très grande partie de cet essai de soir après le travail. De ce fait, je voudrais d'abord remercier ma conjointe qui a su me supporter tout au long de cette étape difficile. Aussi, j'aimerais remercier toute ma famille, notamment mes deux parents, qui ont toujours été là pour moi depuis le tout début de mes études.

J'aimerais également remercier Mickaël Germain, mon directeur de recherche, qui a orienté mes démarches judicieusement et m'a conseillé tout au long de mon processus de rédaction. De plus, j'aimerais remercier Jérôme Théau, professeur au département de géomatique de l'Université de Sherbrooke, pour m'avoir aidé lors du cours de méthodologie à préciser mon sujet de recherche, mes objectifs et ma technique d'analyse.

# Table des matières

1. Introduction.....	1
1.1. Mise en contexte .....	2
1.2 Problématique.....	3
1.3 Objectifs .....	4
2. Cadre théorique .....	5
2.1 Étude d’ACM existantes .....	6
2.1.1 Exemple #1 : application cartographique de la fédération québécoise des clubs quads (FQCQ) 6	
2.1.2 Exemple #2 : application cartographique de Vélo-Québec .....	8
2.1.3 Résumé des observations.....	9
2.2 Les catégories essentielles de logiciels .....	10
2.2.1 Les API cartographiques .....	11
2.2.2 Les cadres de travail adaptatifs.....	12
2.2.3 Les serveurs cartographiques.....	15
2.3 Comment convertir une application web en application mobile?.....	16
2.4 Discrimination et classement des logiciels.....	19
2.5 Bilan de l’état des connaissances et de la revue de littérature.....	20
3. Matériel et méthode .....	21
3.1 Site d’étude.....	21
3.2 Données.....	21
3.3 Méthodologie.....	22
3.3.1 Choix de la méthode d’analyse multicritère .....	23
3.3.2 Acquisition des données .....	30
3.3.3 Formatage des données .....	33
3.3.4 Pondération des critères.....	33
3.3.5 Réalisation de l’analyse multicritère .....	34
4. Résultats.....	36
4.1 Catégorie API cartographique .....	37
4.2 Catégorie cadre de travail adaptatif .....	38
4.3 Catégorie serveur cartographique .....	39
4.4 Catégorie convertisseurs .....	40
4.5 Bilan des résultats et obtention de la boîte à outils .....	41

4.6 Analyse des résultats.....	41
4.6.1 Démarche de création de l'application test.....	42
5. Interprétation et discussion des résultats.....	47
5.1 Points forts et points faibles de la boîte à outils .....	48
5.2 Perspectives de la boîte à outils .....	49
6. Conclusion .....	50
7. Limites du projet et recommandations.....	51
8. Références.....	52

## Liste des abréviations

<b>ACM</b>	Application Cartographique Multiplateformes
<b>API</b>	Application Programming Interface
<b>CAT</b>	Catalogue Service
<b>CSS</b>	Cascading Style Sheet
<b>ÉLECTRE</b>	Élimination Et Choix Traduisant la Réalité
<b>FEQ</b>	Fédération Équestre Québécoise
<b>FQCQ</b>	Fédération Québécoise des Clubs Quads
<b>GeoRSS</b>	Geographic Rich Site Summary
<b>GML</b>	Geographic Markup Language
<b>GPS</b>	Global Positioning System
<b>HTML</b>	Hypertext Markup Language
<b>JPEG</b>	Joint Photographic Experts Group
<b>KO</b>	Kiloctet
<b>KML</b>	Keyhole Markup Language
<b>MB</b>	Mégabytes
<b>MTQ</b>	Ministère des Transports du Québec
<b>MVC</b>	Model View Controller
<b>OGC</b>	Open Geospatial Consortium
<b>PHP</b>	Hypertext Preprocessor
<b>PNG</b>	Portable Network Graphics
<b>PROMETHEE</b>	Preference Ranking Organisation Method For Enrichment Evaluations
<b>SF</b>	Simple Feature

<b>SIG</b>	Systeme d'Information Géographique
<b>SLD</b>	Style Layer Descriptor
<b>SVG</b>	Scalable Vector Graphics
<b>ULB</b>	Université Libre de Bruxelles
<b>WCS</b>	World Catalog Service
<b>WFS</b>	Web Feature Service
<b>WMS</b>	Web Map Service
<b>XML</b>	L'Extensible Markup Language

## Liste des tableaux

<b>Tableau 1</b> : Descriptions et remarques pour chaque serveur cartographique	p. 16
<b>Tableau 2</b> : Descriptions et remarques pour chaque convertisseur	p. 17
<b>Tableau 3</b> : Indicateurs pour chaque catégorie essentielle de logiciels	p. 30
<b>Tableau 4.1</b> : Données brutes pour les API cartographiques	p. 31
<b>Tableau 4.2</b> : Données brutes pour les cadres de travail adaptatif	p. 31
<b>Tableau 4.3</b> : Données brutes pour serveurs cartographiques	p. 32
<b>Tableau 4.4</b> : Données brutes pour les convertisseurs	p. 32
<b>Tableau 5</b> : Pondération des critères pour chacune des catégories	p. 34



## Liste des figures

<b>Figure 1</b> : Application cartographique de la FQCQ vue sur un ordinateur	p. 7
<b>Figure 2</b> : Capture d'écran de l'ACM de Vélo-Québec (vue sur un ordinateur)	p. 8
<b>Figure 3</b> : Capture d'écran de l'ACM de Vélo-Québec (vu sur un téléphone intelligent)	p. 9
<b>Figure 4</b> : Exemple de requête média avec le langage CSS3	p. 13
<b>Figure 5</b> : Définition des classes pour le cadre de travail Bootstrap	p. 13
<b>Figure 6</b> : Principales étapes de la méthodologie	p. 22
<b>Figure 7</b> : Organigramme de construction de la boîte à outils	p. 23
<b>Figure 8</b> : Capture d'écran du logiciel Visual Prométhée montrant la grille d'analyse que nous avons créée pour effectuer l'analyse multicritère	p. 35
<b>Figure 9</b> : Résultat de l'analyse PROMETHEE pour la catégorie API cartographique	p. 37
<b>Figure 10</b> : Résultat de l'analyse PROMETHEE pour la catégorie cadre de travail adaptatif	p. 38
<b>Figure 11</b> : Résultat de l'analyse PROMETHEE pour la catégorie serveur cartographique	p. 39
<b>Figure 12</b> : Résultat de l'analyse PROMETHEE pour la catégorie des convertisseurs	p. 41
<b>Figure 13</b> : Présentation des différentes sections de l'ACM test que nous avons développé	p. 42
<b>Figure 14</b> : Code JavaScript d'initialisation de la carte web	p. 43
<b>Figure 15</b> : Repositionnement des éléments de l'application cartographique test sur un appareil mobile	p. 45
<b>Figure 16</b> : Affichage d'une couche de type WMS sur l'ACM test	p. 46

## **1. Introduction**

Sous la forme d'applications cartographiques multiplateformes (ACM), d'outils d'analyse spatiale ou tout simplement de cartes web interactives, la cartographie en ligne fait désormais partie du quotidien du public (RNCAN, 2013). Des sites internet d'envergure, tels que Google ou Facebook, utilisent abondamment les applications cartographiques. Parallèlement, plusieurs applications mobiles possèdent elles aussi, de plus en plus, des fonctions de localisation spatiale ou de géopositionnement. Récemment, un sondage de la firme ComScore a permis d'évaluer le nombre d'utilisateurs de l'application mobile de Google Map à environ 60 millions pour l'année 2015 ce qui en fait l'une des applications mobiles les plus utilisées (Fulgoni, 2015). Même si Google Map n'est qu'une seule des nombreuses applications de la cartographie en ligne, la popularité de ces applications suit une tendance à la hausse (OCLC, 2010).

L'amélioration rapide des technologies du web et des télécommunications depuis la dernière décennie a permis d'ouvrir un tout nouvel horizon à la géomatique : la géomatique sur le web. Alors qu'un simple navigateur web suffit, il n'est désormais plus nécessaire d'utiliser des logiciels de bureau comme ArcGIS ou QGIS afin de visualiser des données géographiques (Neumann, 2008). Cet avantage marqué de la géomatique sur le web a rejoint un grand nombre d'organismes et d'entrepreneurs qui ont décidé d'employer cette nouvelle convergence des technologies afin d'améliorer leur offre de services (Schiller, Voisard, 2004). Par exemple, la Ville de Montréal a récemment créé une carte web interactive dans le but de mieux informer les citoyens sur l'état du déneigement de ses rues. Un autre exemple est le système d'information géographique (SIG) de Vélo Québec qui permet, entre autres, de visualiser les aménagements cyclables sur l'ensemble du territoire québécois (Vélo-Québec, 2015).

Alors que pour certains l'ordinateur reste encore le meilleur outil pour naviguer sur internet, le marché des supports mobiles comme les tablettes et les téléphones intelligents est en pleine croissance. Entre 2005 et 2010, le nombre d'utilisateurs de téléphones intelligents a augmenté drastiquement de 1050 % et est toujours sur une pente ascendante (OCLC, 2010). Par conséquent, l'accessibilité ainsi que l'instantanéité de l'accès à l'information sont devenues une priorité pour beaucoup de personnes (Wei, 2008). De ce fait, les produits dérivés de la géomatique sur le web gagnent eux aussi à offrir des versions mobiles adaptées à des plus petits

écrans et qui utilisent les composantes internes des téléphones et des tablettes tels que le GPS et l'appareil photo.

Dans cet essai, nous visons à développer une boîte à outils générique permettant aux géomaticiens de concevoir des applications cartographiques web adaptatives fonctionnant à la fois sur un ordinateur et sur des supports mobiles. Cette nouvelle approche sera testée avec les données d'un projet de recherche en cours au Département de géomatique pour la gestion de données géospatiales.

### **1.1. Mise en contexte**

Dans le cadre de la maîtrise en sciences géographiques (cheminement de type cours en géodéveloppement durable) à l'Université de Sherbrooke, nous avons la chance, comme exigence finale, de travailler sur un sujet qui nous a intéressé lors de notre formation.

Dans ce contexte, le projet de développement d'une boîte à outils générique s'imbrique dans le projet de recherche de Mme Simone Moretti (étudiante à la maîtrise type recherche) sur l'élaboration d'un nouveau schéma du format GML (Geographic Markup Language) appelé le TrailGML. Ce nouveau standard agira comme « un langage commun pour permettre la visualisation et l'échange des données géospatiales des sentiers récréatifs qui sont produits par les différents partenaires » afin de faciliter l'interopérabilité de ce type de données (Moretti, 2016). À cet effet, le partenaire principal (la Fédération équestre du Québec) avec lequel Mme Moretti est associée désire, comme résultat final, obtenir une application cartographique web fonctionnant aussi bien sur un ordinateur qu'un téléphone intelligent. Par conséquent, Mme Moretti a déjà réalisé une version préliminaire de son application cartographique lors de la dernière année, mais celle-ci ne respecte pas encore toutes les demandes de ses partenaires. La construction d'une boîte à outils générique permettrait entre autres de l'aider à développer son ACM plus facilement sans avoir à se questionner sur la pertinence des librairies et des logiciels à utiliser.

La courte période de réalisation de l'essai limite l'amplitude de ce travail de recherche au simple développement de la nouvelle boîte à outils. Par conséquent, l'interface développée dans le cadre de ce travail existera uniquement pour effectuer des tests sur cette dernière et d'en tirer des conclusions. Quant au développement d'une interface professionnelle complète, la nouvelle boîte

à outils générique sera proposée à d'autres étudiants en géomatique de l'Université de Sherbrooke afin de les guider dans la construction d'une ACM.

## **1.2 Problématique**

Le grand potentiel des ACM attire plus que jamais les organismes et les entreprises à créer des applications. Toutefois, la construction de ces dernières peut demander des aptitudes avancées en informatique (Brus, 2009), notamment en programmation web et en programmation native, limitant la portée des projets aux professionnels de l'informatique. Par conséquent, plusieurs projets d'ACM pourraient être laissés à des développeurs ne possédant pas assez de connaissances dans les normes en géomatique comme les normes de l'Open Geospatial Consortium (OGC). Cet organisme international à but non lucratif œuvre dans l'élaboration et le maintien de standards géospatiaux très diversifiés. Puisqu'une grande majorité de logiciels cartographiques essaie de respecter les normes de l'OGC, il existe une grande interopérabilité entre les différents systèmes géographiques existants. Toutefois, certains logiciels respectent moins de normes que d'autres les rendant moins attrayants dans le cadre du développement d'une méthode réutilisable.

Dans le cadre de la maîtrise de type cours en géodéveloppement durable de l'Université de Sherbrooke, seulement deux cours touchent aux notions de programmation web alors que la complexité du développement des ACM ne cesse d'augmenter (Brus, 2009). Cette distanciation entre les compétences requises en programmation et l'enseignement est donc problématique. Puisque la demande pour ce genre d'application est grandissante (Reichenbacher, 2003), on assiste à une émergence de logiciels et de cadres de travail (Frameworks), propriétaires ou non, destinés à faciliter le processus de création et à améliorer la qualité des ACM. Bien que ceux-ci aident généralement dans certains aspects du développement des ACM, d'autres sont inefficaces, difficiles à utiliser ou simplement désuets. Malheureusement, il est souvent difficile d'évaluer le niveau de pertinence de ce type de logiciels puisqu'ils offrent des fonctionnalités très différentes quand ils sont comparés les uns aux autres. Pour cette raison, avant chaque nouveau projet d'ACM, un temps important doit être consacré à étudier ces logiciels et les comparer entre eux afin de retenir seulement ceux qui seraient compatibles et complémentaires. Ce processus d'évaluation peut être complexe et fastidieux à effectuer au début de chaque nouveau projet, mais si cette tâche est négligée, la qualité de l'ACM pourrait en écoper notamment sur les plans

de la pérennité et du respect des normes de l'OGC (Steiniger et Hunter, 2011). Il serait donc avantageux de proposer une boîte à outils déjà composée de logiciels adaptés aux ACM permettant de contourner cette étape de recherche. Il n'existe pas encore à ce jour de méthodes complètes permettant de développer des applications cartographiques sur un ordinateur et de les déployer, par la suite, sur des supports mobiles.

### **1.3 Objectifs**

L'objectif principal de ce projet vise à développer une méthodologie générique simple pour développer des ACM. En plus d'être réutilisable, l'idée est aussi de créer une solution flexible qui permettrait aux géomaticiens d'utiliser les logiciels avec lesquels ils sont les plus confortables. Ceci implique donc que les logiciels proposés dans la méthodologie puissent être remplacés par d'autres, à la préférence de l'utilisateur, sans influencer le but de cette dernière.

Dans le cadre de ce travail, les logiciels retenus pour faire partie de la boîte à outils seront ceux qui rencontrent un plus grand nombre de critères de sélection. Plus de détails seront donnés sur ces derniers dans la section méthodologie.

Afin d'atteindre l'objectif principal, plusieurs autres sous-objectifs ont été formulés. Soient :

1. La boîte à outils doit respecter un maximum de normes de l'OGC.
2. Sur les supports mobiles, permettre l'utilisation de l'appareil photo et du GPS.
3. L'application doit pouvoir s'ajuster automatiquement à plusieurs tailles d'écrans.

Concernant le premier sous-objectif, plus un logiciel respecte les normes de l'OGC, plus il sera en mesure de supporter un nombre élevé de formats géographiques (KML, WMS, WFS, etc.) et d'actions sur ceux-ci. Dans le développement d'une ACM, la carte est l'élément le plus important de l'application. Ce premier sous-objectif est donc primordial afin que la boîte à outils soit générique.

Le deuxième sous-objectif est plus spécifique à une utilisation de l'ACM sur un support mobile. En effet, celui-ci concerne l'utilisation des dispositifs internes du support mobile comme l'appareil photo et le GPS. La raison pour laquelle l'utilisation de ces dispositifs est un objectif de notre démarche réside dans le fait qu'ils offrent la possibilité de créer des ACM plus versatiles (Reichenbacher, 2001). Par exemple, le GPS pourrait permettre à un utilisateur d'observer sa

position sur une carte et de se repérer à travers un réseau de sentiers ou encore d'enregistrer un parcours et de le partager avec d'autres utilisateurs.

Le dernier sous-objectif vise à permettre une adaptation de l'ACM à toutes les tailles d'écrans. En effet, il est très important que, peu importe la dimension de l'écran sur lequel l'ACM est utilisé, l'utilisateur soit en mesure d'effectuer les mêmes opérations. Ce sous-objectif est donc très important parce que, s'il n'est pas respecté, l'application cartographique ne pourra pas fonctionner de façon efficace sur plusieurs plateformes.

Un autre volet d'objectifs est aussi important dans le cadre de cette étude : les objectifs pédagogiques. Ils visent le cheminement de nos aptitudes personnelles soit :

1. Développer nos compétences à effectuer un travail de recherche d'envergure en respectant une démarche méthodologique stricte.
2. Améliorer notre autonomie et notre aptitude à gérer un projet du début à la fin.
3. Consolider les connaissances que nous avons acquises dans les différents cours du programme.

En prenant en compte les différents objectifs décrits précédemment, un état des connaissances a été effectué sur plusieurs sujets permettant une pleine compréhension des ACM et de leur développement en 2016. Aussi, plus particulièrement pour l'élaboration de la méthodologie, une revue de la littérature a été réalisée pour déterminer le meilleur moyen de discriminer les différents logiciels à notre portée.

## **2. Cadre théorique**

Pour répondre aux différents objectifs du projet, il est nécessaire de bien comprendre la manière dont les ACM sont développés et fonctionnent. Pour ce faire, nous devons, dans un premier temps, décortiquer plusieurs ACM déjà existantes afin de cibler les composantes principales. En d'autres mots, est-il possible de déterminer des catégories de logiciels essentiels au fonctionnement des ACM? Par la suite, en nous basant sur ces catégories, nous serons en mesure de faire ressortir et d'étudier des logiciels qui pourront potentiellement faire partie de la boîte à outils. Il est aussi nécessaire de déterminer une méthode de discrimination de ces derniers afin de retenir les plus pertinents.

## **2.1 Étude d'ACM existantes**

Comme mentionné en introduction, les ACM sont de plus en plus populaires auprès des organismes et entreprises. Il en existe déjà beaucoup, simples ou complexes, déployées sur internet que nous pouvons visualiser gratuitement et utiliser comme exemples. En comparant deux d'entre elles, nous allons identifier quelques fonctionnalités récurrentes sur lesquelles nous pourrions nous baser pour développer notre méthodologie. Plus particulièrement, les deux ACM que nous allons étudier présentent chacune une cartographie d'un réseau axé sur le géodéveloppement durable.

### **2.1.1 Exemple #1 : application cartographique de la fédération québécoise des clubs quads (FQCQ)**

La première ACM est celle de la FQCQ (figure 1). Principalement conçue pour consulter le réseau de quads au Québec, l'application est également très complète en ce qui a trait aux différents critères de recherche et au visionnement des services disponibles sur le réseau. En effet, il est possible de filtrer les différents sentiers selon la saison, le type de sentier, l'état de surfaçage du sentier et son appartenance (à qui appartient le sentier).

Pour les fonctions des bases, cette application cartographique possède l'avantage d'avoir des éléments adaptatifs à la taille du navigateur web. Par exemple, la barre de navigation se transforme en un menu déroulant lorsque l'application est visionnée sur un plus petit écran. Toutefois, ce ne sont pas tous les éléments de la carte qui s'ajustent tels que les menus sur les côtés et la carte qui ne sont pas adaptatifs. En effet, mise à part la barre de navigation, l'espace cartographique n'est pas conservé sur des plus petits écrans. Pour remédier à ce problème, la FQCQ a développé une version mobile de l'application appelée « iquad » qui est disponible sur Google Play et l'App Store. Sur celle-ci, le positionnement des éléments est différent de la version disponible sur internet ce qui permet une utilisation optimale sur des supports mobiles.

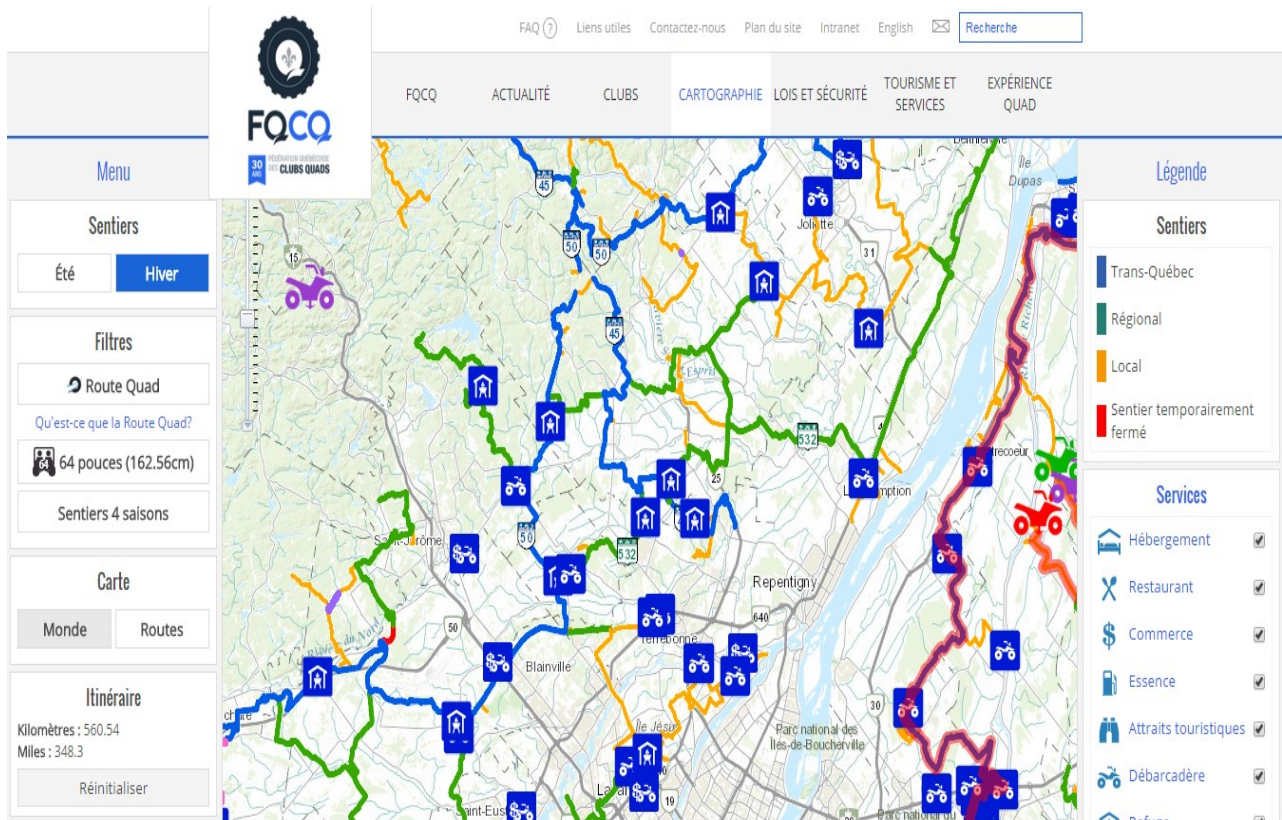


Figure 1 : capture d'écran de l'ACM de la FQCQ sur un ordinateur. L'image a été extraite de leur site internet. (FQCQ, 2016)

Au niveau des fonctionnalités de l'application, le réseau de sentiers affiché est 100 % interactif. Un clic sur un sentier le sélectionne en rouge sur la carte et une fenêtre informative s'affiche avec les informations descriptives du sentier. Cela est aussi vrai pour tous les autres icônes de service.

Une autre fonctionnalité intéressante est la possibilité de construire un itinéraire en utilisant les icônes de quads. Une fois les icônes positionnés, l'application va déterminer le trajet à effectuer en fonction de ceux-ci et mettre en surbrillance les sentiers qui le compose sur la carte. Un point intéressant par rapport à cette fonction d'itinéraire est qu'il est possible de positionner plusieurs points de départ et d'arrivée en même temps. Cela est pratique pour un utilisateur qui désire afficher plus d'un trajet sur une même carte.



### 2.1.2 Exemple #2 : application cartographique de Vélo-Québec

Le deuxième exemple est l'application cartographique de Vélo-Québec (figure 2). Cette application a été développée pour rendre disponible sur internet une cartographie des routes vertes et des pistes cyclables au Québec. Elle permet aussi de signaler un problème sur l'une d'entre elles (route fermée, chaussée endommagée) et d'y observer la localisation de différents services d'hébergement et de restauration dans le but de planifier un itinéraire. Elle est très similaire à celle de la FQCQ en termes d'offre de service.

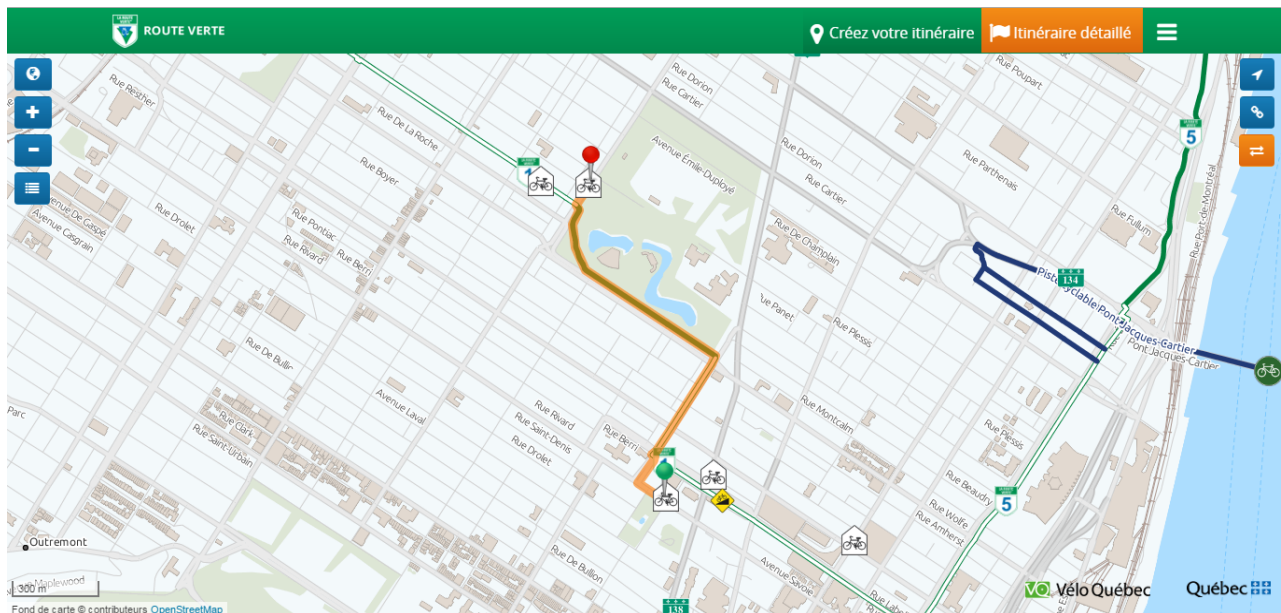


Figure 2 : Capture d'écran de l'ACM de Vélo-Québec sur un ordinateur tirée de leur site internet. (Vélo-Québec, 2016)

Au niveau des fonctionnalités de base, l'application possède également la caractéristique de s'ajuster automatiquement à des tailles d'écrans différentes. De ce fait, il est possible de l'utiliser sur différents supports tout en gardant la même facilité d'utilisation. Les figures 2 et 3 montrent comment la disposition des éléments de l'application se replace en fonction de la dimension de l'écran. La figure 2 montre l'application vue sur un ordinateur alors que la figure 3 montre l'application vue sur un téléphone intelligent. En comparant le haut des deux images, on s'aperçoit qu'il y a eu une contraction de la barre de navigation située en haut de la carte. En effet, le texte « créer votre itinéraire » et « itinéraire détaillé » disparaît lorsque l'on visionne l'application sur un plus petit écran pour laisser place seulement aux icônes. De plus, on observe aussi que l'itinéraire sélectionné reste toujours positionné au centre de l'écran malgré le fait que

tous les autres éléments de l'application se sont repositionnés. Ce repositionnement est donc une méthode très efficace pour conserver l'ergonomie de l'application d'un support à l'autre.

En termes de fonctionnalités plus complexes, cette application permet d'effectuer quelques opérations sur la carte comme la sélection d'un itinéraire. Pour ce faire, l'utilisateur doit sélectionner un point de départ et un point d'arrivée basé sur des coordonnées ou des lieux existants sur la carte (hôtels, restaurants, etc.) et l'outil calculera automatiquement le trajet à effectuer en plus de l'afficher sur la carte. La figure 3 présente l'exemple d'un trajet d'un point A à un point B représenté par la ligne orange. Une fois terminé, l'utilisateur peut décider de partager une carte contenant un trajet avec d'autres personnes en copiant le lien URL propre à celle-ci.



Figure 3 : capture d'écran de l'ACM de Vélo-Québec sur un téléphone intelligent extraite de leur site internet. (Vélo-Québec, 2016)

### 2.1.3 Résumé des observations

En comparant les deux exemples d'ACM sur les sentiers récréatifs, on remarque des fonctionnalités communes. D'abord, ces dernières sont des interfaces adaptatives qui permettent de s'adapter à différentes tailles d'écrans. L'application de la FQCQ possède même une version

spécifiquement adaptée aux supports mobiles. De plus, ces deux applications permettent d'afficher différents types de couches. Certaines sont statiques alors que d'autres sont interactives et permettent d'obtenir de l'information précise sur un objet de la carte. Ces couches sont toujours superposées à une carte de fond qui possède ses propres éléments tels que le réseau routier ou même la topographie. D'ailleurs, ces deux applications permettent d'alterner entre plusieurs couches de base afin de raffiner l'expérience de l'utilisateur selon ses besoins. Finalement, on constate avec ces deux ACM la possibilité de construire son propre itinéraire et de l'afficher sur la carte.

## **2.2 Les catégories essentielles de logiciels**

Les exemples présentés sur le développement des ACM mettent de l'avant certaines fonctionnalités. Ces dernières s'obtiennent avec certains logiciels que nous pouvons classés en différentes catégories :

- (1) l'affichage d'une couche de fond et des couches secondaires (statiques ou interactives) ainsi que la création d'un itinéraire s'effectue principalement via une API (Application Programming Interface) cartographique (Neumann, 2008);
- (2) l'aspect adaptatif d'un site web ou d'une application peut s'effectuer à l'aide du langage de style CSS (Cascading Style Sheet) ou, préférablement, avec l'utilisation d'un cadre de travail adaptatif ou « Framework adaptatif » (Neumann, 2008).

Avec ces deux catégories de logiciels, nous serions théoriquement en mesure de reproduire les mêmes fonctionnalités observées sur les deux exemples précédents. Cependant, nous décrivons à ce niveau seulement la partie « client » et nous n'aborderons pas la partie « serveur » d'un ACM, et notamment le serveur cartographique. En effet, le serveur cartographique a comme but de gérer, de modifier et de diffuser des données géographiques que nous pouvons par la suite superposer à d'autres couches sur une interface client. Si l'on possède une base de données, il est alors possible de connecter ses propres couches sur un serveur cartographique (Beesley, Childs, 2004). Dans ce contexte, nous pouvons déterminer qu'il existe trois catégories de logiciels au développement d'une ACM, soient les API cartographique, les cadres de travail adaptatifs et finalement les serveurs cartographiques. Toutefois, l'utilisation d'un serveur cartographique n'est pas forcément nécessaire si l'on dispose d'un jeu de données en XML (comme le GML et le KML) puisque leur intégration peut s'effectuer directement à partir de l'API cartographique.

### 2.2.1 Les API cartographiques

Les API cartographiques ou « Cartographic Application Programming Interface » sont des fonctionnalités qui permettent de développer des applications pour visualiser et manipuler une carte sur internet, la plupart du temps interactive, pouvant posséder plusieurs couches de base et superposées (Hoarau, 2012). Les API utilisent généralement le langage JavaScript, un langage populaire reconnu et utilisé par tous les navigateurs web, ce qui permet d'offrir une multitude d'interactions avec la carte (Leaflet, 2016). La majorité des API cartographiques permettent d'afficher une légende des couches présentes, un outil pour augmenter ou réduire l'échelle cartographique ou encore d'afficher une vue généralisée aidant à la navigation. Toutefois, il est à noter que certaines API vont offrir une gamme plus complète de fonctionnalités que d'autres. Pour combler cette déficience, certaines d'entre elles permettent l'utilisation de composantes externes appelées « plug-in ». Développées dans un but précis, les composantes externes visent à rendre disponible l'utilisation de nouvelles fonctionnalités complémentaires à celles déjà offertes par l'API (Leaflet, 2016). De cette manière, la plupart des API sont de nos jours en mesure de répondre aisément aux différents besoins des organisations et des entreprises en matière de cartographie web (Hoarau, 2012).

En faisant suite à ce qui a été dit, dans le cadre de cette étude, la revue de la littérature a permis d'identifier quelques API cartographiques qui semblent être particulièrement populaires.

- *Leaflet* : cette API de type « open source » est utilisée par un nombre important de sites internet comme Facebook, Foursquare et Pinterest. Relativement récente, la version stable actuelle a été dévoilée le 18 novembre 2013 et a dernièrement été mise à jour en octobre 2015. Parmi ses points forts, on retrouve sa faible dimension (seulement 33Ko), une simplicité efficace et une gamme très étoffée de composantes externes (Leaflet, 2016).
- *Google Map API* : contrairement à Leaflet, Google Map est une application de type propriétaire et entraîne son lot d'avantages et d'inconvénients. Par exemple, l'utilisation de cette API peut être payante si l'on dépasse 25 000 visiteurs par jour et il ne peut pas y avoir de la publicité sur notre carte. En revanche, Google Map API possède un lot de

fonctionnalités très diversifiées, la capacité de s'adapter à plusieurs supports mobiles, une documentation exemplaire et un support incontesté (Google developers, 2016).

- *OpenLayers 3* : la troisième version de cette API cartographique « open source » est très polyvalente et supporte de nombreuses normes de l'OGC telles que WMS et WFS ainsi que des données vectorielles de type GeoRSS, KML et GML. Très robuste, elle est simple d'utilisation et possède l'avantage d'être souvent mise à jour pour suivre les dernières tendances. L'application OpenLayers est particulièrement appréciée puisqu'elle offre plusieurs fonctionnalités de géotraitement avancées, mais aussi parce qu'elle est très rapide et, tout comme Google Map API, s'adapte automatiquement aux supports mobiles (OpenLayers, 2016).
- *PolyMaps* : cette API « open source » diffère légèrement des trois autres bibliothèques puisque son but principal, mis à part l'affichage d'une carte web, est de rendre la visualisation de projet comprenant beaucoup de données plus rapides en utilisant le SVG pour rendre les tuiles vectorielles. En plus de l'efficacité et de la polyvalence que cela procure, l'utilisation du SVG permet, entre autres, d'utiliser le langage CSS pour le style éléments de notre carte (Polymaps, 2016). Toutefois, il n'y a pas encore beaucoup de composantes externes disponibles et le code nécessaire à son fonctionnement est assez complexe.

### **2.2.2 Les cadres de travail adaptatifs**

Étant une pièce centrale des applications multiplateformes en général, le cadre de travail adaptatif permet de repositionner les éléments d'une page web selon la taille de l'écran. De cette façon, une application qui est développée de façon « adaptative » peut être utilisée aussi bien sur un téléphone intelligent que sur un ordinateur (Reichenbacher, 2003). Pour y arriver, le cadre de travail adaptatif utilise ce qu'on appelle des « media queries » (Bootstrap, 2016). Les requêtes médias, apparus avec l'arrivée du langage CSS3, permettent d'afficher certains éléments en se basant sur une ou plusieurs particularités d'un écran. Parmi ces particularités, on trouve la largeur, la hauteur, l'orientation et la résolution de l'écran. La figure 4 montre un exemple d'une requête média qui utilise comme critère la largeur de l'écran, mesurée en pixels, créant ainsi une limite minimale et maximale. Comme résultat, cette requête va seulement afficher les éléments

de la page web affectés par celle-ci sur un support dont la largeur de son écran se situe entre les deux limites soit 401 pixels et 600 pixels de largeur (Adobe, 2011).

```
media="screen and (min-width: 401px) and (max-width: 600px) "
```

Figure 4 : Exemple de requête média avec le langage CSS tirée du site internet d'Adobe. (Adobe, 2016)

Développé à partir de ces requêtes média, le fonctionnement des cadres de travail adaptatifs repose donc en grande partie sur l'assignation des éléments HTML (Hypertext Markup Language) par classes à des dimensions d'écran. Par conséquent, les développeurs peuvent aisément créer des applications web qui s'adaptent automatiquement aux différents supports sans avoir à écrire eux-mêmes les requêtes média. La figure 5 montre, pour le cadre de travail Bootstrap, quelle classe est associée à quelle taille d'écran ainsi que les propriétés qui lui sont associées.

	Extra small devices Phones (<768px)	Small devices Tablets (≥768px)	Medium devices Desktops (≥992px)	Large devices Desktops (≥1200px)
<b>Grid behavior</b>	Horizontal at all times		Collapsed to start, horizontal above breakpoints	
<b>Container width</b>	None (auto)	750px	970px	1170px
<b>Class prefix</b>	.col-xs-	.col-sm-	.col-md-	.col-lg-
<b># of columns</b>	12			
<b>Column width</b>	Auto	~62px	~81px	~97px
<b>Gutter width</b>	30px (15px on each side of a column)			
<b>Nestable</b>	Yes			
<b>Offsets</b>	Yes			
<b>Column ordering</b>	Yes			

Figure 5 : Définition des classes pour le cadre de travail Bootstrap provenant de leur site internet. (Bootstrap, 2016)

Dans cet exemple, la taille d'un écran (évaluée en pixels) a été associée à des supports tels qu'un téléphone intelligent ou une tablette avec des classes prédéfinies. Toutefois, tous les cadres de travail de ce type ne fonctionnent pas de la même façon et possèdent des avantages et des inconvénients.

Dans la section suivante, nous allons énumérer quelques cadres de travail adaptatifs les plus populaires dans le développement d'ACM et présenter quelques-unes de leurs caractéristiques les plus intéressantes.

- *Twitter Bootstrap* : utilisant les langages HTML, CSS et JavaScript, Bootstrap est un cadre de travail « open source » offrant plusieurs éléments HTML et JQuery qui peuvent être très pratiques dans le développement d'ACM. Comme mentionné précédemment, son système de classe est construit spécifiquement pour permettre la conception d'applications multiplateformes (Bootstrap, 2016). De plus, c'est un cadre de travail qui est extrêmement populaire auprès de la communauté de développeurs web (Borges *et al.*, 2015).
- *Foundation 6* : il est semblable à Bootstrap. Foundation 6 a été construit dans le même but de faciliter le développement d'applications multiplateformes à partir du web. Ce cadre de travail est particulièrement bien adapté pour les supports de petites tailles. De plus, Foundation 6 a l'avantage de générer des pages HTML faciles à comprendre même avec peu de connaissances en programmation web. Pour résumer, c'est un produit très complet (Zurb, 2016).
- *Skeleton* : avec environ 400 lignes de code, Skeleton est un cadre de travail « open source » très léger qui ne possède que l'essentiel. Développé pour être un point de départ, Skeleton a été construit dans le but de s'intégrer très rapidement à de petits projets qui ne nécessitent pas la lourdeur des plus gros cadres de travail (Skeleton, 2016).
- *HTML5 Boilerplate* : avec des fonctionnalités prometteuses, ce cadre de travail offre la possibilité de créer des applications d'envergure très performantes, incluant un support sur la grande majorité des navigateurs web. Par contre, le produit n'a pas été construit spécifiquement pour le développement d'applications multiplateformes comme d'autres cadres de travail décrits précédemment. Cependant, il permet d'organiser le code selon les préférences de l'utilisateur (HTML5 Boilerplate, 2016).
- *Pure* : tout comme Skeleton, Pure est un cadre de travail extrêmement léger avec une taille de seulement 4 ko une fois compressée. Un de ses avantages réside dans le fait que son système de disposition des éléments HTML est très élaboré. De plus, ses modules

fonctionnent indépendamment les uns des autres permettant de l'adapter facilement à des projets déjà existants en prenant tout juste le nécessaire (Yahoo, 2014).

- *Semantic UI*: il est très intéressant dans la mesure où il offre beaucoup d'éléments HTML simples et complexes qui peuvent être utilisés rapidement dans la conception d'une application multiplateforme. De ce fait, un développeur peut seulement se concentrer sur le côté client de son application sans avoir à utiliser de bibliothèques JavaScript externes comme JQuery (Semantic, 2016).

Évidemment, les cadres de travail adaptatifs décrits ci-dessus ne représentent qu'une petite partie de tous les produits de ce genre. Toutefois, ces derniers ressortent du lot par leur popularité et leur facilité d'utilisation.

### **2.2.3 Les serveurs cartographiques**

Les serveurs cartographiques servent à gérer et à diffuser sur le web des données géographiques, provenant par exemple de fichiers shapefiles ou d'une base de données. Une fois diffusées, celles-ci pourront être affichées sur un navigateur web via une API cartographique sous la forme d'une image ou de données vectorielles. La plupart du temps, ces serveurs peuvent être contrôlés à partir de langages tels que PHP, Python et JavaScript, ou directement à partir d'une page HTML (Johnston, 2013). Les caractéristiques des serveurs cartographiques varient beaucoup suivant les logiciels. Certains, plus complets, vont permettre de modifier le style de certaines couches avec un fichier SLD (Styled Layer Descriptor), de changer la projection des couches ou encore d'ajouter des métadonnées (Peterson, 2008). Plusieurs serveurs cartographiques sont « open source » et peuvent être utilisés gratuitement afin d'afficher des données sur une carte

Une liste a été élaborée afin de présenter trois serveurs cartographiques accompagnés d'une brève description sous forme de points et de quelques remarques personnelles. Ceux-ci ont été présélectionnés notamment parce qu'ils ressortaient le plus souvent dans la littérature. Toutefois, il existe d'autres solutions propriétaires comme ArcGIS Server qui n'ont pas été retenus dans le cadre de ce travail puisque nous désirons se concentrer sur l'aspect « open source ».



Tableau 1 : Descriptions et remarques pour chaque serveur cartographique

Nom	Description	Remarques personnelles
<b>GeoServer</b>	<ul style="list-style-type: none"> <li>➤ Open source</li> <li>➤ Utilise les standards de l'OGC</li> <li>➤ Fonctionne avec beaucoup d'applications SIG telles que Google Map, Google Earth et ESRI ArcGIS, QGIS, PostGIS</li> <li>➤ Fonctionnalités nombreuses et avancées</li> </ul>	<ul style="list-style-type: none"> <li>➤ Simple d'utilisation</li> <li>➤ Beaucoup de documentation</li> </ul>
<b>MapServer</b>	<ul style="list-style-type: none"> <li>➤ Open source</li> <li>➤ Très rapide et personnalisable</li> <li>➤ Fiable</li> <li>➤ Supporte beaucoup de normes de l'OGC</li> </ul>	<ul style="list-style-type: none"> <li>➤ Références positives venant de la communauté de géomatique sur le web</li> </ul>
<b>Map Fish</b>	<ul style="list-style-type: none"> <li>➤ Open source</li> <li>➤ Respect de plusieurs normes de l'OGC</li> <li>➤ Outils spécifiques pour faire de l'édition et des requêtes spatiales</li> <li>➤ Offre beaucoup d'outils intéressants</li> <li>➤ Combinaison d'ExtJs, GeoEXT et OpenLayers</li> </ul>	<ul style="list-style-type: none"> <li>➤ Dernière mise à jour 2009</li> </ul>

En définitive, même si certains serveurs cartographiques possèdent des fonctionnalités avancées, ils accomplissent tous le même rôle, c'est-à-dire la diffusion des couches sur internet avec une norme définie. Pour cette raison, ils représentent une pièce centrale d'une ACM qui ne peut être négligée dans le cadre du développement d'une méthodologie générique.

### 2.3 Comment convertir une application web en application mobile?

En utilisant les trois catégories de logiciels présentées précédemment, nous sommes théoriquement en mesure de construire une ACM. L'API cartographique permet de gérer les interactions avec la carte, le cadre de travail adaptatif va repositionner les éléments d'une page web selon la taille d'un écran et le serveur cartographique permet la gestion et la diffusion des

couches géographiques supplémentaires. Toutefois, sur un support mobile, il existe des avantages à utiliser une application mobile « native ». Par définition, les applications natives sont des applications à part entière qui fonctionnent sur des systèmes d'exploitation mobiles comme Android, Windows mobile et IOS (Bathelot, 2011). Puisqu'elles communiquent directement avec le système d'exploitation de l'appareil, elles offrent des fonctionnalités auxquelles une application cartographique sur un navigateur web n'a pas accès. Parmi ces fonctionnalités, l'application native peut utiliser les composants externes de l'appareil comme la caméra, le GPS ainsi que la liste de contacts en plus d'augmenter la rapidité de l'application (Michaels *et al.*, 2015). Les possibilités qu'offrent ces composants dans une application cartographique sont très nombreuses. Pour cette raison, nous devons inclure dans notre développement un moyen de transformer notre application web en application mobile. Malheureusement, les technologies que nous utilisons jusqu'à présent ne nous permettent pas de créer des applications mobiles avec la possibilité d'utiliser l'appareil photo, le GPS, etc. Puisque nous voulons éviter la voie de la programmation native, beaucoup trop compliquée pour des géomaticiens, nous devons nous tourner vers un autre type de technologie : les convertisseurs d'application web vers des applications natives.

Les convertisseurs ont été créés dans le but de faciliter le développement d'applications natives, généralement en utilisant les langages HTML, CSS et JavaScript (Hasan, 2012). Le but est simple : permettre de créer des applications natives avec les technologies du web et d'avoir le même fonctionnement peu importe la plateforme mobile. La liste ci-dessous présente quelques convertisseurs populaires dans la littérature avec une brève description et des remarques personnelles.

Tableau 2 : Description et commentaire pour chaque logiciel de conversion

Nom	Description	Remarques
<b>Appcelerator</b>	➤ Open Source	➤ Ne permet pas de créer des applications pour les téléphones Windows
<b>Titanium</b>	➤ Utilise le langage JavaScript	➤ Offre un cadre de travail MVC optionnel : Alloy
	➤ Supporte un très grand nombre de composants des supports mobiles	
	➤ Possibilité d'utiliser la librairie JQuery	

<b>Phone Gap Build</b>	<ul style="list-style-type: none"> <li>➤ Distribué par Adobe, basé sur la plateforme d'Apache Cordova</li> <li>➤ Permet de transformer une application web en application native en quelques secondes</li> <li>➤ Fonctionne sur tous les systèmes d'exploitation</li> <li>➤ Permet d'accéder aux composantes de base des supports mobiles</li> </ul>	<ul style="list-style-type: none"> <li>➤ Très grande communauté</li> <li>➤ Pas besoin de télécharger quoi que ce soit sur son ordinateur, mise à part l'application compilée</li> </ul>
<b>App Deck</b>	<ul style="list-style-type: none"> <li>➤ Open Source</li> <li>➤ Fonctionne en partie comme la librairie JQuery Mobile</li> <li>➤ Utilise le langage HTML pour l'affichage de l'application, mais le reste est en code natif</li> </ul>	<ul style="list-style-type: none"> <li>➤ Très peu de références</li> <li>➤ Fonctionne seulement sur Android et iOS</li> <li>➤ Version différente pour les deux systèmes d'exploitation</li> </ul>
<b>Cordova</b>	<ul style="list-style-type: none"> <li>➤ Open Source</li> <li>➤ Support pour tous les systèmes d'exploitation</li> <li>➤ Utilise les langages HTML, CSS et JavaScript</li> <li>➤ Un seul code pour toutes les plateformes</li> <li>➤ Supporte le mode hors ligne</li> </ul>	<ul style="list-style-type: none"> <li>➤ Projet d'envergure largement utilisé</li> <li>➤ Sert de base pour PhoneGap</li> <li>➤ Offre énormément de plug-ins et de documentation</li> </ul>

Notre étude nous a permis d'identifier les différentes catégories de logiciels au développement d'une ACM. Pour chacune de ces catégories, nous avons été en mesure d'élaborer une liste de logiciels les plus prometteurs dans le cadre de l'objectif principal de cet essai : le développement d'une méthodologie générique pour concevoir des ACM. Toutefois, puisque nous devons sélectionner un seul logiciel par catégorie, nous devons déterminer une façon de les classer entre eux. Ce classement sera présenté plus en détail dans la partie méthodologique de cet essai.

## 2.4 Discrimination et classement des logiciels

Afin de sélectionner quel logiciel sera associé à chaque catégorie, nous devons effectuer un classement. Chaque logiciel possède des avantages et des inconvénients, les rendant difficiles à évaluer simplement à partir d'une revue de la littérature. De ce fait, afin d'effectuer des choix impartiaux, il est nécessaire d'utiliser une méthode d'aide à la décision. Il existe une multitude de méthodes qui permettent de guider le choix d'un scénario ou même d'offrir un classement des meilleurs choix dans une situation donnée. Une d'entre elles, l'analyse multicritère, offre la possibilité de poser des critères et de leur attribuer un poids en fonction de leur importance relative. Il permet également de modéliser les différents scénarios en fonction des critères sélectionnés. Ces derniers pourraient être définis comme étant « des facteurs à prendre en considération pour évaluer un scénario donné ou pour apprécier une occasion d'action » (Prévil *et al.*, 2003).

L'analyse multicritère repose essentiellement sur des actions (ou procédures) qui vont découper un problème complexe en de plus petites entités. Ces entités pourraient être perçues comme étant les critères décisionnels. La résultante de cette analyse permettra de mettre à l'avant un scénario se présentant comme le plus pertinent vis-à-vis de l'entièreté des critères décisionnels. L'analyse multicritère est très pratique dans la mesure où elle permet d'évaluer des scénarios qui possèdent des critères très différents les uns des autres. Par conséquent, il est possible de mélanger les critères de nature quantitative et qualitative sans problème (Prévil *et al.*, 2003).

Il existe plusieurs méthodes d'analyse multicritère qui se différencient surtout au niveau des ensembles de solutions qu'elles proposent. On peut les classer en quatre catégories de problématiques définies par Roy (1985). Soient :

1. de choix : lorsque le résultat recherché découle d'une procédure de sélection;
2. de tri : pour une procédure d'affectation;
3. de classement : pour établir des classes de priorités;
4. de description : pour améliorer la compréhension d'ensemble de différentes actions afin de mettre en contexte les conséquences possibles des interventions.

## **2.5 Bilan de l'état des connaissances et de la revue de littérature**

Dans un premier temps, nous avons présenté deux exemples d'ACM dans le domaine des sentiers récréatifs au Québec. Il a été possible de déterminer des fonctionnalités communes, autant sur le fonctionnement des applications que sur les outils qu'elles offraient. À partir de nos observations, nous avons pu élaborer une liste de fonctionnalités de base que nous avons par la suite traduites en catégories de logiciel que nous jugeons essentielle au développement de notre méthodologie. Cette dernière vise à répondre aux différents sous-objectifs que nous avons spécifiés dans ce travail.

Trois catégories de logiciel essentielles ont été considérées à partir de la revue de littérature : les API cartographiques, les cadres de travail adaptatifs et les serveurs cartographiques. En effet, les API cartographiques représentent des bibliothèques JavaScript qui permettent d'afficher une carte sur un navigateur web et de gérer les interactions avec ce dernier. Les cadres de travail adaptatifs, quant à eux, servent essentiellement à adapter une page web correctement sur des petits et grands écrans. En terminant, les serveurs cartographiques sont des logiciels qui œuvrent en arrière-plan sur un serveur pour la gestion des couches géographiques et de les diffuser sous différents formats comme WMS ou WFS. Par la suite, le serveur cartographique se connecte à une API cartographique et permet d'afficher une ou plusieurs couches de données sur une carte web.

Pour continuer, la revue de littérature nous a également permis d'identifier une manière simple et efficace de transférer une application web en application mobile sans avoir à programmer de code natif. En effet, les convertisseurs d'applications sont en mesure de prendre une application et d'en faire une version mobile. En ajoutant quelques lignes de code, ces outils de conversion rendent disponible l'utilisation des périphériques des appareils mobiles comme la caméra, l'appareil photo et le GPS.

Dans un autre ordre d'idée, nous avons effectué une revue scientifique sur les différentes méthodes d'analyse multicritère. En effet, nous devons déterminer une façon de discriminer les logiciels entre eux.

### **3. Matériel et méthode**

#### **3.1 Site d'étude**

L'objectif premier de cet essai est de développer une méthodologie et non d'effectuer des traitements géospatiaux ou de produire des cartes thématiques sur un secteur en particulier. Dans ce contexte, il est relativement abstrait de parler de site d'étude. Cependant, puisque cet essai repose en partie sur le travail de recherche de Mme Simone Moretti et que, par le fait même, nous utiliserons une partie de ses données géospatiales pour tester notre boîte à outils via une application prototype, nous pouvons déterminer un site d'étude en fonction des données utilisées par Mme Moretti. Pour cette raison, puisque la zone d'étude du travail de Mme Moretti se concentre essentiellement sur la province du Québec, nous pouvons affirmer que nous travaillerons avec des données portant sur cette dernière. Toutefois, il est important de mentionner que nous aurions pu utiliser d'autres données afin de tester notre boîte à outils et que le site d'étude a peu d'importance.

#### **3.2 Données**

Les données utilisées se divisent en deux grandes sections : les données relatives à l'analyse multicritère et les données géographiques que nous utiliserons pour faire les tests. Dans un premier temps, pour être en mesure d'effectuer l'analyse multicritère, nous devons recueillir des données correspondant à chacun des critères que nous avons préalablement définis dans les sections précédentes. La grande majorité des données utilisées pour la pondération des critères proviennent du site internet GitHub, un site internet d'envergure qui permet d'héberger le code des logiciels. Ce site internet est utilisé par une très grande communauté de programmeurs, d'informaticiens et même de géomaticiens qui désirent contribuer à un projet ouvert déjà existant ou rendre disponible leur propre projet. Donc, à partir de ce site, nous serons en mesure d'aller chercher les données correspondant à la majorité de nos critères tels que la popularité (représentée par le classement) et la pérennité (représentée par le nombre de contributeurs).

Pour les autres critères, comme le support de certains standards de l'OGC, le poids des bibliothèques ou le support des navigateurs web, les données seront recueillies sur les sites internet des différents produits. En effet, si cette information n'est pas déjà disponible sur le site GitHub, il est possible d'avoir accès à la documentation ou le site internet du fabricant.

Pour ce qui est des données que nous utiliserons pour tester notre méthodologie, Mme Moretti a mis à notre disposition un serveur cartographique (Geoserver) qui contient des données géographiques fournies par la FEQ (Fédération équestre du Québec) et leurs partenaires qui représentent des tracés de sentiers équestres, certains points d'intérêts et des métadonnées (noms et descriptions). Finalement, la couche géographique de base utilisée dans notre application prototype proviendra d'OpenStreetMap.

### 3.3 Méthodologie

Pour poursuivre, la méthode utilisée pour construire la boîte à outils (méthodologie de développement d'une ACM) se divise principalement en cinq grandes étapes représentées sur l'organigramme méthodologique ci-dessous.



Figure 6 : Principales étapes de la méthodologie

La première étape vise à sélectionner le type d'analyse multicritère correspondant le plus à nos besoins suivi de l'acquisition des données pour chaque logiciels et librairies. Ces données seront, par la suite, formatées afin d'être facilement comparées les unes aux autres. L'étape qui s'en suit est la pondération des critères et finalement la réalisation de l'analyse multicritère.

Ultimement, l'analyse multicritère permettra de sélectionner, pour chacune des catégories, le logiciel ayant obtenu le meilleur score final. Ainsi, ce dernier sera intégré dans la boîte à outils que nous testerons par la suite en créant une ACM servant à vérifier si la boîte à outils obtenue permet d'atteindre les différents objectifs de ce travail. Le schéma conceptuel présenté ci-dessous représente ce cheminement.

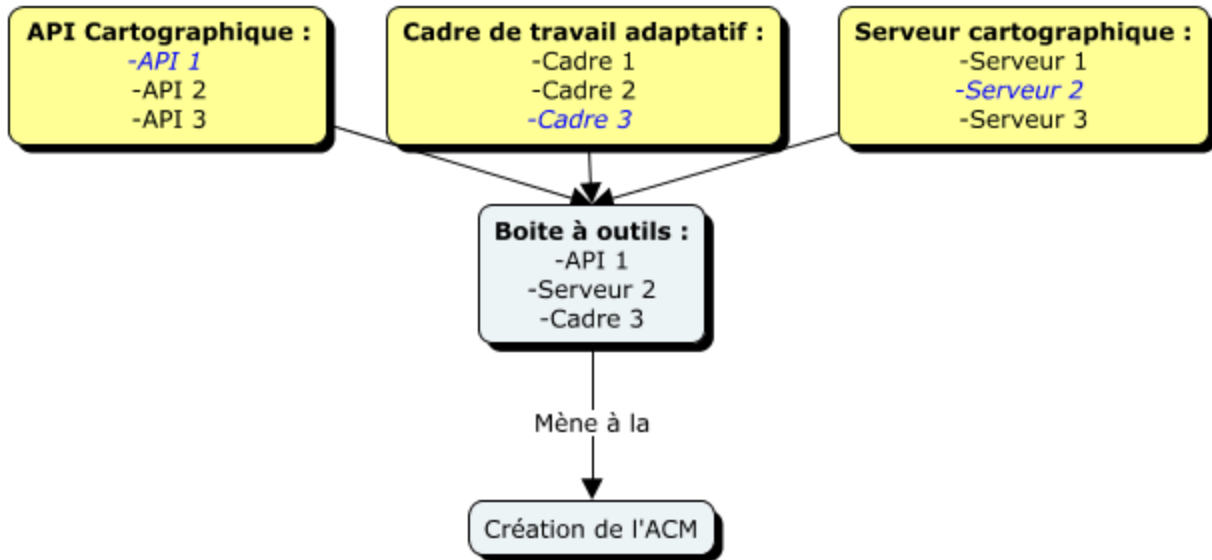


Figure 7 : Organigramme démonstratif de construction de la boîte à outils

### 3.3.1 Choix de la méthode d'analyse multicritère

L'ensemble des méthodes décrites dans la revue de la littérature proposent toutes un arbre décisionnel différent. Il est donc important de bien définir la problématique de départ afin de sélectionner une méthode qui correspond à la nature de la décision que l'on souhaite effectuer. Par conséquent, dans le cadre de cet essai, puisque l'objectif principal repose sur un classement (en ordre de pertinence) des différents logiciels, nous nous concentrerons uniquement sur les méthodes dites de classement ou de surclassement. Dans ce contexte, le classement des logiciels consistera à donner à chaque logiciel un rang en rapport avec son niveau de pertinence à l'intégrer dans la méthodologie. Par la suite, il sera possible de classer les différents logiciels les uns par rapport aux autres. Afin d'arriver à cet objectif, nous devons étudier une catégorie de méthode en particulier : les méthodes de surclassement.

#### 3.3.1.1 Les méthodes de surclassement

Dans les méthodes dites de surclassement, il existe deux grandes catégories : les méthodes ÉLECTRE (Élimination Et Choix Traduisant La Réalité) et PROMETEHEE (Preference Ranking Organisation Method For Enrichment Evaluations). Celles-ci demandent généralement très peu d'informations pour être utilisées, ce qui est pratique dans certains scénarios où



l'information se fait rare. Les résultats de ce type d'analyse sont très souvent pertinents et permettent d'effectuer le choix d'un meilleur scénario ou de la meilleure action (Caillet, 2003).

Pour débiter, les méthodes ÉLECTRE, développées dans les années 1970 par Bernard Roy, demandent de générer une liste de critères en fonction du problème auquel le décideur est confronté et de leur attribuer des poids. C'est à partir de cette liste que les différentes actions seront comparées. Les différentes méthodes ÉLECTRE sont assez similaires excepté quelques petites différences. Par exemple, la méthode ÉLECTRE IV élimine l'étape de pondération des critères présente dans l'ensemble des autres méthodes ÉLECTRE (Caillet, 2003). Généralement, la résultante des méthodes ÉLECTRE est un graphe orienté qui représente les liens présents entre les différentes actions. Il est donc possible d'avoir plusieurs bons scénarios constituant les pointes du graphe. Pour cette raison, puisque l'un des sous-objectifs de notre projet est de proposer une méthodologie flexible, c'est-à-dire qu'une personne peut changer un des logiciels que nous choisirons, cette méthode n'est pas idéale.

Les méthodes PROMETHEE (Preference Ranking Organisation Method for Enrichment Évaluations) sont des méthodes qui ont été conçues dans les années 1980 par Jean-Pierre Brans et Philippe Vincke (Henriet, 2000). Elles sont basées sur le fonctionnement des méthodes ÉLECTRE donc elles sont relativement simples à utiliser. Il suffit de définir une pondération sur les critères sélectionnés. Dans cette catégorie de méthode, les plus utilisées sont PROMETHEE I et II, possédant une démarche initiale très semblable, mais dont le but ultime est différent. En effet, la méthode PROMETHEE I vise à faire ressortir des relations dans le classement (partiel) des actions alors que PROMETHEE II permet d'arriver à un classement (total) de tous les scénarios en fonction de leur note finale (Caillet, 2003). Cela est nettement plus intéressant dans le cadre de notre travail puisque les meilleures options sont celles qui obtiendront le bilan positif le plus élevé. De ce fait, la méthode PROMETHEE II départage toutes les actions de la meilleure à la moins bonne (Kourouma, 2007).

### ***3.3.1.2 Choix de la méthode***

Relativement à ce qui a été dit précédemment, les méthodes de surclassement permettent de faire ressortir une ou plusieurs actions qui sont considérées comme étant les meilleures dans un scénario donné. Toutefois, pour déterminer la méthode à choisir, il est nécessaire de considérer des facteurs clefs comme le nombre de décideurs, les préférences naturelles des décideurs, le

type d'information qui sera traitée et l'hypothèse sur laquelle repose la méthode (Martel et Guiltoni, 1998) en plus des besoins et des contraintes que nous avons.

Dans le cadre de ce travail, puisqu'il n'y a qu'un seul décideur, le facteur le plus important est relatif au type d'information traitée. Puisque les données que nous recueillerons seront autant de nature quantitative que qualitative, il est impératif de sélectionner une méthode qui est en mesure de fonctionner avec ces deux types.

Il est aussi important de prendre en considération les besoins que nous avons. En effet, dans le cadre de ce travail, le résultat que nous désirons obtenir est un classement des différentes actions de la meilleure à la pire ou un classement absolu.

Au niveau des contraintes, il est aussi nécessaire de sélectionner une méthode simple et efficace qui n'est pas difficile à mettre en place. En effet, certaines méthodes d'analyse multicritère comportent des étapes complexes qui peuvent être un frein à leur mise en place. Pour cette raison, le choix d'une méthode relativement simple à utiliser est prioritaire.

En prenant en considération tous les facteurs décrits ci-dessus, la méthode d'analyse multicritère PROMETHEE semble être la plus adéquate pour effectuer un classement des meilleurs librairies et logiciels pour chacune des catégories.

### ***3.3.1.3 Critères de recherche et poids des critères***

Dans le cadre d'une analyse multicritère, la définition des critères est probablement un des aspects les plus importants de la démarche (Urli, 2013). En général, deux approches sont principalement utilisées pour déterminer un ensemble de critères : l'approche descendante et l'approche ascendante (Urli, 2013). L'approche descendante a pour objectif diviser l'objectif principal en sous-objectifs (ou attributs) dans un format hiérarchique jusqu'à obtenir des attributs mesurables de manière qualitative ou quantitative. Quant à la deuxième approche, son fonctionnement est l'inverse de la première, c'est-à-dire que la liste d'attributs sera élaborée en fonction des demandes spécifiques des parties prenantes que l'on appelle «conséquences élémentaires». Ces demandes seront ultimement traduites en critères de manière à être évaluées via l'analyse multicritère (Urli, 2013).

Les critères de recherche pourraient être définis comme étant un moyen de mesurer des facettes importantes d'un problème selon un décideur ou un acteur principal. Puisque toutes les actions sont évaluées selon les mêmes critères, il est donc possible de les comparer en fonction de la nature des critères. De plus, peu importe l'approche choisie, il est important que le nombre final de critères ne soit pas très élevé et que ces derniers ne soient pas redondants. En d'autres mots, chaque critère doit être différent de ses pairs et doit être cohérent avec le problème initial.

Pour continuer, un autre aspect important associé aux critères est leur poids. Lorsqu'il y a plusieurs décideurs dans une même situation, il est possible d'agréger les poids proposés par les différents acteurs, de déterminer les poids par consensus ou encore privilégier la pondération complète par un seul acteur. Dans le cadre de ce travail il n'y a qu'un seul décideur, les poids pourront être attribués de manière objective en fonction de la revue de littérature sur les critères les plus importants et ceux qui le sont moins. De plus, afin de donner un score à un critère, il est également possible d'avoir recours à des instruments de mesure, des sites internet ou encore des bases de données (Urli, 2013). Toutefois, il est important que la note attribuée à un critère respecte la nature de ce critère. Par exemple, si l'unité de mesure d'un critère est de nature quantitative, sa note doit en être de même nature. Les sections suivantes présenteront les critères retenus pour chacune des catégories essentielles de logiciels afin de procéder à leur évaluation.

### **Critères associés aux catégories de logiciels et aux convertisseurs**

Malgré le fait que les catégories soient très différentes les unes des autres, certains critères se retrouvent dans chacune des catégories. En effet, en lien avec le contexte de réalisation de ce travail portant sur la réalisation d'une méthodologie générique, un aspect très important est d'utiliser le plus possible des logiciels « open source ». Pour cette raison, le premier critère de notre analyse sera celui de l'accessibilité. Les logiciels de type ouvert sont gratuits et donc, par le fait même, plus facilement accessible par des étudiants ou des petits groupes de travail qui n'ont pas les moyens de se procurer des logiciels propriétaires dispendieux (Hippel, 2003). De plus, l'utilisation de logiciels ouverts permet de travailler avec des produits qui possèdent souvent une grande interopérabilité, c'est-à-dire qu'ils sont en mesure de fonctionner avec une multitude d'autres logiciels ou de normes. Finalement, puisque leur code est ouvert à tous, il est possible de faire une version du logiciel personnalisée qui est mieux adaptée aux besoins de l'utilisateur ou d'un projet (Serrano, 2006).

Pour continuer, un autre critère important est la pérennité. En effet, afin que la boîte à outils soit générique et réutilisable à long terme, il est nécessaire que les logiciels que nous utiliserons soient disponibles sur le long terme. Ce critère vise également à évaluer le niveau d'engagement vis-à-vis un logiciel donné. Une des façons reconnues d'évaluer la pérennité d'un logiciel ouvert est de regarder le nombre de personnes qui ont contribué à son développement. Plus le nombre de contributeurs est élevé, plus il y a de chance que des personnes continuent à entretenir ou à faire progresser le logiciel dans le futur (Onoue, 2016). C'est en s'appuyant sur ce type de données que nous effectuerons notre évaluation sur la pérennité des logiciels.

Enfin, le dernier critère est la popularité d'un logiciel. C'est un aspect à ne pas négliger parce que cela implique plusieurs points importants. Dans un premier temps, plus un logiciel est populaire, plus il y a de chance que sa communauté soit importante et donc qu'il y ait de l'information accrue avec de la documentation, des conseils d'experts ou même des blogues internet (Aggarwal, 2014). Deuxièmement, la popularité d'un logiciel se traduit aussi très souvent par sa facilité d'utilisation. En effet, les gens ont tendance à utiliser et à adopter les logiciels qui sont les plus simples à utiliser (Lee, 2009). De cette manière, il est possible de faire une corrélation entre la popularité d'un logiciel et sa facilité d'évaluation. Pour évaluer la popularité des logiciels, il est possible d'avoir recours au classement associé à chacun des logiciels sur leur site d'hébergement de code. Ceci est un moyen efficace et simple pour évaluer la popularité des logiciels auprès d'un site internet de grande envergure.

### **Critère spécifique aux API et serveurs cartographiques**

Il existe un critère qui s'applique seulement aux API et aux serveurs cartographiques : le respect des normes de l'OGC. En effet, ce dernier représente un aspect important à regarder lorsqu'il s'agit de sélectionner l'API et le serveur cartographique. Puisque l'OGC définit des standards ouverts qui sont reconnus et fortement utilisés dans le domaine de la cartographie web, une API ou un serveur cartographique qui en respecte plusieurs bénéficiera d'une grande interopérabilité notamment avec les nombreux systèmes déjà existants (Percivall, 2010). Cependant, il serait fastidieux et parfois complexe de vérifier la compatibilité de l'entièreté des standards de l'OGC puisqu'il en existe un très grand nombre réparti en plusieurs sous-catégories. Pour cette raison, nous nous concentrerons spécifiquement sur les six standards les plus importants selon (Pick, 2008) : le WMS (Web Map Service), le CAT (Catalogue Service), le GML (Geography Markup

Language), le WFS (Web Feature Service), la SF (simple Feature) et finalement le WCS (Web Coverage Service). Ainsi, afin d'évaluer ce critère, nous vérifierons combien de ces six standards les API et serveurs cartographiques sont en mesure de respecter.

### **Critère spécifique aux API cartographique et cadre de travail adaptatif**

Pour ces catégories, le poids du logiciel s'avère un critère qui peut faire une différence lorsque viendra le temps d'utiliser une ACM. En effet, celui-ci peut influencer le temps de chargement d'une page web de manière assez drastique (Buckler, 2014). Alors qu'en 2013 le volume total moyen d'une page web était de 1.7 Mo, le temps de chargement d'une telle page, aujourd'hui, sur un téléphone intelligent connecté à un réseau 3G est d'environ une minute (Buckler, 2014). Par conséquent, l'utilisation de logiciels légers dans le développement d'une ACM pourrait s'avérer un choix judicieux puisqu'ils augmenteraient la vitesse de chargement de l'application. De façon générale, si on combine le poids total d'une API cartographique et d'un cadre de travail adaptatif, on obtient un pourcentage non négligeable du poids total d'une page internet. (HTTP archive, 2015). Par exemple, l'utilisation du cadre de travail Bootstrap à lui seul ajoute 368 Ko à notre page web et 740 Ko pour OpenLayers3 pour un total de 1108 Ko, soit un peu plus de 1 Mo (Bootstrap, 2016) (OpenLayers3, 2016). En comparant cette taille au volume total moyen d'une page web de 1.7 Mo (1724 Ko), cela représente environ 64 % du poids total. Pour cette raison, le volume des cadres de travail et des API peut représenter un aspect important afin d'améliorer la performance d'une ACM.

### **Critère unique - Cadre de travail adaptatif**

L'évaluation des cadres de travail adaptatifs passe aussi par un autre critère : le support des navigateurs web. En effet, il existe sur le web une multitude de navigateurs qui peuvent être utilisés afin d'ouvrir une page internet. Pour cette raison, dans le but de créer une méthodologie qui puisse être la plus générique possible, il est nécessaire de faire en sorte que notre application puisse être fonctionnelle sur un maximum de navigateurs web. Malheureusement, ce ne sont pas tous les cadres de travail qui fonctionnent sur tous les navigateurs web. Dans ce contexte, ce critère a donc comme but d'attribuer une meilleure note à un cadre de travail qui s'utilise sur un grand nombre de navigateurs web. Les plus importants aujourd'hui sont Chrome, Firefox, Internet Explorer, et Safari (Statista, 2016). C'est donc basé sur ces derniers que nous évaluerons ce critère.

### **Critères associés aux convertisseurs**

Contrairement aux logiciels axés sur le volet web de notre méthodologie, les logiciels de conversion possèdent trois critères d'évaluation supplémentaires et uniques à cette catégorie. En effet, dans le cadre de nos objectifs secondaires, un aspect important est l'utilisation des périphériques des supports mobiles. Il est aussi important de vérifier si le logiciel de conversion permet de générer des applications qui fonctionnent autant sur les systèmes d'exploitation mobiles Android, Windows et Apple. Ces derniers, considérés comme étant les trois plus importants (Idc, 2015), sont essentiels au fonctionnement de la méthodologie.

Enfin, le dernier critère spécifique à cette catégorie est l'utilisation des technologies du web. En effet, il existe sur le marché beaucoup de convertisseurs qui utilisent des langages de programmation complexes les rendant pratiquement impossibles à utiliser pour des géomaticiens. Ce critère vise à différencier les convertisseurs qui permettent d'utiliser les langages HTML, CSS et JavaScript afin de transformer une application web en application mobile.

#### ***3.3.1.4 Bilan des critères d'évaluation***

Pour conclure cette section, le tableau 3 présente l'ensemble des indicateurs d'évaluation qui seront utilisés pour faire l'analyse multicritère classés par catégorie. Les trois critères en vert représentent les critères identiques aux trois catégories de logiciels essentiels. La colonne à l'extrême droite, soit celle représentant les logiciels de conversion, possède quant à elle des critères un peu différents excepté le critère « open source » comme mentionné dans la section précédente. La pondération attribuée aux critères sera discutée dans la prochaine section sur la méthodologie.

Tableau 3 : Indicateurs pour chaque catégorie essentiel de logiciels

Volet web de la méthodologie			Volet mobile
<p>API cartographique</p> <ul style="list-style-type: none"> <li>•C1: Open source</li> <li>•C2: Pérennité</li> <li>•C3: Popularité</li> <li>•C4: Standards OGC</li> <li>•C5: Poids</li> </ul>	<p>Cadre de travail adaptatif</p> <ul style="list-style-type: none"> <li>•C1: Open source</li> <li>•C2: Pérennité</li> <li>•C3: Popularité</li> <li>•C4: Support navigateur web</li> <li>•C5: Poids</li> </ul>	<p>Serveur cartographique</p> <ul style="list-style-type: none"> <li>•C1: Open source</li> <li>•C2: Pérennité</li> <li>•C3: Popularité</li> <li>•C4: Standards OGC</li> </ul>	<p>Logiciel de conversion</p> <ul style="list-style-type: none"> <li>•C1: Open source</li> <li>•C2: Pérennité</li> <li>•C3: popularité</li> <li>•C4: Technologies web</li> <li>•C5: Périphériques</li> <li>•C6: Support système d'exploitation</li> </ul>

### 3.3.2 Acquisition des données

L'acquisition des données, comme mentionnée précédemment, se fera principalement à partir du site internet GitHub. Les tableaux 4.1, 4.2, 4.3 et 4.4 présentent, pour chacun des logiciels et des critères, les données qui leur sont associées.

À titre informatif, le poids des API cartographiques et des cadres de travail adaptatif représente celui de leur version non minimisée. De plus, il est important de mentionné que c'est la version de base des serveurs cartographique qui est étudiée. Ainsi, alors que des extensions peuvent être ajoutées pour permettre de couvrir la majorité des standards de l'OGC, nous ne prenons pas en considération celles-ci dans notre analyse.

Tableau 4.1 : Données brutes pour les API cartographiques

Données		API cartographiques			
		Leaflet	Google Map API	OpenLayers3	Polymaps
Critères	Open Source (critère données ouverte)	oui	Non	Oui	Oui
	Popularité (critère nombre d'étoile)	13401	4099	1740	1354
	Pérennité (critère nombre de contributeur)	226	Maximum	142	5
	Interopérabilité (critère normes de l'OGC)	6 sur 6	6 sur 6	6 sur 6	0 sur 6
	Rapidité de chargement (critère poids en Kb)	33 Kb	0 Kb (lien fournit par Google)	4200 Kb	5200 Kb
	Utilisation de l'ACM (critère support navigateur web)	N/A	N/A	N/A	N/A
	Accessibilité (critère technologie du web)	N/A	N/A	N/A	N/A
	Plugins (critère périphériques mobiles)	N/A	N/A	N/A	N/A
	Utilisation de l'ACM (critère systèmes d'exploitation)	N/A	N/A	N/A	N/A

Tableau 4.2 : Données brutes pour les cadres de travail adaptatifs

Données		Cadres de travail adaptatifs					
		Bootstrap	Foundation 6	Skeleton	HTML5 Boilerplate	Pure	Semantic UI
Critères	Open Source (critère données ouverte)	Oui	Oui	Oui	Oui	Oui	Oui
	Popularité (critère nombre d'étoile)	94978	23122	11501	33611	13924	24766
	Pérennité (critère nombre de contributeur)	672	847	23	199	40	135
	Interopérabilité (critère normes de l'OGC)	N/A	N/A	N/A	N/A	N/A	N/A
	Rapidité de chargement (critère poids en Kb)	3900 Kb	2949 Kb	10.9 Kb	165 Kb	36.6 Kb	2800 Kb
	Utilisation de l'ACM (critère support navigateur web)	5 sur 5	5 sur 5	5 sur 5	5 sur 5	4 sur 5	4 sur 5
	Accessibilité (critère technologie du web)	N/A	N/A	N/A	N/A	N/A	N/A
	Plugins (critère périphériques mobiles)	N/A	N/A	N/A	N/A	N/A	N/A
	Utilisation de l'ACM (critère systèmes d'exploitation)	N/A	N/A	N/A	N/A	N/A	N/A



Tableau 4.3 : Données brutes pour les serveurs cartographiques

Données		Serveurs cartographiques		
		GeoServer	MapServer	Map Fish
Critères	Open Source (critère données ouverte)	Oui	Oui	Oui
	Popularité (critère nombre d'étoile)	509	257	5
	Pérennité (critère nombre de contributeur)	139	67	12
	Interopérabilité (critère normes de l'OGC)	3 sur 6	4 sur 6	4 sur 6
	Rapidité de chargement (critère poids en Kb)	N/A	N/A	N/A
	Utilisation de l'ACM (critère support navigateur web)	N/A	N/A	N/A
	Accessibilité (critère technologie du web)	N/A	N/A	N/A
	Plugins (critère périphériques mobiles)	N/A	N/A	N/A
	Utilisation de l'ACM (critère systèmes d'exploitation)	N/A	N/A	N/A

Tableau 4.4 : Données brutes pour les convertisseurs

Données		Convertisseurs			
		Appcelerator titanium	PhoneGap Build	App Deck	Cordova
Critères	Open Source (critère données ouverte)	Oui	Oui	Oui	Oui
	Popularité (critère nombre d'étoile)	2066	4182	19	2479
	Pérennité (critère nombre de contributeur)	159	19	3	288
	Interopérabilité (critère normes de l'OGC)	N/A	N/A	N/A	N/A
	Rapidité de chargement (critère poids en Ko)	N/A	N/A	N/A	N/A
	Utilisation de l'ACM (critère support navigateur web)	N/A	N/A	N/A	N/A
	Accessibilité (critère technologie du web)	Oui	Oui	Oui	Oui
	Plugins (critère périphériques mobiles)	Oui	Oui	Oui	Oui
	Utilisation de l'ACM (critère systèmes d'exploitation)	2 sur 3 (Android et iOS)	3 sur 3	2 sur 3	3 sur 3

### **3.3.3 Formatage des données**

Dans certains cas, il est nécessaire d'effectuer un formatage des données brutes afin de pouvoir les utiliser dans l'analyse multicritère. Par exemple, les critères suivants : interopérabilité et utilisation de l'ACM (sur ordinateur et mobile), ont des données brutes sous la forme d'une fraction. Nous avons donc transformé ces fractions en pourcentage.

De plus, dans les tableaux 4, il est possible d'observer que le critère pérennité pour l'API cartographique Google Map a obtenu la notation de « maximum ». Cela est le fait de l'impossibilité d'évaluer le nombre de contributeurs de cette API puisqu'elle est propriétaire. Toutefois, compte tenu du fait que Google est une compagnie internationale de très grande envergure, la notation de maximum a été attribuée à cette API puisque nous jugeons qu'elle sera certainement entretenue et mise à jour dans le futur. Le convertisseur PhoneGap a reçu la même note que le convertisseur Cordova puisqu'il est basé sur ce dernier. En effet, la pérennité de PhoneGap est intimement liée à celle de Cordova.

### **3.3.4 Pondération des critères**

La pondération des critères a été effectuée en deux grandes étapes. Premièrement, l'ensemble des critères qui sont identiques à chacune des catégories a été regroupé dans une seule section que nous appelons le bloc évaluatif de base. Ce bloc est composé des critères les plus importants de notre analyse multicritère et a obtenu une pondération globale de 75%. Ce pourcentage a été scindé en trois sous-pourcentages pour chaque critère. Or, dans le cadre de cette étude, le critère jugé le plus important dans le bloc évaluatif de base est celui sur les données ouvertes. Pour cette raison, ce dernier a reçu une pondération de 30 %. Les deux autres critères ont reçu une pondération égale soit de 22,5 %.

Les autres critères ont obtenu une pondération égale. Par exemple, pour la catégorie des API cartographiques, les critères normes de l'OGC et poids ont été pondérés à 12,5 % chacun.

Pour la catégorie des convertisseurs, l'attribution des poids est différente. En effet, puisque les critères ne faisant pas partie du bloc évaluatif de base sont également très importants afin de sélectionner le bon logiciel, leur pondération globale a été augmentée. Pour ce faire, la pondération du critère des données ouvertes dans le bloc évaluatif de base a été réduite à 22,5 %. De cette manière, les critères utilisations des technologies du web et utilisation des périphériques

ont obtenu une pondération de 12,08 % et le critère de support des systèmes d'exploitation mobile a obtenu une pondération de 8,33 %. Aucune méthode particulière n'a été utilisée pour effectuer la pondération des critères. En effet, l'un des avantages des méthodes d'analyse multicritère de type surclassement est que les poids peuvent être attribués de manière objective par le décideur. Cette étape a donc été faite en fonction de la revue de littérature qui a été effectuée sur le sujet. Pour résumer, le tableau 5 présente, pour chaque catégorie, la pondération des critères qui a été utilisée pour effectuer l'analyse multicritère.

Tableau 5 : Pondération des critères pour chacune des catégories

	Catégories			
Critères	API cartographiques	Cadre de travail adaptatif	Serveur cartographique	Logiciel de conversion
Technologies ouvertes	30 %	30 %	30 %	22.5 %
Popularité	22.5 %	22.5 %	22.5 %	22.5 %
Pérennité	22.5 %	22.5 %	22.5 %	22.5 %
Normes de l'OGC	12.5 %	N/A	25 %	N/A
Poids	12.5 %	12.5 %	N/A	N/A
Support navigateur web	N/A	12.5 %	N/A	N/A
Technologies du web	N/A	N/A	N/A	12.08 %
Périphériques	N/A	N/A	N/A	12.08 %
Support systèmes d'exploitation mobile	N/A	N/A	N/A	8.33 %

### 3.3.5 Réalisation de l'analyse multicritère

Pour effectuer l'analyse multicritère, nous avons eu recours à un logiciel d'un tiers parti nommé Visual Prométhée. Ce logiciel est disponible gratuitement sur le site internet [www.promethee-](http://www.promethee-)

gaia.net et permet de mettre en œuvre facilement la méthode PROMETHEE de façon automatisée. De plus, ce logiciel propose une variété de graphiques afin d'étudier et de comprendre les scénarios. Ce logiciel a aussi été sélectionné parce qu'il est le seul à avoir été développé sous la supervision des auteurs de la méthode PROMETHEE et Gaia. La version utilisée du logiciel est la version académique (version 1.4).

Fichier Edition Modèle Contrôle PROMETHEE-GAIA GDSS SIG Custom Assistants Snapshots Options Aide

Scénario1	Open source	Popularité	Pérennité	Normes OGC	Poids	Support web	Techno web	Périphériques	Support mobile
Unité	o/n	Nombre	Nombre	5 points	unit	5 points	unit	o/n	Pourcentage
Cluster/Groupe	◇	◇	◇	◇	◇	◇	◇	◇	◇
<b>Préférences</b>									
Min/Max	max	max	max	max	min	max	max	max	max
Poids	22,50	22,50	22,50	25,00	13,00	0,00	12,08	12,08	8,33
Fn. de préférence	A paliers	Forme en V	Forme en V	A paliers	Linéaire	A paliers	Usuel	Usuel	Usuel
Seuils	absolu	absolu	absolu	absolu	absolu	absolu	absolu	absolu	absolu
- Q: Indifférence	1	n/d	n/d	1	1,0	1,00	n/d	n/d	n/d
- P: Préférence	1	11140	204	2	2,0	2,00	n/d	n/d	n/d
- S: Gaussien	n/d	n/d	n/d	n/d	n/d	n/d	n/d	n/d	n/d
<b>Statistiques</b>									
<b>Evaluations</b>									
<input checked="" type="checkbox"/>	Leaflet	oui	13401	226	très bon	33,0	n/d	n/d	n/d
<input checked="" type="checkbox"/>	Google Maps	non	4099	226	très bon	0,0	n/d	n/d	n/d
<input checked="" type="checkbox"/>	OpenLayers3	oui	1740	142	très bon	4200,0	n/d	n/d	n/d
<input checked="" type="checkbox"/>	Polymaps	oui	1354	5	très mauvais	5200,0	n/d	n/d	n/d
<input checked="" type="checkbox"/>	Bootstrap	oui	94978	672	n/d	3900,0	très bon	n/d	n/d
<input checked="" type="checkbox"/>	Foundation 6	oui	23122	847	n/d	2949,0	très bon	n/d	n/d
<input checked="" type="checkbox"/>	Skeleton	oui	11501	23	n/d	10,9	très bon	n/d	n/d
<input checked="" type="checkbox"/>	HTML5 Boilerplate	oui	33611	199	n/d	165,0	très bon	n/d	n/d
<input checked="" type="checkbox"/>	Pure	oui	13924	40	n/d	36,6	bon	n/d	n/d
<input checked="" type="checkbox"/>	Semantic UI	oui	24766	135	n/d	2800,0	bon	n/d	n/d
<input checked="" type="checkbox"/>	Geoserver	oui	509	139	mauvais	n/d	n/d	n/d	n/d
<input checked="" type="checkbox"/>	MapServer	oui	257	67	moyen	n/d	n/d	n/d	n/d
<input checked="" type="checkbox"/>	Map Fish	oui	5	12	moyen	n/d	n/d	n/d	n/d
<input checked="" type="checkbox"/>	Appcelerator Tit...	oui	2066	159	n/d	n/d	n/d	oui	oui
									66,66

Tous Scénario1

Figure 8 : Capture d'écran du logiciel Visual Promethee montrant la grille d'analyse que nous avons créée pour effectuer l'analyse multicritère provenant du logiciel Visual Prométhée

### 3.3.5.1 Configuration de base des scénarios

Pour débiter une analyse multicritère avec le logiciel Visual Promethee, la première étape consiste à créer un nouveau scénario. Le logiciel nous demande de spécifier le nombre de critères et d'actions pour le scénario en question. Pour nous aider, le logiciel nous permet de déterminer

des groupes d'actions et de critères que nous pouvons sélectionner ou désélectionner selon nos préférences. Tous les scénarios de notre projet ont été inclus dans une seule page afin de simplifier la visualisation des actions et des critères par rapport aux tableaux présentés précédemment dans ce travail. Pour cette étape, nous avons donc créé une feuille avec dix-sept actions (représentant chaque logiciel et librairie) et neuf critères représentés sur la figure 9. Chaque action et critère correspondant à la même catégorie ont été associés à une couleur différente pour mieux les identifier dans la page d'analyse.

### ***3.3.5.2 Transcription des données formatées et des poids des critères dans le logiciel***

Une fois la configuration de base effectuée, il faut effectuer la pondération des différents critères et les données formatées pour les actions dans la page d'analyse. À cette étape, nous ne nous occupons pas des préférences avancées telles que la fonction de préférence, les seuils, le degré d'indifférence et de préférence. Seule la cardinalité des critères, c'est-à-dire si la meilleure valeur d'un critère est la plus élevée ou la plus faible, et le poids ont été saisis dans la page d'analyse.

### ***3.3.5.3 Insertion des préférences avancées***

À cette étape, nous saisissons les préférences avancées dans le modèle. Celles-ci comprennent les fonctions de préférence, les seuils, le degré d'indifférence et de préférence. Afin de faciliter le processus, le logiciel effectue automatiquement les calculs nécessaires afin de déterminer les différents seuils et nous propose les choix idéaux concernant les fonctions de préférence. Toutefois, en tout temps, il est possible de personnaliser le scénario. Cependant, le logiciel optimise la bonne option à prendre en compte afin de maximiser le critère en question. Pour cette raison, nous avons toujours utilisé les propositions du logiciel.

## **4. Résultats**

Pour chaque catégorie, les bons critères et actions ont été sélectionnés dans le logiciel Visual Prométhée. Ainsi, les sections suivantes présentent le résultat de l'analyse multicritère. L'appellation « PHI+ », représenté par un nombre positif ou négatif, désigne le niveau de surclassement d'une action par rapport aux autres. Ainsi, plus le PHI+ est élevé, plus cette option est à privilégier sur les autres (Figueria *et al.*, 2015).

#### 4.1 Catégorie API cartographique

Le graphique produit par le logiciel Visual Promethee résultant de l'analyse Promethee II montre très bien que l'API cartographique Leaflet se démarque des autres. En effet, alors que OpenLayers3 et Google Map API sont relativement proches l'un de l'autre, l'API Polymaps termine dernière dans le classement. D'un point de vue de notation, Leaflet a obtenu un PHI+ de 0,4981 alors que son plus proche rival, OpenLayers3, n'a obtenu qu'un PHI+ de 0,0025. De ce fait, ces résultats nous indiquent que l'API Leaflet s'est distancée des autres API lors de cette évaluation. Ce résultat peut-être attribuable en partie aux critères de popularité et de pérennité qui possèdent tous les deux un poids important dans l'analyse. L'API Leaflet se distingue notamment sur la popularité avec une note presque 300% plus élevé que celui de OpenLayers3. Le classement final pour cette catégorie est donc le suivant :

1. Leaflet (PHI+ = 0,4981)
2. OpenLayers3 (PHI+ = 0,0025)
3. Google Map API (PHI+ = -0,0508)
4. Polymaps (PHI+ = -0,4499)

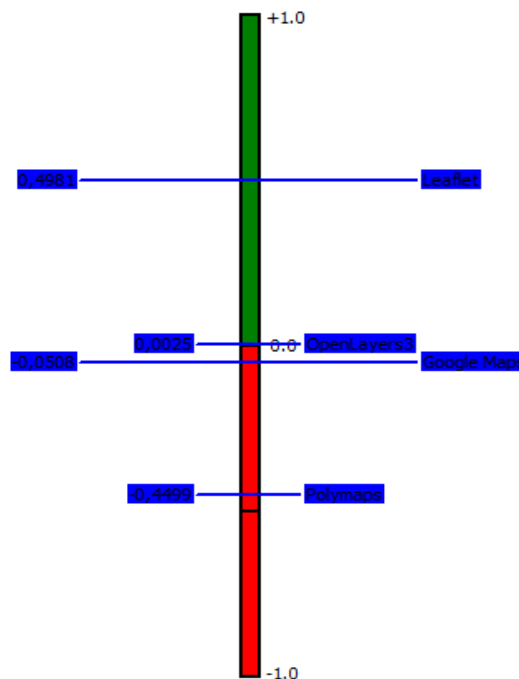


Figure 9 : Résultat de l'analyse PROMETHEE pour la catégorie API cartographique provenant du logiciel Visual PROMETHEE

## 4.2 Catégorie cadre de travail adaptatif

L'analyse multicritère a révélé que Bootstrap est celui qui a obtenu la meilleure note finale. Toutefois, comparativement aux API cartographiques, les notes obtenues par les différentes solutions sont beaucoup plus rapprochées les unes des autres. Cela signifie donc que la différence entre les cadres de travail est moindre en fonction des critères définis. Le classement final pour cette catégorie est le suivant :

1. **Bootstrap (PHI+ = 0,2624)**
2. HTML5 Boilerplate (PHI+ = 0,1660)
3. Foundation 6 (PHI+ = 0,1427)
4. Semantic UI (PHI+ = -0,0791)
5. Skeleton (PHI+ = -0,2406)
6. Pure (PHI+ = -0,2514)

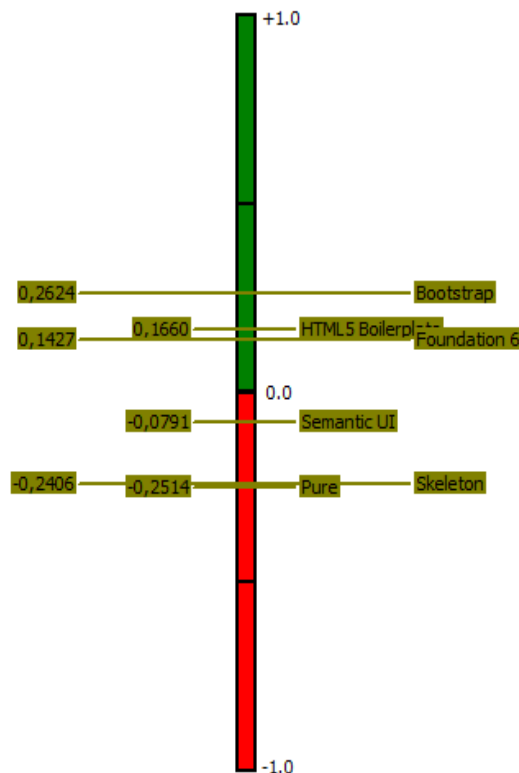


Figure 10 : Résultat de l'analyse PROMETHEE pour la catégorie cadre de travail adaptatif provenant du logiciel Visual PROMETHEE

### 4.3 Catégorie serveur cartographique

L'analyse multicritère a permis d'identifier que le serveur cartographique ayant obtenu la meilleure note est Geoserver. Avec un PHI+ de 0,1174, il est le seul à avoir obtenu un résultat supérieur à 0. Tout comme les API cartographiques, ce résultat peut être lié aux fortes notes du serveur cartographique Geoserver aux critères popularité et pérennité. Malgré le fait qu'individuellement ces deux critères possèdent un poids moins élevé que le respect des normes de l'OGC, leur poids combiné ainsi que leur si forte note a permis à Geoserver de prendre la première place. La liste des résultats est donc la suivante :

1. Geoserver (PHI+ = 0,1174)
2. MapServer (PHI+ = -0,0094)
3. Map Fish (PHI+ = -0,1080)

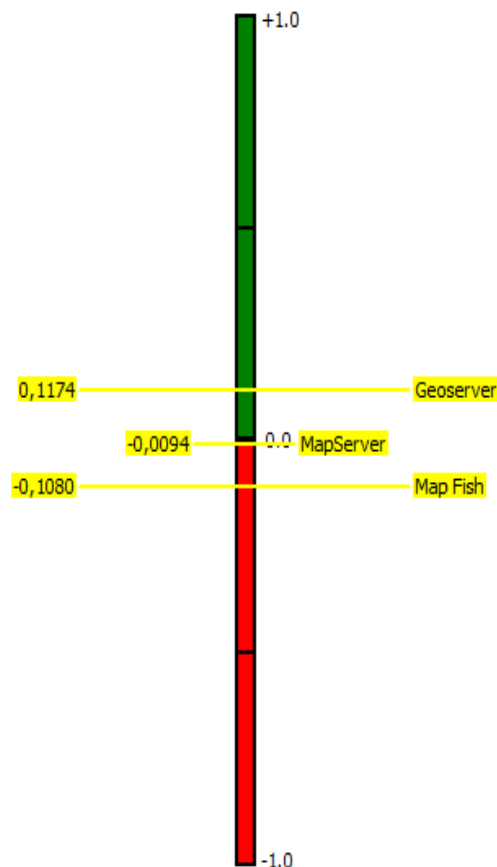


Figure 11 : Résultat de l'analyse PROMETHEE pour la catégorie serveur cartographique provenant du logiciel Visual PROMETHEE



#### 4.4 Catégorie convertisseurs

Finalement, pour cette dernière catégorie, notre analyse a permis d'identifier le logiciel PhoneGap Build comme étant celui ayant obtenu le meilleur résultat. Toutefois, la différence avec son plus proche rival (Cordova) est assez petite puisqu'il n'y a qu'environ 0,5 point qui les sépare. Le classement final pour cette catégorie est le suivant :

1. PhoneGap Build (PHI+ = 0,2317)
2. Cordova (PHI+ = 0,1859)
3. Appcelerator Titanium (PHI+ = -0,0963)
4. App Deck (PHI+ = -0,3213)

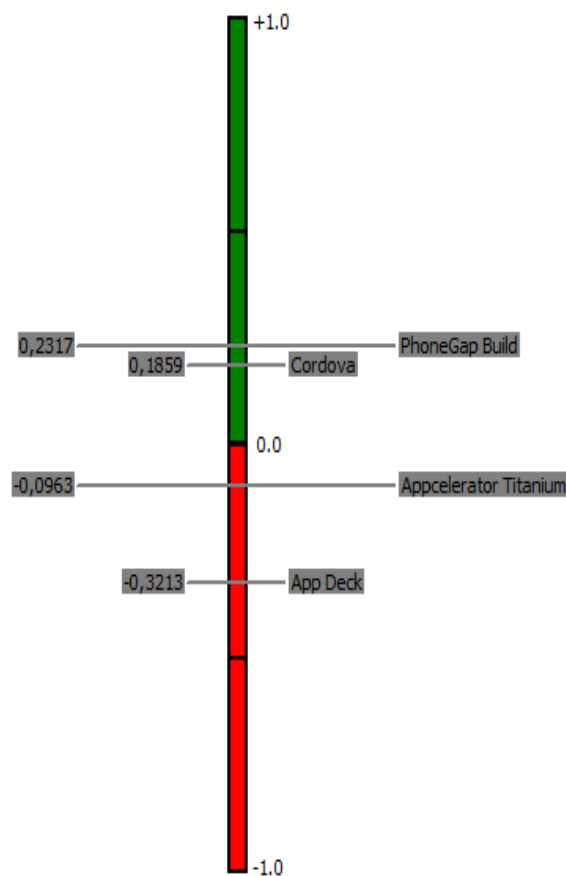


Figure 13 : Résultat de l'analyse PROMETHEE pour la catégorie des convertisseurs provenant du logiciel Visual PROMETHEE

#### 4.5 Bilan des résultats et obtention de la boîte à outils

Pour résumer, l'analyse multicritère de type PROMETHEE II nous a permis d'identifier les logiciels, parmi chaque catégorie, qui ont obtenu la meilleure note en fonction des critères que nous avons définis. Ainsi, en prenant en compte le gagnant pour chacune des quatre analyses que nous avons effectuées, nous sommes en mesure de constituer notre boîte à outils et, par le fait même, de déterminer les logiciels et librairies qui feront partie de notre méthodologie.

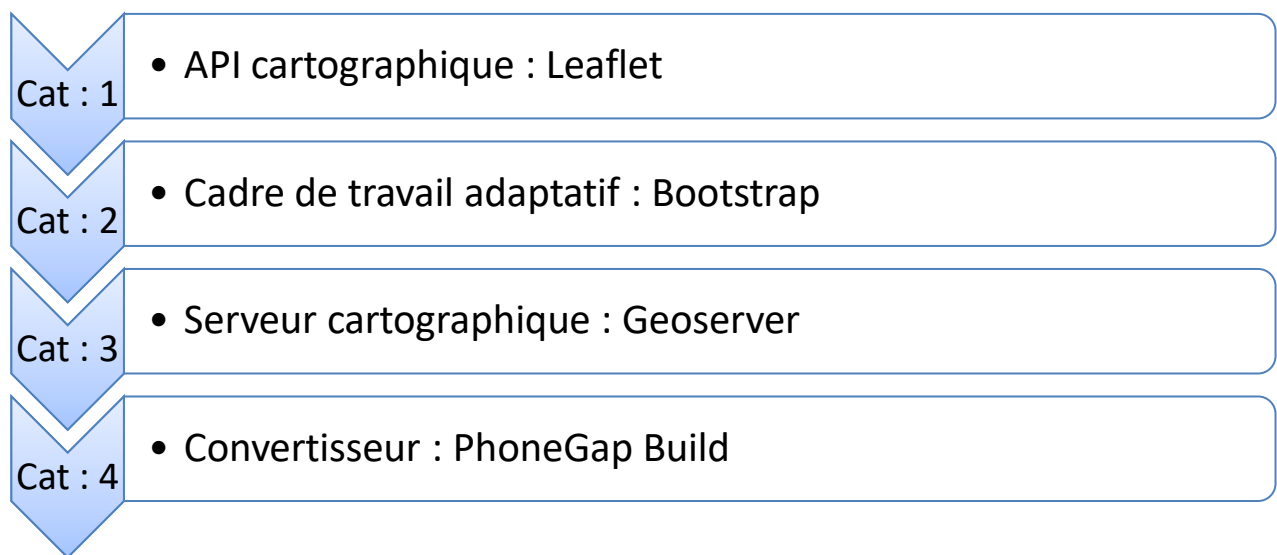


Figure 12 : Résultat de l'analyse multicritère pour chacune des catégories

#### 4.6 Analyse des résultats

Grâce à une revue de la littérature et d'une analyse multicritère avancée, nous avons été en mesure de compléter notre méthodologie de création d'une ACM. Toutefois, afin de véritablement être en mesure de vérifier la pertinence de cette méthodologie pour les différents sous-objectifs de ce travail, nous devons développer notre propre ACM.

### 4.6.1 Démarche de création de l'application test

Pour la création du prototype test, quelques étapes ont été nécessaires. Elles seront simplement énumérées et rapidement décrites. Le code de l'application a été joint en annexe afin d'être consulté indépendamment de ce texte, mais aussi afin de pouvoir être réutilisé comme application de base par d'autres étudiants du Département de géomatique de l'Université de Sherbrooke.

La première étape a été de créer l'aspect général de l'application cartographique web. Pour ce faire, nous avons utilisé le cadre de travail adaptatif Bootstrap qui nous permet de créer une barre de navigation en haut de la page en plus d'un menu à la gauche de l'écran. Un aspect intéressant du cadre de travail qui a été employé est le système de grille qu'il propose qui nous a permis d'attribuer au menu de gauche un quart de la largeur de la page. L'espace cartographique occupe donc trois quarts de la page web en termes de largeur. La figure 13 montre, sur un ordinateur, le résultat obtenu jusqu'à présent. Les trois sections ont été encadrées en rouge afin de les différencier plus facilement les unes des autres. La section #1 représente le menu de navigation, la section #2 l'espace cartographique et la section #3 la barre de navigation.

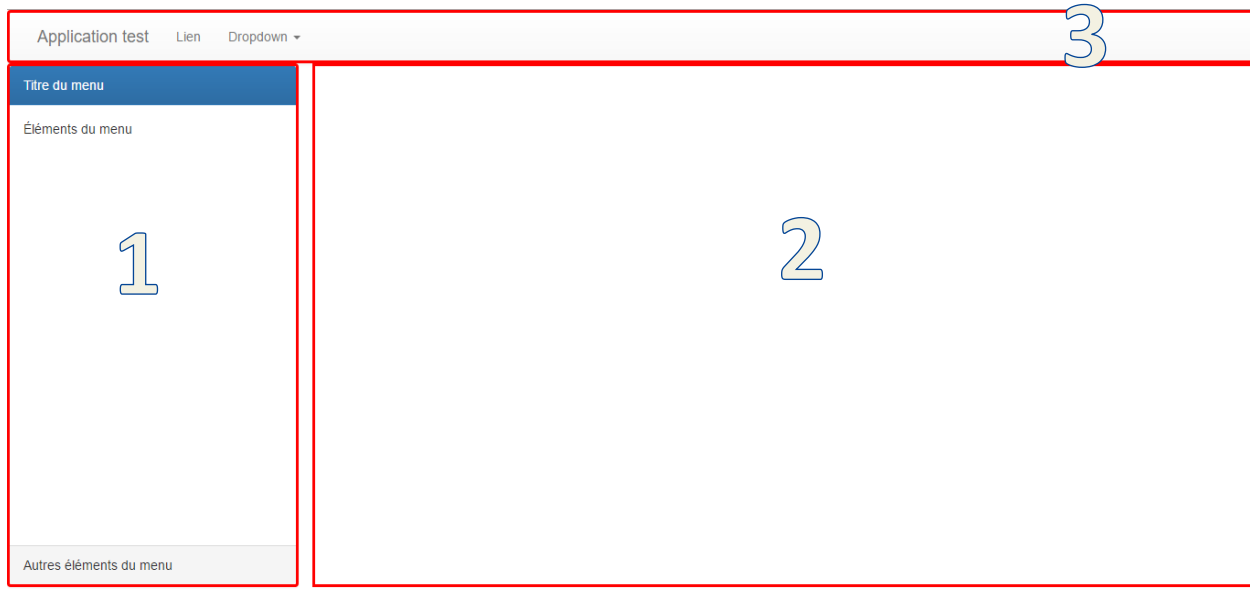


Figure 13 : Présentation des différentes sections de l'ACM test que nous avons développé

Ensuite, alors que les sections principales de notre application ont été définies, nous avons été capables d'insérer une carte à l'aide de l'API cartographique Leaflet dans la section #2. Pour y

arriver, il a été nécessaire d'ajouter la librairie JavaScript de leaflet à notre dossier librairies et, dans le fichier *index.html*, d'y insérer les liens faisant référence au style et au code JavaScript de l'API cartographique. De plus, la portion de code ci-dessous a été rajoutée afin de positionner la carte au bon endroit au moment du chargement de la page web et de définir le niveau de zoom désiré. Finalement, nous avons ajouté la couche de fond d'OpenStreetMap.

```
<script>

    var map = L.map('map').setView([-41.2858, 174.78682], 14);

    mapLink = '<a href="http://openstreetmap.org">OpenStreetMap</a>';
    L.tileLayer('http://{s}.tile.openstreetmap.org/{z}/{x}/{y}.png', {
        attribution: '&copy; ' + mapLink + ' Contributors',
        maxZoom: 18,
    }).addTo(map);

</script>
```

Figure 14 : Code JavaScript d'initialisation de la carte web tiré d'un éditeur de texte

Pour continuer, à l'aide du serveur cartographique GeoServer, nous avons ajouté à notre carte des couches géographiques représentant les sentiers récréatifs du Québec. Comme mentionné dans la section sur les données, celle-ci proviennent de Mme Simone Moretti qui avait préalablement mis au point un espace de stockage de ses données géographiques avec GeoServer. De ce fait, nous avons simplement utilisé le lien faisant référence à son serveur cartographique afin d'ajouter ses couches à notre carte. Toutefois, il est important de mentionner que Mme Moretti utilisait déjà le serveur cartographique GeoServer dans le cadre de son travail de recherche. Dans le cas où notre analyse multicritère aurait pointé vers un autre logiciel, nous aurions demandé à Mme Moretti de nous fournir une copie de ses données brutes dans le but de nous-mêmes les entreposer dans le logiciel en question.

Finalement, la dernière étape a été de transformer notre application cartographique en application native. Pour ce faire, le logiciel web PhoneGap Build fût utilisé. En effet, la première étape est d'ajouter un fichier de configuration (.XML) dans le répertoire de base de notre projet d'ACM.

Ce dernier permet de définir les différents « plug-ins » que nous souhaitons utiliser avec notre application et toute autre information sur celle-ci comme la version, le nom ainsi que différentes permissions comme la possibilité d'utiliser une orientation paysage et portrait avec notre application. Afin de nous simplifier la vie, nous avons simplement utilisé un fichier de configuration open source déjà existant qu'il est possible de trouver sur le site internet GitHub au répertoire : *com.phonegap.helloworld*. L'avantage d'utiliser ce fichier de configuration est qu'il contient déjà une très grande partie des plug-ins qui nous intéressent tels que la géolocalisation et l'appareil photo. Cela nous a permis d'économiser beaucoup de temps en plus de pouvoir servir de base comme fichier de configuration pour toutes ACM confondues. De surcroît, au besoin du développeur de l'application cartographique, certaines lignes inutiles peuvent être simplement effacées du fichier de configuration ou laissées telles quelles. Avec une taille négligeable, ces dernières, n'influencent pas vraiment le poids final de l'application.

Alors que le fichier .XML a bien été configuré, nous avons pu écrire les quelques lignes de code en JavaScript permettant de faire le lien entre les plug-ins et l'application. Plus précisément, les lignes que nous avons écrites permettent d'obtenir, en appuyant sur un bouton, les coordonnées géographiques (longitude et latitude) de l'endroit où nous nous trouvons.

Finalement, la dernière étape est de se rendre sur le site internet de PhoneGap Build. Après s'être créé un nouveau compte, l'interface de PhoneGap Build nous permet de téléverser un fichier compressé (.zip) contenant l'ensemble des dossiers nécessaires au fonctionnement de notre application. Avant d'effectuer la conversion de notre application vers une application mobile, une option s'intitulant « hydratation » s'offre à nous. Celle-ci permet à notre application, une fois installée sur un support mobile, de vérifier si une nouvelle version est disponible. En effet, le code que l'on dépose sur PhoneGap Build peut être mis à jour à tout moment. De ce fait, si l'option « hydratation » est cochée, l'utilisateur n'a pas besoin de mettre à jour lui-même son application mobile en téléchargeant la nouvelle version. Dans notre cas, nous avons activé cette option, mais elle reste non-obligatoire. Au final, le logiciel PhoneGap Build a converti l'ensemble de notre fichier compressé en trois fichiers différents tels qu'un fichier .apk pour Android. Ceux-ci correspondent aux fichiers qui peuvent être lus par les différents systèmes d'exploitation mobiles. Ainsi, nous avons donc téléchargé ces fichiers et les avons transférés sur des appareils mobiles afin d'effectuer nos différents tests.

#### 4.6.1.1 Repositionnement des éléments selon la taille de l'écran

En ouvrant l'application test à la fois sur un ordinateur et sur un appareil mobile, nous avons pu constater que les éléments de la page web se positionnent différemment sur les deux plateformes. En effet, lorsque l'on regarde la barre de navigation de l'application située dans le haut de la page, on observe que sur un ordinateur (figure 14), celle-ci possède toute sa grandeur alors que sur un support mobile (figure 16), la barre de navigation se rétracte pour laisser la page à un menu plus adapté aux plus petits écrans. Aussi, on remarque également que les autres éléments de la page web tels que le menu et la carte se positionnent les uns en dessous des autres sur un écran de petite taille. Les chiffres sur l'image 16 indiquent les mêmes sections sur la figure 13 afin de pouvoir comparer l'emplacement des éléments facilement en fonction des différents supports.

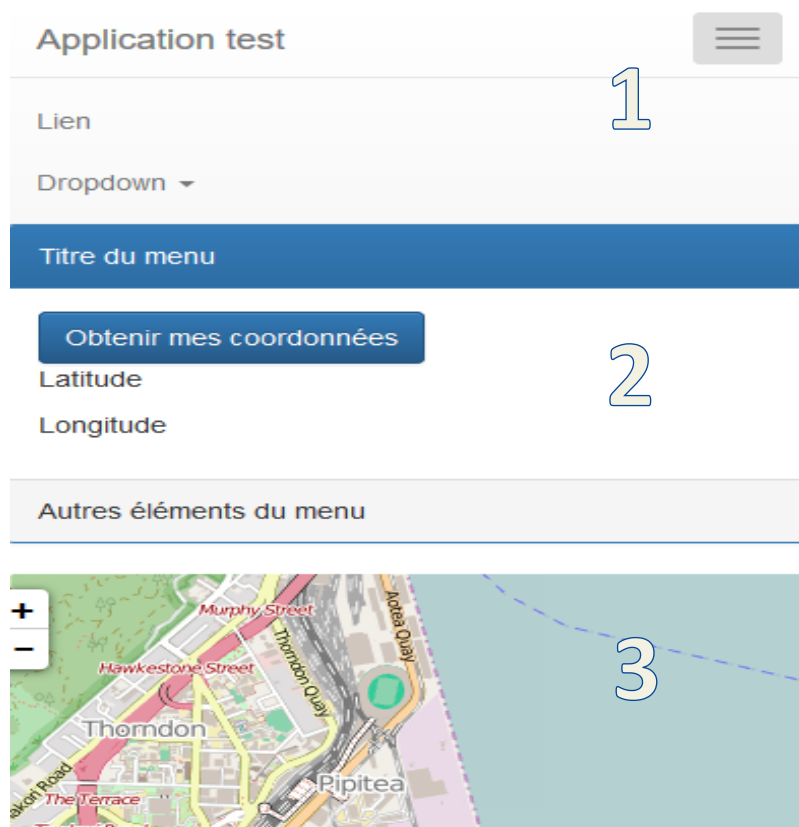


Figure 15 : Repositionnement des éléments de l'ACM test sur un téléphone intelligent

#### 4.6.1.2 Affichage d'une couche WMS

En regardant la carte de l'application web, on s'aperçoit que dans la région de Mascouche, il y a plusieurs lignes rouges qui se superposent à la couche de base d'OpenStreetMap. En effet, ces lignes représentent la couche de type WMS provenant du serveur cartographique de Mme Moretti. Ainsi, cela démontre que l'API cartographique que nous avons utilisée pour développer l'application test est en mesure d'afficher des couches WMS qui proviennent d'un serveur cartographique.

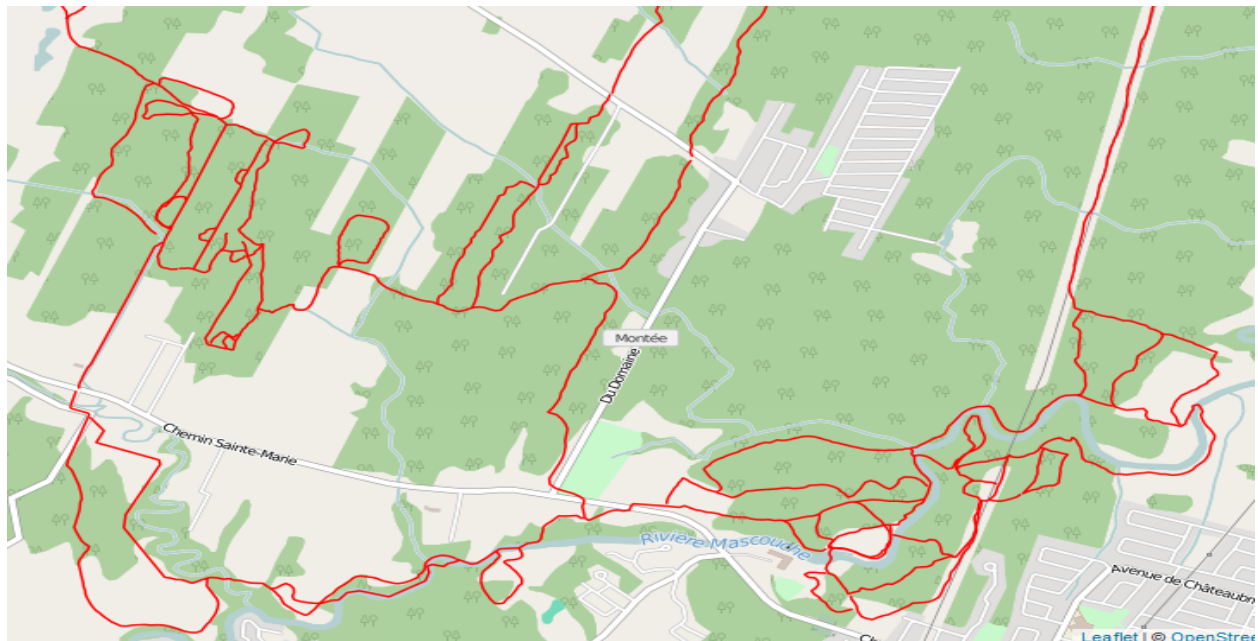


Figure 16 : Affichage d'une couche de type WMS sur l'ACM test que nous avons mise au point

#### 4.6.1.3 Affichage sur plusieurs plateformes

Comme mentionné précédemment, le logiciel PhoneGap Build nous a permis de convertir notre application test en application native. Nous avons été capables d'ouvrir l'application sur des téléphones intelligents de type Android et Windows en cliquant sur l'icône de l'application. Rapidement, celle-ci s'est exécutée sans aucune erreur dans les deux cas.

#### 4.6.1.4 Utilisation du GPS via un téléphone intelligent

En cliquant sur le bouton bleu dans le menu, on obtient immédiatement les coordonnées GPS de la position actuelle de l'appareil mobile. Pour tester cette fonctionnalité, nous nous sommes rendus à deux endroits différents à Montréal afin de vérifier la validité des coordonnées. Dans un

premier temps, nous sommes allés au centre du Parc Jarry et nous avons obtenu les coordonnées 45.535132, -73.628869. En entrant ces dernières dans l'application Google Map, on s'aperçoit en effet que ces coordonnées pointent vers le milieu du Parc Jarry. Deuxièmement, nous sommes allés refaire le même test au Parc Lafontaine au bout de la rue Calixa-Lavallée (près de l'Art Neuf) avec les coordonnées 45.526823, -73.569872. Le même résultat a été obtenu lors de la saisie de ces coordonnées dans Google Map.

## **5. Interprétation et discussion des résultats**

Avec les différents résultats obtenus, nous pouvons dire que l'objectif principal et les objectifs secondaires de ce travail d'étude ont été atteints avec succès. En effet, avec la boîte à outils que nous avons proposée, il a été possible de créer une ACM test répondant à l'ensemble des objectifs que nous nous étions fixés. En premier lieu, l'objectif principal visant à développer une méthodologie générique pour la création d'ACM a été atteint parce que nous avons été en mesure de créer une boîte à outils en fonction de plusieurs critères représentant la notion de « méthodologie générique ». Parmi ces critères, nous avons notamment évalué, pour chacun des logiciels et librairies recensés dans notre revue de littérature, leur niveau de pérennité et de popularité. La méthode d'analyse multicritère PROMETHEE nous a offert un classement des meilleures actions (logiciels et librairies) pour chaque catégorie essentielle au développement d'une ACM nous ayant permis, par la suite, de créer notre propre ACM test avec succès.

Pour continuer, le repositionnement des éléments visuels de notre application en fonction de différentes tailles d'écrans nous indique une utilisation possible sur plusieurs supports comme des ordinateurs, mais aussi des tablettes et des téléphones sans sacrifier son ergonomie initiale. De plus, cela nous indique aussi que le cadre de travail adaptatif que nous avons utilisé fonctionne et qu'il permet de développer une application unique et adaptative.

Autre point important, nous avons réussi à faire fonctionner l'application test sur les systèmes d'exploitation mobile Android et Windows sans réglage supplémentaire. Ceci démontre la versatilité du logiciel de conversion employé. Les ACM développées à partir de notre boîte à outils sont en mesure de rejoindre plus facilement un nombre élevé de personnes. Pour cette raison, d'un point de vue stratégique, c'est un avantage non négligeable pour les OBNL ou tout



autres groupes/personnes qui souhaitent augmenter la qualité de leurs services ou encore, plus simplement de créer une ACM consultable sur plusieurs systèmes d'exploitation mobile.

Pour conclure, un dernier point intéressant est la possibilité d'utiliser le GPS à partir d'un téléphone intelligent. En effet, cela ouvra la porte à une multitude d'applications qui pourraient utiliser cette fonctionnalité intégrée à la majorité des supports mobiles. Par ailleurs, le logiciel de conversion PhoneGap Build utilisé pour créer notre application test permet aussi d'intégrer à une ACM d'autres extensions telles que l'appareil photo, la caméra, l'utilisation de la liste de contacts et plus encore. Ainsi, la variété des ACM qui peuvent être développées à partir de ces derniers est extrêmement grande en plus de ne pas restreindre le développeur seulement qu'à une ou deux extensions.

### **5.1 Points forts et points faibles de la boîte à outils**

L'un des points forts de la boîte à outils est sa facilité d'utilisation. En effet, la simplicité et la rapidité avec laquelle nous avons réussi à mettre sur pied l'ACM qui nous a permis de tester la boîte à outils furent idéales. Malgré le fait que nous avons utilisé pour la première fois cette combinaison de logiciels et de bibliothèques pour développer des ACM, le développement de notre application a été complété très rapidement. Un autre point fort de la boîte à outils est son interopérabilité. En effet, l'analyse multicritère que nous avons effectuée a entre autres évalué l'interopérabilité des bibliothèques et logiciels qui ont été décelés lors de la revue de littérature. Pour cette raison, il pourrait être facile d'ajouter une autre bibliothèque de cartographie web à la boîte à outils sans compromettre son fonctionnement. Cela est particulièrement intéressant dans la mesure où certains développeurs d'ACM ou étudiants en géomatique souhaiteraient travailler avec un autre logiciel. Enfin, un autre point fort est l'utilisation du logiciel de conversion de l'application web en application native avec PhoneGap Build. En effet, alors que d'autres logiciels offrant des fonctionnalités similaires demandent un plus grand travail de codage, PhoneGap Build rend possible un transfert sur les plateformes Android, Windows mobile et iOS d'une application web sans avoir à programmer ne serait-ce qu'une seule ligne de code. Évidemment, l'ajout d'extensions dans l'ACM demande plus de travail, mais, pour beaucoup de personnes qui désirent simplement bénéficier des avantages d'une application native, cela est avantageux

À l'inverse, la boîte à outils comprend aussi son lot de points faibles. Alors que son but est de faciliter au maximum le processus de création d'une ACM, le développeur utilisant la boîte à outils doit, dans une certaine mesure, posséder des connaissances dans les langages informatiques web HTML, JavaScript et CSS3. En effet, le développement d'une ACM requiert tout de même l'utilisation de ces langages afin d'initialiser, par exemple, l'API cartographique. Pour continuer, alors que le logiciel PhoneGap Build est particulièrement intéressant dans le développement d'une ACM fonctionnant sur un support mobile, la librairie des extensions qu'il propose fonctionne en grande partie seulement sur un système d'exploitation à la fois. Même si très souvent plusieurs versions sont offertes, le développeur doit initialiser chacune des versions d'une même extension afin qu'elle puisse fonctionner sur chacun des systèmes d'exploitation mobiles.

## **5.2 Perspectives de la boîte à outils**

À court terme, puisque cet essai s'imbrique en partie dans le travail de maîtrise de Mme Moretti, la boîte à outils développée pourrait lui faciliter la tâche quant à la mise sur pied de l'ACM. En effet, Mme Moretti avait soulevé, lors d'une rencontre avec M. Germain (Professeur au Département de géomatique de l'Université de Sherbrooke), la problématique qu'il n'existe pas encore de méthodologie ou boîte à outils permettant de développer des ACM de manière simple et efficace. La boîte à outils proposée lors de cet essai lui permettrait de créer son ACM en toute confiance sans douter de la pertinence des logiciels et librairies qu'elle utilisera.

À moyen terme, la boîte à outils pourrait faire l'objet d'un enseignement dans le cadre d'un cours de géomatique sur le web offert par le département. En effet, comme il a été mentionné au début de ce travail, les étudiants inscrits à la maîtrise en géomatique possèdent, pour la plupart, des cheminements scolaires différents et très diversifiés. Ainsi, alors que plusieurs d'entre eux pourraient ne pas savoir comment s'y prendre pour développer une ACM, la boîte à outils permettrait de les guider dans la création d'une telle application.

Finalement, à plus long terme, la boîte à outils pourrait être utilisée dans un contexte professionnel au sein d'OBNL ou d'entreprises.

## 6. Conclusion

Pour conclure, au début de cette étude, nous avons établi plusieurs objectifs, tant spécifiques qu'académiques, que nous avons tentés de relever tout au long de cet essai. Afin d'explorer le sujet des ACM et de leur développement en 2016, nous avons réalisé une revue de littérature qui a fait ressortir les catégories de logiciels et librairies essentielles au développement de telles applications. De plus, à partir de ces catégories, nos lectures ont permis d'identifier, pour chacune d'entre elles, plusieurs produits parmi les plus cités et utilisés de nos jours. Aussi, une revue de littérature concernant les méthodes d'aide à la décision a également été effectuée dans le but de sélectionner, pour chacune des catégories, le logiciel ou la librairie la plus pertinente. À partir de toutes ces informations, il a été possible d'effectuer une analyse multicritère de type PROMETHEE résultant en une boîte à outils comprenant une API cartographique, un cadre de travail adaptatif, un serveur cartographique et finalement un convertisseur d'applications web en application mobile. Afin de tester nos résultats, une ACM test a été créée. Les différents tests ont démontré que l'ACM test permettait bel et bien d'effectuer les fonctionnalités définies dans les objectifs.

Sur le plan pédagogique, ce travail a été extrêmement bénéfique à plusieurs niveaux. D'abord, il a été nécessaire d'avoir une très bonne organisation du temps afin d'être en mesure de compléter la rédaction dans les délais, mais aussi de peaufiner la stratégie de recherche documentaire. En effet, les documents sources nécessaires à la compréhension de l'analyse multicritère ont été particulièrement difficiles à trouver et, pour cette raison, un grand effort fut fourni pour la recherche de documents. Le travail de cette envergure a permis d'expérimenter quelque chose de complètement nouveau, notamment concernant la méthodologie, la forme et le respect de nombreuses normes de travail spécifiques à la rédaction d'un essai de maîtrise.

En dernier lieu, il serait bien intéressant, dans quelques années, de refaire un projet de recherche similaire afin de créer une boîte à outils strictement axée sur le développement d'application cartographique mobile. En effet, lors de cet essai, nous avons seulement couvert l'aspect d'utilisation d'une l'ACM sur plusieurs systèmes d'exploitation mobiles ainsi que l'utilisation de leurs extensions. Toutefois, bien d'autres volets sont disponibles dans le développement sur plateformes mobiles comme une utilisation hors-ligne des applications.

## **7. Limites du projet et recommandations**

En considérant l'ensemble du travail effectué, il est possible d'énumérer quelques limites pour ce travail. La première limite est le temps alloué à la rédaction. Puisque la période à laquelle nous avons été conseillés fut seulement de quatre mois, il aurait été possible de passer plus de temps sur la revue de littérature et, par le fait même, d'effectuer une analyse multicritère de plus grande envergure avec plus de logiciels. Une autre limite de ce travail a été l'ampleur limitée dans l'analyse. Par exemple, peut-être qu'il aurait été intéressant, pour l'analyse multicritère, de faire remplir des formulaires par des professionnels d'expérience en développement web afin d'obtenir un avis sur les différents librairies et logiciels. Par la suite, ces formulaires auraient pu être analysés afin d'alimenter certains critères de la méthode d'aide à la décision.

Comme dit précédemment, certains aspects de ce travail pourraient être améliorés, notamment par d'autres étudiants du département de géomatique de l'Université de Sherbrooke. Voici nos principales recommandations à cet effet :

- Élargir l'échantillon de librairies et de logiciels étudiés.
- Effectuer un sondage auprès de professionnels dans le domaine de la géomatique sur le web afin d'obtenir leur avis sur les librairies et les logiciels étudiés.
- Élaborer une application test de plus grande ampleur comprenant plus de fonctionnalités.
- Employer la boîte à outils dans une situation réelle de développement d'une ACM, autant dans un milieu professionnel qu'à l'Université de Sherbrooke.

Malgré ces recommandations, l'utilisation de cette boîte à outils demande malgré tout un certain niveau de connaissance dans le domaine de la géomatique sur le web. De ce fait, il serait pertinent de créer un guide de l'utilisateur comprenant, entre autres, la définition de certains termes et des captures d'écrans d'étapes clés permettant de faciliter l'utilisation de la boîte à outils.

## 8. Références

Adobe. *Introduction to media queries – part 1 : What are media queries?*, [En ligne], 2016, <http://www.adobe.com/devnet/archive/dreamweaver/articles/introducing-media-queries.html> (page consultée le 2 mars 2016).

Aggarwal, Karan, Hindle, Abram, et Stroulia, Eleni. Co-evolution of project documentation and popularity within github. In : *Proceedings of the 11th Working Conference on Mining Software Repositories*. ACM, 2014. p. 360-363.

Bathelot B. (2011). Application native. Dans Définitions marketing. Repéré à <http://guides.bib.umontreal.ca/uploads/uploads/original/tableau-apa.pdf?1436471733>

Beesley Darin J. et Michael Childs. System and method for calculating a navigation route based on adjacent cartographic map databases. US 6708112 B1 [En ligne]. 16 mars 2004. Disponible sur : <https://www.google.com/patents/US6708112#backward-citations> (consulté le 18 mai 2016).

Bootstrap. *Bootstrap is the most popular HTML, CSS, and JS framework for developing responsive, mobile first projects on the web*, [En ligne], 2016, <http://getbootstrap.com/> (page consultée le 27 mars 2016).

Borges Hudson, Marco Tulio Valente, Andre Hora et Jailton Coelho. On the Popularity of GitHub Applications: A Preliminary Note [PDF]. Repéré à <http://arxiv.org/pdf/1507.00604v2.pdf>

Buckler, Craig. *The Complete Guide to Reducing Page Weight*, [En ligne], 14 janvier 2014, <http://www.sitepoint.com/complete-guide-reducing-page-weight/> (Page consultée le 3 avril 2016).

Caillet, Renaud. Analyse multicritère : Étude de comparaison des méthodes existantes en vue d'une application en analyse de cycle de vie [PDF]. Repéré à <http://www.cirano.qc.ca/pdf/publication/2003s-53.pdf>

Esri France. *L'intelligence géographique, Qu'est-ce qu'un SIG?*, [En ligne], 2016, <http://www.esrifrance.fr/geomatique.aspx> (Page consultée le 14 mars 2016).

Figueria, José, Mousseau, Vincent et Roy, Bernard. In Figueria, José. Salvatore, Greco. In : *Multiple criteria decision analysis : State of the art surveys* . » Springer Verlag, 2005, p. 133-162.

FQCQ. *Carte interactive*, [En ligne], 2016, <http://www.fqcq.qc.ca/cartographie/carte-interactive/> (Page consultée le 16 mars 2016).

Fugioni, Glan M. «The Rise of the Digital Omnivore : What It Means for advertisers, Publishers, and app Developers», *Journal of advertising research*, juin 2015, p. 115-119.

Google Developers. *Commencez avec les cartes que vos utilisateurs adorent*, [En ligne], 2016, <https://developers.google.com/maps/documentation/javascript/?hl=fr> (Page consultée le 15 avril 2016).

Hasan Yousouf, Zaidi Mustafa, Haider Najmi, Hasan W.U. et Amin I. Smart Phones Application development using HTML5 and related technologies: A tradeoff between cost and quality. *International Journal of Computer Science Issues*, 2012, Vol. 9, no 3, p.455-461.

Henriet, Laurent. *Systèmes d'évaluation et de classification multicritères pour l'aide à la décision : construction de modèles et procédures d'affectation*, Thèse, Université de Paris, 2000, 161 p.

Hippel, Eric von et Krogh, Georg von. Open source software and the “private-collective” innovation model: Issues for organization science. *Organization science*, 2003, vol. 14, no 2, p. 209-223.

Hoarau, Charlotte. Orthoimage or map visualization in use in geoportals : Case study on the French Geoportal [PDF]. Repéré à <http://recherche.ign.fr/labs/cogit/publiCOGITDetail.php?idpubli=4753>

HTML5 Boilerplate. *The web's most popular front-end template*, [En ligne], 2016, <https://html5boilerplate.com/> (Page consultée le 15 avril 2016).

IDC Research, INC. *Smartphone OS Market Share, 2015 Q2* [En ligne], 2016, <http://www.idc.com/prodserv/smartphone-os-market-share.jsp> (Page consultée le 15 avril 2016).

Intelligent Networking And Collaborative (2009 : Barcelone). *Utilization of expert systems in thematic cartography*, IEEE, p.285-289.

Iosifescu Ionut, Marco Hugentobler et Lorenz Hurni. Cartographic web services and cartographic rules – a new approach for web cartography [PDF]. Repéré à [http://icaci.org/files/documents/ICC\\_proceedings/ICC2009/html/nonref/9\\_3.pdf](http://icaci.org/files/documents/ICC_proceedings/ICC2009/html/nonref/9_3.pdf)

Johnston, Lindsay (2013). *Geographic Information Systems : Concepts, Methodologies, Tools, ans Applications*, États-Unis, Édition Information Science Référence, 2172 p.

Kourouma Dan Lansana et Jean-Philippe Waaub. L'utilisation de l'aide multicritère en évaluation environnementale stratégique du secteur de l'énergie : comparaison d'options énergétiques dans l'étude de cas de la Guinée maritime [PDF]. Repéré à [http://www.sifee.org/static/uploaded/Files/ressources/actes-des-colloques/ouagadougou/session-1/4\\_Kourouma\\_communication.pdf](http://www.sifee.org/static/uploaded/Files/ressources/actes-des-colloques/ouagadougou/session-1/4_Kourouma_communication.pdf)

Leaflet. *An open-source JavaScript library for mobile-friendly interactive maps*, [En ligne], 2016, <http://leafletjs.com/> (Page consultée le 15 avril 2016).

Lee, Sang-Yong Tom, Kim, Hee-Woong, et Gupta, Sumeet. Measuring open source software success. *Omega*, 2009, vol. 37, no 2, p. 426-438.

Li, Jun, Xiong, Yingfei, Liu, Xuanzhe, *et al.* How does web service API evolution affect clients?. In : *Web Services (ICWS), 2013 IEEE 20th International Conference on*. IEEE, 2013. p. 300-307.

Michael, Ross & Cole. Native mobile apps: The wrong choice for business? Why businesses should think twice before building native mobile applications [PDF]. Repéré à <http://www.mrc-productivity.com/research/whitepapers/NativeAppsWrongChoice.pdf>

Neumann, Andreas. Web Mapping and Web Cartography. In : *Encyclopedia of GIS*. SpringerScience+Buisness Media, 2008. p . 1261-1269.

Oclc. (2010). Information consumer technologies 2010 [PDF]. Repéré à <http://www.oclc.org/content/dam/oclc/reports/2010perceptions/informationconsumertechnologies2010.pdf>

Onoue, Saya, Hideaki, H. A. T. A., Monden, Akito, *et al.* Investigating and Projecting Population Structures in Open Source Software Projects: A Case Study of Projects in GitHub. *IEICE TRANSACTIONS on Information and Systems*, 2016, vol. 99, no 5, p. 1304-1315.

Openlayers. *A high performance, feature-packed library for all your mapping needs*, [En ligne], 2016, <http://openlayers.org/>, (Page consultée le 19 février 2016).

Percivall, George. The application of open standards to enhance the interoperability of geoscience information. *International Journal of Digital Earth*, 2010, vol. 3, no S1, p. 14-30.

Peterson, Michael P. (2008). *International Perspectives on Maps and the Internet*, États-Unis, Édition Springer-Verlag, 440 p.

Peterson, Michael P. (2012). *Online Maps with APIs and WebServices*, États-Unis, Édition Springer-Verlag, 317 p.

Pick, James B. (2008). *Geo-Business : GIS in the Digital Horizon*, États-Unis, Édition John Wiley & Sons, Inc., 383 p.

Polymaps. *A JavaScript library for image- and vector-tiled map using SVG*, [En ligne], 2016, <http://polymaps.org/>, (Page consultee le 24 mars 2016).

Powers, David. *Introduction to media queries – Part 1: What are media queries?*, [En ligne], 25 juillet 2011, <http://www.adobe.com/devnet/archive/dreamweaver/articles/introducing-media-queries.html> (Page consultée le 20 mars 2016).

Pure. *A set of small, responsive CSS modules that you can use in every web project*, [En ligne], 2016, <http://purecss.io/>, (Page consultée le 27 mars 2016).

Reichenbacher, Tumasch. The world in your pocket-towards a mobile cartography. In : *Proceedings of the 20th International Cartographic Conference, Beijing, Chine*. 2001. p. 2514-2521.

Reichenbacher, Tumasch. *Mobile Cartography – Adaptive Visualisation of Geographic Information on Mobile Devices*, Thèse, Université de Munich, 2004, 189 p.

Ressources naturelles Canada [RNCCan]. (2013, 18 octobre). *CGDI in Action: Geomatics - Improving our everyday world* [Vidéo en ligne]. Repéré à <http://www.nrcan.gc.ca/earth-sciences/geomatics/canadas-spatial-data-infrastructure/publications/8944>

Rost, Mattias, Cramer, Henriette, Belloni, Nicolas, *et al.* Geolocation in the mobile web browser. In : *Proceedings of the 12th ACM international conference adjunct papers on Ubiquitous computing-Adjunct*. ACM, 2010. p. 423-424.

Rouffignat, Joel. Analyse multicritère et SIG pour faciliter la concertation en aménagement du territoire : vers une amélioration du processus décisionnel?. *Cahiers de géographie du Québec*, 2003, Vol. 47, no 130, p. 35-61.

Schiller Jochen et Agnès Voisard. (2004). *Location Based-Services*, États-Unis, Édition Elsevier Inc., 231 p.

Schnabel Olaf et Lorenz Hurni. *Cartographic web applications – developments and trends* [PDF]. Repéré à [http://icaci.org/files/documents/ICC\\_proceedings/ICC2009/html/nonref/13\\_4.pdf](http://icaci.org/files/documents/ICC_proceedings/ICC2009/html/nonref/13_4.pdf)

Semantic UI. *User interface is the language of the web*, [En ligne], 2016, <http://semantic-ui.com/> (Page consultée le 15 avril 2016).

Serrano, Nicolas et Sarriegi, Jose Maria. Open source software ERPs: a new alternative for an old need. *IEEE software*, 2006, vol. 23, no 3, p. 94.

Skeleton. *A dead simple, responsive boilerplate*, [En ligne], 2016, <http://getskeleton.com/> (Page consultée le 2 mars 2016).

Statista. *The Statistics Portal : Market share held by leading desktop internet browsers in the United States from January 2015 to April 2016* [En ligne], 2016, <http://www.statista.com/statistics/272697/market-share-desktop-internet-browser-usa/> (Page consultée le 28 février 2016).

Statista. *The statistic portal : Number of smartphone users worldwide from 2014 to 2019 (in millions)*, [En ligne], 2016, <http://www.statista.com/statistics/330695/number-of-smartphone-users-worldwide/> (Page consultée le 28 février 2016).



Steiniger, Stefan et Andrew J.S. Hunter. Free and Open Source GIS Software for Building a Spatial Data Infrastructure. In : *Free and Open Source GIS Software for Building a Spatial Data Infrastructure*, Springer Verlag, 2011, p. 247-261.

Urli, Bruno. (2013). Méthode Omnicritère - méthode d'aide à la concertation, à la décision, et à la gestion de projet [PDF]. Repéré à <http://www.bv.transports.gouv.qc.ca/mono/1169003.pdf>

Vélo-Québec. *Route verte*, [En ligne], 2016, <http://carto.routeverte.com/fr> (Page consultée le 28 février 2016).

Wei, Ran. Motivations for using the mobile phone for mass communications and entertainment. *Telematics and Informatics*, 2008 Vol. 25, no 1, p.36-46.

Zurb. *Foundation : The most advanced responsive front-end framework in the world* [En ligne], 2016, <http://foundation.zurb.com/> (Page consultée le 12 février 2016).